



**Universidade de Aveiro**  
2012

Departamento de Electrónica,  
Telecomunicações e Informática

**SAMUEL DA  
COSTA CAMPOS**

**DICOOGLE – PLATAFORMA DE ANÁLISE  
PARA REDES DE IMAGEM MÉDICA**

**DICOOGLE – ANALYSIS PLATFORM TO  
MEDICAL IMAGING NETWORKS**





**SAMUEL DA  
COSTA CAMPOS**

**DICOOGLE – PLATAFORMA DE ANÁLISE  
PARA REDES DE IMAGEM MÉDICA**

**DICOOGLE – ANALYSIS PLATFORM TO  
MEDICAL IMAGING NETWORKS**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Dr. Carlos Manuel Azevedo Costa, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.



## **o júri**

presidente

**Prof. Dr. António Manuel Melo de Sousa Pereira**

Professor Catedrático do Departamento de Eletrónica,  
Telecomunicações e Informática da Universidade de Aveiro

vogais

**Prof. Dr. Rui Pedro Sanches de Castro Lopes**

Professor Coordenador do Departamento de Informática e  
Comunicações do Instituto Politécnico de Bragança

**Prof. Dr. Carlos Manuel Azevedo Costa**

Professor Auxiliar do Departamento de Eletrónica, Telecomunicações  
e Informática da Universidade de Aveiro



## **Agradecimentos / acknowledgments**

Ao longo dos vários meses de trabalho recebi o apoio de diversas pessoas, sem as quais não teria conseguido atingir esta meta.

Agradeço em primeiro lugar ao meu professor e orientador Carlos Costa pelo acompanhamento, disponibilidade, orientação, ajuda e ideias inspiradoras.

Agradeço a todo o grupo de bioinformática do IEETA, sobretudo ao Luís Bastião, pelo apoio, ajuda e acompanhamento em todo o processo.

Ao Dr. José Ribeiro pelos esclarecimentos e indicações que me fizeram perceber algumas das necessidades que existem atualmente e ajudaram a estabelecer os requisitos do sistema.

A todos os meus amigos da UA pelas ideias e dicas que fomos trocando.

Em especial ao meu pai, João, e à minha mãe, Leonilde, pela oportunidade de aqui estar a concluir esta etapa, ao meu irmão, David, pelo suporte e apoio incondicional.

Finalmente, agradeço e dedico esta dissertação à minha amada noiva, Daniela, pela paciência, afeto e ânimo nos momentos em que a motivação escasseava.

A todos estes e, acima de tudo, a Deus um muito obrigado.





**Palavras-chave**

Fluxo de Trabalho; PACS; DICOM; Radiologia; Analisador de Rede; Avaliação de Serviços Médicos

**Resumo**

O uso de sistemas de informação em instituições de saúde tem aumentado nos últimos anos. O crescente uso de sistemas computacionais tem feito aumentar a produtividade e qualidade dos serviços prestados nas instituições de saúde. Estes sistemas geram uma quantidade enorme de informação digital que pode ser processada e analisada para extrair métricas relevantes para a prática clínica. Contudo, existem poucos sistemas que permitam agregar informação proveniente das várias fontes de dados existentes nas instituições médicas. Além disso, existem algumas fontes de dados que raramente são usadas e permitem extrair informação potencialmente útil para avaliação e otimização dos fluxos de trabalho e da qualidade dos serviços prestados. Esta dissertação apresenta um sistema baseado em sensores de informação, capaz de recolher informação de várias fontes existentes em departamentos de imagiologia. O sistema permite que os utilizadores analisem e estudem a informação recolhida. Adicionalmente o sistema permite obter métricas de avaliação do desempenho e qualidade dos serviços prestados. Estas métricas são importantes para avaliar, planear e otimizar os serviços, numa época em que a racionalização de recursos na saúde é uma realidade.



**Keywords**

Workflow; PACS; DICOM; Radiology; Network Analyzer; Medical Services Evaluation

**Abstract**

The usage of information systems in healthcare institutions has increased in recent years. The growing use of computer systems has increased the productivity and quality of provided services by medical institutions. These systems can generate a huge amount of digital information. This digital information can be processed and analyzed to extract relevant metrics for the clinical practice. However, there are few systems capable of assemble multiple data sources within a medical institution. Moreover, there are some data sources that usually are not used to extract information that can be useful to evaluate and optimize the workflows and the quality of provided services. This thesis presents a system based on information sensors, capable of collecting information from several sources found in radiology departments. This system allows users to analyze and study the collected information from the various sources. The system also retrieves metrics for evaluating the performance and quality of provided services. These metrics are important in planning and service optimization, at a time when the rationalization of resources in healthcare is a reality.



# Conteúdos

<b>1</b>	<b>Introdução .....</b>	<b>1</b>
1.1.	Visão Geral.....	1
1.2.	Objetivos da Dissertação.....	2
1.3.	Estrutura da Dissertação.....	3
<b>2</b>	<b>Sistemas de Informação em Saúde .....</b>	<b>5</b>
2.1.	Sistemas de informação usados em medicina .....	5
2.1.1.	Sistemas PACS.....	6
2.2.	Norma DICOM .....	7
2.2.1.	Dicionário de dados.....	7
2.2.2.	Modelo de Informação .....	8
2.2.3.	Formato dos ficheiros .....	10
2.2.4.	Protocolo de Rede.....	11
2.3.	HL7 .....	16
2.4.	Ecocardiografia .....	16
2.5.	Dicoogle .....	19
2.6.	Monitorização de fluxos de trabalho em instituições médicas.....	20
<b>3</b>	<b>Conceptualização da plataforma.....</b>	<b>23</b>
3.1.	Os desafios.....	23
3.2.	Requisitos funcionais.....	23
3.3.	Requisitos não funcionais.....	24
3.3.1.	Segurança .....	25
3.4.	Arquitetura geral do sistema.....	26
3.4.1.	Segurança das comunicações.....	27
3.4.2.	Anonimização de dados clínicos.....	27
3.5.	Adaptações ao Dicoogle.....	29
3.5.1.	Serviço de pré-processamento.....	30
<b>4</b>	<b>Sensor de rede.....</b>	<b>33</b>
4.1.	Componentes usados.....	33
4.2.	Reconstrução dos fluxos TCP .....	34
4.3.	Arquitetura.....	35
4.4.	Redes de <i>switches</i> .....	38
4.5.	Alertas e extração de imagens transmitidas.....	38
4.6.	<i>Web services</i> .....	40
4.6.1.	Gestão de alertas .....	40
4.6.2.	Gestão da aplicação .....	41
4.6.3.	Pesquisa de informação.....	43
<b>5</b>	<b>Sensor de relatórios de ecocardiogramas .....</b>	<b>45</b>
5.1.	Formato dos relatórios .....	45
5.1.1.	Processo de recolha dos relatórios.....	47
5.2.	Componentes usados.....	47
5.3.	Arquitetura.....	48
5.3.1.	Normalização e processamento das estruturas dos relatórios.....	51
5.4.	<i>Web services</i> .....	52
5.4.1.	Serviço de gestão do sensor.....	52
5.4.2.	Serviço de localização.....	53

5.4.3. Serviço de pesquisa .....	54
5.4.4. Serviço de pré-processamento .....	56
<b>6 Medical Analysis Web .....</b>	<b>57</b>
6.1. Visão geral .....	57
6.2. Arquitetura.....	59
6.2.1. CakePHP .....	60
6.2.2. AJAX.....	61
6.2.3. Outros componentes usados .....	62
6.2.4. Comunicação com os sensores .....	64
6.3. Interface de utilizador .....	66
6.3.1. Área de trabalho .....	66
6.3.2. Conta de utilizador .....	68
6.3.3. Pesquisa.....	69
6.3.4. Métricas e estatísticas .....	71
6.3.5. Administração.....	72
<b>7 Resultados e discussão.....</b>	<b>77</b>
7.1. Sensor de rede .....	77
7.1.1. Desempenho e fiabilidade .....	77
7.2. Sensor de relatórios de ecocardiogramas .....	77
7.3. Medical Analysis Web.....	79
7.3.1. Modalidades de imagens médicas.....	79
7.3.2. Tipos de relatórios de ecocardiografias .....	80
7.3.3. Evolução do número de ecocardiogramas por ano .....	80
7.3.4. Qualidade dos exames.....	81
7.3.5. Duração e número de imagens dos exames.....	83
7.3.6. Origem dos pacientes .....	84
7.3.7. Atividade na rede do PACS.....	85
7.4. Discussão .....	86
<b>8 Conclusão e Trabalho Futuro .....</b>	<b>87</b>
8.1. Principais questões.....	87
8.2. Principais contribuições.....	87
8.3. Trabalho Futuro.....	88
<b>9 Referências.....</b>	<b>91</b>
<b>10 Anexos .....</b>	<b>95</b>
10.1. Diagrama físico da base de dados do sensor de rede .....	95
10.2. Captura e decodificação de pacotes.....	96
10.3. Construção de <i>Queries</i> .....	97

## Lista de Figuras

Figura 1.1: Uso dos meios informáticos nos hospitais portugueses [1,2].	1
Figura 2.1: Componentes principais de um sistema PACS [5].	6
Figura 2.2: Formato TLV usado na norma DICOM.	8
Figura 2.3: Principais estruturas do <i>DICOM Information Model</i> [12].	8
Figura 2.4: Quatro níveis da hierarquia de informação DICOM. Cada nível é identificado pelo seu elemento chave.	9
Figura 2.5: Estrutura conceptual de um ficheiro DICOM. No cabeçalho ilustra alguns dos elementos de informação contidos no ficheiro.	10
Figura 2.6: Fluxo normal dos DICOM PDUs. As setas vermelhas correspondem a falhas e as verdes correspondem ao fluxo de associações bem sucedidas (adaptado de [5]).	13
Figura 2.7: Analogia do modelo SCU-SCP [5].	13
Figura 2.8: Serviço de armazenamento DICOM (C-STORE).	14
Figura 2.9: Serviço de pesquisa DICOM (C-FIND).	15
Figura 2.10: Serviço de recuperação DICOM (C-GET).	15
Figura 2.11: Ecógrafo moderno da Siemens.	17
Figura 2.12: Exemplo de uma imagem de um ecocardiograma transtorácico [17].	18
Figura 2.13: Arquitetura de alto-nível do Dicoogle [7].	20
Figura 2.14: Janela principal do DICOMetrix [27].	21
Figura 2.15: Janela principal do DICOM Detective [29].	22
Figura 2.16: Wireshark a apresentar os pacotes DICOM [30].	22
Figura 3.1: Arquitetura de alto nível do sistema.	26
Figura 3.2: Arquitetura de anonimização.	28
Figura 3.3: Preferências de Anonimização do sensor de rede.	28
Figura 3.4: Exemplo de uma lista com dois resultados em resposta a uma pesquisa no Dicoogle.	30
Figura 4.1: Principais componentes do sensor de rede.	33
Figura 4.2: Classes do <i>package</i> TCP.	35
Figura 4.3: Diagrama de Classes ( <i>package</i> DICOM.UL).	36
Figura 4.4: Diagrama de Classes ( <i>package</i> DICOM).	37
Figura 4.5: Janelas principal (à esquerda) e de preferências (à direita) do sensor de rede.	38
Figura 4.6: Diagrama de classes do <i>package</i> DICOM.ProcessObjects.	39
Figura 4.7: Exemplo de uma regra de alerta com duas condições, em formato XML.	40
Figura 4.8: Mensagem HTTP POST com as definições do sensor de rede em formato JSON.	42
Figura 4.9: Estado atual das interfaces de rede em formato JSON.	42

Figura 4.10: Mensagem HTTP POST usada para definir o estado das interfaces de rede em formato JSON. ....	42
Figura 4.11: Exemplo de estrutura enviada para os métodos de pesquisa, em formato XML. ....	43
Figura 4.12: Retorno de um registo de um comando C-STORE como resposta a uma pesquisa. ....	44
Figura 5.1: Porção da estrutura XML de um relatório antes de 2009. ....	46
Figura 5.2: Porção do conteúdo do campo "Relat" com texto semiestruturado. ....	46
Figura 5.3: Processo de recolha e indexação dos relatórios. ....	47
Figura 5.4: Diagrama de componentes do sensor de relatórios. ....	48
Figura 5.5: Diagrama de classes com as classes usadas na indexação de relatórios com omissão das classes usadas para obter o conteúdo das estruturas XML ( <i>package</i> "XML"). ....	50
Figura 5.6: Preferências de indexação do sensor de rede. ....	51
Figura 5.7: Diagrama de classes do <i>package</i> XML com adição das classes RelatDocument e FieldTranslations. ....	51
Figura 5.8: Parte da estrutura com a lista de correspondências entre os valores numéricos e textuais dos campos em formato JSON. ....	53
Figura 5.9: Parte inicial da lista de distritos portugueses em formato JSON. ....	54
Figura 5.10: Retorno do método de pesquisa com um único resultado em formato XML. ....	56
Figura 6.1: Casos de utilização relacionados com a pesquisa de informação nos sensores. ....	57
Figura 6.2: Casos de utilização relacionados com as estatísticas e métricas extraídas pela aplicação. ....	58
Figura 6.3: Casos de utilização relacionados com as contas de utilizadores. ....	58
Figura 6.4: Casos de utilização apenas disponíveis aos administradores. ....	59
Figura 6.5: Mecanismos de comunicação do Medical Analysis Web. ....	60
Figura 6.6: Arquitetura MVC usada pelo CakePHP [54]. ....	61
Figura 6.7: Modelo físico da Base de Dados do Medical Analysis Web. ....	61
Figura 6.8: Modelos de desenvolvimento de aplicações Web. À esquerda o modelo clássico e à direita o novo modelo AJAX (adaptado de [56]). ....	62
Figura 6.9: Formas de acesso aos sensores a partir do cliente Web. ....	64
Figura 6.10: Classes de conexão aos sensores. ....	65
Figura 6.11: Diagrama de sequência de uma pesquisa nos repositórios de imagens médicas com dois sensores de imagens médicas. ....	66
Figura 6.12: Organização gráfica da aplicação Web desenvolvida. ....	67
Figura 6.13: Página inicial do Medical Analysis Web com indicação a vermelho das áreas que apenas são visíveis aos administradores. ....	67
Figura 6.14: Página de início de sessão. ....	68
Figura 6.15: Diagrama de atividades do processo de registo de um utilizador. ....	68
Figura 6.16: Formulário de edição de perfil de um utilizador. ....	69



Figura 6.17: Página de pesquisa com resultados anonimizados de uma pesquisa por relatórios de ecocardiogramas.....	70
Figura 6.18: Janela de listagem de todos os campos indexados para um relatório (Dump). ....	71
Figura 6.19: Zona de pesquisa no sensor de rede. ....	71
Figura 6.20: Gráfico gerado pela aplicação com o número de imagens médicas digitais transferidas ao longo do dia 27 de Junho de 2012, registadas pelo sensor “NetworkSensor2”.....	72
Figura 6.21: Lista de sensores utilizados pelo Medical Analysis Web. ....	72
Figura 6.22: Parte do formulário de visualização e edição de definições do sensor de rede....	73
Figura 6.23: Página de interação com o serviço de pré-processamento durante a execução do mesmo. ....	73
Figura 6.24: Condições de uma regra de alerta para o comando C-STORE configurada no sensore “NetworksSensor2”.....	74
Figura 6.25: Lista de utilizadores do Medical Analysis Web. ....	74
Figura 6.26: Formulário de edição de um perfil fornecido ao administrador. ....	75
Figura 7.1: Gráfico que apresenta as modalidades de imagens médicas digitais indexadas e a sua respetiva percentagem. ....	79
Figura 7.2: Evolução do número de exames entre os anos 1999 e 2012 salientando o peso de cada faixa etária.....	80
Figura 7.3: Evolução do peso de cada faixa etária entre os anos 1999 e 2012 para o número final de ecocardiogramas realizados.....	81
Figura 7.4: Qualidade dos exames por faixa etária com os tipos de qualidade representados em percentagem.....	81
Figura 7.5: Qualidade dos exames por peso do paciente com os tipos de qualidade representados em percentagem. ....	82
Figura 7.6: Qualidade do exame por ecógrafo com mais de 1000 exames realizados. ....	82
Figura 7.7: a) Tempo médio de duração do exame em função da qualidade do mesmo. b) Número médio de imagens adquiridas também em função da qualidade. ....	83
Figura 7.8: Tempo médio de aquisição por imagem para cada tipo de qualidade de exame....	84
Figura 7.9: Tempo médio de aquisição de uma imagem médica em função do ecógrafo utilizado. ....	84
Figura 7.10: Origem dos pacientes da instituição. ....	85
Figura 7.11: Resumo da atividade da última hora, com os principais comandos DICOM, na rede do PACS.....	85
Figura 10.1: Sequência de invocações entre as classes responsáveis pela captura e descodificação de pacotes.....	96
Figura 10.2: Análise de objetos DICOM DIMSE quando o PDU é do tipo P-Data-TF e ocorre anonimização da informação recolhida pelo sensor.....	97
Figura 10.3: Janela de ajuda à construção de <i>queries</i> para os sensores de relatórios de ecocardiogramas.....	98

## Lista de Tabelas

Tabela 2.1: Principais serviços DIMSE.....	14
Tabela 3.1: Métodos do <i>web service</i> de anonimização.....	29
Tabela 3.2: <i>Web services</i> previamente existentes no Dicoogle.....	29
Tabela 3.3: Serviços desenvolvidos no Dicoogle para o sistema desenvolvido.....	30
Tabela 4.1: Métodos do serviço de gestão de alertas.....	40
Tabela 4.2: Métodos do serviço de gestão da aplicação.....	41
Tabela 4.3: Métodos do serviço de pesquisa de informação.....	43
Tabela 5.1: Métodos do serviço de gestão do sensor.....	52
Tabela 5.2: Métodos do serviço de localização.....	53
Tabela 5.3: Métodos do serviço de pesquisa do sensor de relatórios.....	55
Tabela 5.4: Métodos do serviço de pré-processamento do sensor de relatórios de ecocardiogramas.....	56
Tabela 7.1: Tempos de indexação do sensor de relatórios de ecocardiogramas com um <i>indexing step</i> de 50.....	78
Tabela 7.2: Tempos de indexação sem anonimização do sensor de relatórios de ecocardiogramas variando o <i>indexing step</i> .....	78

## Acrónimos

<b>Termo</b>	<b>Descrição</b>
<b>AE</b>	Application Entity
<b>AJAX</b>	Asynchronous JavaScript and XML
<b>API</b>	Application Programming Interface
<b>CSS</b>	Cascading Style Sheets
<b>CSV</b>	Comma-Separated Values
<b>CT</b>	Computed Tomography
<b>DICOM</b>	Digital Imaging and Communication in Medicine
<b>DIM</b>	DICOM Information Model
<b>DIMSE</b>	DICOM Message Service Elements
<b>DR</b>	Digital Radiography
<b>DS</b>	Digital Subtraction Angiography
<b>FTP</b>	File Transfer Protocol
<b>HIS</b>	Hospital Information System
<b>HTTP</b>	Hypertext Transfer Protocol
<b>JSON</b>	JavaScript Object Notation
<b>MIME</b>	Multipurpose Internet Mail Extensions
<b>MPPS</b>	Modality Performed Procedure Step
<b>MR</b>	Magnetic Resonance
<b>MRI</b>	Magnetic Resonance Imaging
<b>MVC</b>	Model-View-Controller
<b>ODBC</b>	Open Data Base Connectivity
<b>PACS</b>	Picture Archiving and Communication System
<b>PDU</b>	Protocol Data Unit
<b>PERL</b>	Practical Extraction and Report Language
<b>REST</b>	Representational State Transfer
<b>RIS</b>	Radiology Information System
<b>SCP</b>	Service Class Provider
<b>SCU</b>	Service Class User
<b>SOA</b>	Service-Oriented Architecture
<b>SOP</b>	Service Object Pair
<b>SSL</b>	Secure Sockets Layer
<b>TCP</b>	Transmission Control Protocol

<b>TLS</b>	Transport Layer Security
<b>TLV</b>	Tag, Length, Value
<b>UID</b>	Unique Identifier
<b>URL</b>	Uniform Resource Locator
<b>US</b>	Ultrasound
<b>WAN</b>	Wide Area Network
<b>XA</b>	X-Ray Angiography

# 1 Introdução

Este capítulo contém a introdução da dissertação, a definição dos objetivos e a estrutura geral da mesma.

## 1.1. Visão Geral

O uso de sistemas informáticos nas mais variadas áreas da nossa vida quotidiana tem vindo a aumentar nos últimos anos. O sector da saúde não é exceção e, por isso, esta tendência tem sido verificada especialmente nas instituições médicas que prestam serviços de saúde [1,2]. Estes sistemas são utilizados, maioritariamente, para gerir o registo dos pacientes, os exames e o respetivo relatório.

A Figura 1.1 apresenta um gráfico com dados do Instituto Nacional de Estatística com a evolução do uso de sistemas de informação nos hospitais portugueses entre os anos 2004 e 2010. Esta figura apresenta a evolução relativa à troca de imagens médicas, de ficheiros e ao planeamento e marcação de atividades. Observando o gráfico, podemos verificar que existe uma tendência de subida generalizada dos três indicadores. No entanto, a troca interna de imagens médicas é o indicador que apresenta a subida mais acentuada.

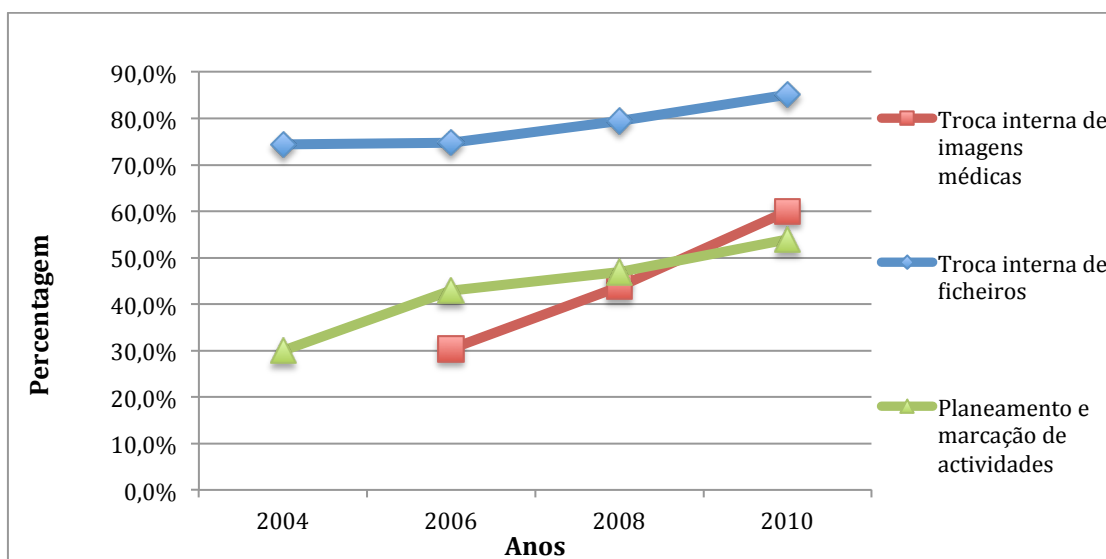


Figura 1.1: Uso dos meios informáticos nos hospitais portugueses [1,2].

Os aparelhos e meios informáticos disponíveis atualmente nas instituições médicas vieram aumentar o seu desempenho de forma muito significativa [3,4]. Além disso, a informação em formato digital traz enormes potencialidades no que diz respeito à gestão e fiabilidade das atividades desenvolvidas nessas instituições. Esta permite criar mecanismos automáticos de processamento para otimizar o tempo e os recursos utilizados, garantindo mesmo assim o normal funcionamento da instituição médica.

A grande quantidade de informação em formatos digitais tornou possível a criação de grandes repositórios de dados médicos, nas instituições de saúde. Estes contêm a informação relativa aos pacientes, aos exames, aos profissionais de saúde e à própria instituição.

O manuseamento de imagens médicas digitais é normalmente feito com recurso a sistemas PACS (*Picture Archiving and Communication System*). Estes sistemas trazem muitas

vantagens relativamente aos sistemas analógicos tradicionais que se baseiam em imagens guardadas em películas ou em papel. Um sistema PACS contempla 4 grandes áreas. São elas: a aquisição digital das imagens, os mecanismos e rede de transmissão, as estações de trabalho, utilizadas para visualização e interpretação das imagens, e o arquivo digital das mesmas. A esmagadora maioria dos computadores e dispositivos utilizados nestas 4 áreas usam a norma DICOM (*Digital Imaging and Communications in Medicine*). Esta define o protocolo de transferência e o formato de armazenamento das imagens médicas digitais [5]. É graças a esta norma que os dispositivos e *software* dos diversos fabricantes podem hoje cooperar e ser interoperáveis.

Os sistemas PACS representam uma parte significativa de todos os sistemas informáticos utilizados nos hospitais e instituições de saúde, mas não são os únicos. Existem outros sistemas informáticos importantes na área da saúde como, por exemplo, o HIS (*Hospital Information System*) e o RIS (*Radiology Information System*) que, na maioria dos casos, são proprietários. Além destes existem também muitas instituições de saúde que possuem outros tipos de sistemas de informação para fazer face às necessidades específicas da instituição numa determinada área.

Apesar de todas as vantagens que estes novos sistemas de informação trouxeram às instituições médicas, em muitos casos, é difícil tirar partido de toda a informação dispersa, armazenada nesses sistemas, pois a integração entre eles não é elevada. Muitas das potencialidades decorrentes de ter informação médica em formato digital não são aproveitadas por causa da dispersão e variedade de formatos dessa mesma informação.

A atual conjuntura económica tem trazido à saúde a necessidade de aproveitar todas as potencialidades existentes de forma a aumentar o desempenho, eficiência e qualidade dos serviços prestados e, ao mesmo tempo, reduzir custos. O contexto socioeconómico que se vive atualmente em Portugal, salienta ainda mais esta necessidade. São muitos os pontos do memorando de entendimento entre Portugal, a União Europeia, o Banco Central Europeu (BCE) e o Fundo Monetário Internacional (FMI) (1.9, 1.10, 1.20, 1.29, 3.50, entre outros) que apontam para esta obrigação de redução de custos e racionalização de recursos na saúde, por parte do estado português [6].

A informação que está disponível nos vários sistemas informáticos das instituições médicas pode ser utilizada para medir o desempenho, qualidade dos serviços prestados e extrair informação relevante para os profissionais de saúde. O sistema apresentado nesta dissertação irá de encontro a esta necessidade e poderá ajudar nestas tarefas de avaliação e extração de conhecimento para ajudar a aumentar a eficiência das instituições de saúde.

## **1.2. Objetivos da Dissertação**

Os sistemas apresentados anteriormente permitem armazenar e gerir grandes quantidades de informação médica. Esta informação é utilizada, em processos muito bem definidos, como suporte às atividades quotidianas das instituições médicas. Existem, no entanto, outros contextos onde estas informações podem ser utilizadas, como por exemplo a investigação científica, extração de métricas de qualidade desempenho, assim como permitir a deteção de possíveis problemas na prestação de serviços.

Esta dissertação pretende apresentar um sistema capaz de integrar diversas fontes de informação relativas a laboratórios de imagem médica, disponíveis em grande parte das instituições médicas. A integração será feita através de sensores distribuídos que permitirão pesquisar e analisar a informação disponível nessas fontes, de forma centralizada numa aplicação que interage com os sensores.

As fontes de informação utilizadas serão o fluxo de imagens médicas digitais na rede informática, o repositório de imagens médicas digitais e, como caso de estudo, um repositório não normalizado de relatórios de ecocardiogramas. Adicionalmente, a arquitetura definida permitirá incluir outras fontes de dados.

O fluxo de imagens médicas na rede será útil para registrar e analisar a atividade na rede do PACS de qualquer departamento de imagiologia. As restantes fontes de informação foram escolhidas porque contêm informação médica que pode ser utilizada em alguns dos contextos referidos acima e porque foi dado acesso a dois grandes repositórios. Um repositório de imagens médicas digitais com cerca de 1,6 milhões imagens (um exame produz, normalmente, várias imagens médicas), a maioria das quais de exames de ecocardiografia, e outro com mais de 84.000 relatórios de ecocardiogramas relativos aos mesmos exames de ecocardiografia do repositório anterior.

### ***Sensor de imagens médicas***

O repositório de imagens médicas digitais é um conjunto de ficheiros que seguem o formato definido na norma DICOM. Para facilitar a pesquisa e o acesso a estes ficheiros, normalmente eles são registados em bases de dados ou indexados em servidores de arquivos DICOM. No trabalho desenvolvido será utilizada a aplicação Dicoogle [7] para aceder à informação destes repositórios. Alguns detalhes sobre esta aplicação serão apresentados no subcapítulo 2.5.

Para as restantes fontes de informação é necessário desenvolver os sensores pois o que existe na área, que será apresentado no próximo capítulo desta dissertação, não permite obter a informação pretendida para análise do fluxo de trabalho e extração de conhecimento.

### ***Sensor de fluxos***

O sensor de fluxos de imagens médicas digitais na rede informática é uma ferramenta que analisa o tráfego da rede da instituição e permite registrar a informação relevante que está contida nos pacotes do protocolo DICOM. Esta aplicação irá capturar e analisar o tráfego que circula na rede em tempo real, guardar a informação imagiológica relevante e, por fim, disponibilizar serviços que permitam pesquisar e analisar a informação guardada.

### ***Sensor de relatórios de ecocardiogramas***

Por último, é necessária uma ferramenta capaz de analisar o repositório de relatórios de ecocardiogramas e que permita estudar a informação ali contida. Ao contrário do que acontece com o repositório de imagens médicas digitais, cujo formato segue a norma DICOM, os relatórios de ecocardiogramas aos quais se obteve acesso não estão normalizados. O protocolo usado para fazer transferência deste tipo de relatório é, tipicamente, o HL7. O sensor desenvolvido tem módulos específicos para os formatos dos relatórios utilizados como casos de estudo. No entanto, pretende-se que a adaptação deste sensor a novos formatos não exija um esforço muito grande de trabalho e tempo. À semelhança dos sensores anteriores, esta ferramenta permitirá realizar pesquisas sobre a informação indexada e terá mecanismos de comunicação com o exterior.

## **1.3. Estrutura da Dissertação**

Esta dissertação está dividida em 8 capítulos. De seguida faremos uma breve descrição do seu conteúdo:

**Capítulo 2:** apresenta os sistemas usados em instituições de saúde, dando especial ênfase aos sistemas PACS. A norma DICOM também será abordada neste capítulo em

conjunto com breves noções de ecocardiografia. Para finalizar este capítulo serão apresentados alguns dos principais analisadores DICOM e as características principais de cada um deles.

**Capítulo 3:** detalha os requisitos do projeto. Apresenta a arquitetura proposta para o sistema e as adaptações feitas às ferramentas pré-existentes para que se enquadrassem na arquitetura proposta.

**Capítulo 4:** descreve a arquitetura e implementação do sensor de rede realçando as suas funcionalidades e interfaces com o exterior através de *web services*.

**Capítulo 5:** apresenta o sensor de relatórios de ecocardiogramas começando por apresentar a estrutura dos próprios relatórios, os métodos de acesso aos mesmos, a arquitetura do sensor e, por fim, as suas interfaces com o exterior.

**Capítulo 6:** apresenta o componente do sistema que permite agregar e estudar a informação obtida por todos os sensores. Foca-se na arquitetura, interface com o utilizador e nas sequências de ações necessárias para produzir informação relevante.

**Capítulo 7:** apresenta os resultados obtidos com o sistema desenvolvido, principalmente ao nível de desempenho do sistema e indicadores obtidos.

**Capítulo 8:** contém as conclusões do trabalho desenvolvido e aponta direções para desenvolver trabalho futuro na área.



## 2 Sistemas de Informação em Saúde

Este capítulo apresenta uma visão geral sobre os sistemas de informação utilizados em saúde, especialmente os sistemas PACS. Também são apresentados os protocolos e normas importantes para o trabalho realizado.

### 2.1. Sistemas de informação usados em medicina

Os sistemas de informação têm permitido automatizar muitas tarefas dentro das instituições de saúde. Os meios analógicos tradicionais para guardar e transferir informação têm sido progressivamente substituídos por meios digitais, devido às suas reconhecidas vantagens [4].

A quantidade de informação digital existente nas instituições médicas, especialmente nos hospitais, é grande e tem tendência a aumentar à medida que as instituições usam cada vez mais sistemas de informação para suportar as suas atividades. Ao nível da imagiologia digital, a quantidade de informação também tem tendência a aumentar à medida que as instituições compram mais aparelhos de aquisição digitais e a quantidade de informação produzida pelos novos equipamentos também aumenta. Por exemplo, os novos aparelhos de aquisição melhoram frequentemente a qualidade das imagens médicas digitais produzidas aumentando a resolução das mesmas.

É portanto imperativo que existam sistemas capazes de processar, armazenar e organizar toda a informação gerada. É ainda necessário que os sistemas de informação hospitalares consigam disponibilizar toda a informação necessária em tempo útil. Só assim será possível aos profissionais de saúde prestar serviços de forma eficiente e ter todas as informações e meios necessários para tomar as melhores decisões médicas. Nos últimos anos têm-se atribuído cada vez mais importância a estes sistemas de informação. A sua importância surge principalmente a dois níveis: económico e eficiência dos serviços prestados [8].

O HIS é um sistema que gere principalmente dois tipos de informação: médica e administrativa. Este pode ser considerado o sistema de informação base de um hospital, sendo os restantes subsistemas deste, embora os restantes possam existir de forma isolada numa instituição. Uma das tarefas importantes do HIS é gerir a faturação das atividades hospitalares [9].

O RIS é um sistema de informação complementar ao HIS que é usado nos departamentos de imagiologia. Este sistema tem capacidades de monitorização e marcação de exames para as diversas modalidades, gestão dos relatórios de exames e referenciação das imagens existentes no PACS [9].

Existem várias modalidades de imagens médicas digitais tais como: Tomografias Computorizadas (CT), Ultrassonografias (US), Ressonâncias Magnéticas (MR ou MRI), Angiografias (XA), Radiografias Computorizadas (CR), entre outras [9]. Os sistemas PACS são os sistemas de informação que processam, armazenam e gerem as imagens médicas em formato digital. Os PACS são utilizados em conjunto com o RIS nos departamentos de imagiologia. As imagens do PACS devem estar associadas com os registos de exames do RIS. Estes sistemas serão abordados com maior detalhe no subcapítulo 2.1.1.

### 2.1.1. Sistemas PACS

Os sistemas PACS têm o objetivo de facilitar a manipulação, armazenamento e transmissão de imagens médicas e relatórios estruturados em formato digital. Estes sistemas eliminam a necessidade de usar os antigos sistemas analógicos que eram completamente dependentes da intervenção humana para transferir, guardar e procurar a informação. Os sistemas analógicos eram, por isso, muito mais propensos a erros. O termo PACS diz respeito tanto a *hardware* como a *software*. Estes sistemas integram dispositivos de aquisição (respeitantes às modalidades já mencionadas anteriormente e outras), arquivos de imagem digital (onde as imagens digitais são armazenadas) e estações de trabalho (onde os radiologistas vêm as imagens) [5], conforme podemos ver na Figura 2.1. Além disto, os sistemas PACS contemplam ainda os mecanismos de transmissão e distribuição das imagens digitais entre os vários componentes, através da rede informática.

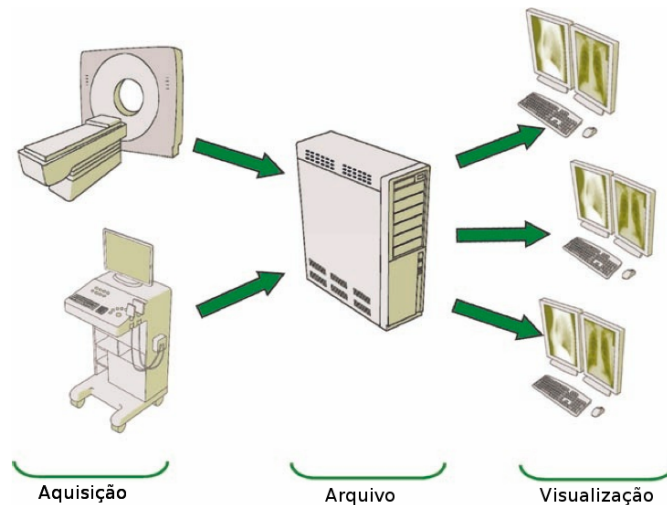


Figura 2.1: Componentes principais de um sistema PACS [5].

Os sistemas PACS baseiam-se na norma DICOM de forma a garantir a interoperabilidade entre componentes de diversos fabricantes [5]. Cada componente de um sistema PACS (*hardware* ou *software*) tem associada uma Declaração de Conformidade (*Conformance Statement*). A Declaração de Conformidade é um documento muito importante pois indica que partes da norma DICOM são suportadas [10]. Um dispositivo ou aplicação nunca implementa a totalidade da norma mas só uma (pequena) parte tendo em vista os serviços propostos.

Estes sistemas devem ter a capacidade de adquirir imagens provenientes de diferentes modalidades de imagens. Ao nível do armazenamento dessas imagens, os sistemas devem ser capazes de garantir a sua persistência durante longos períodos de tempo. Os fluxos de trabalho dos sistemas PACS variam conforme a instituição. O número e tipo de componentes do sistema PACS pode variar consoante as necessidades da instituição. A transmissão das imagens médicas digitais entre os vários componentes do sistema é um fator crítico que é tido em conta na construção e implementação destes sistemas, conforme iremos ver adiante neste documento.

Outros pontos cada vez mais importantes para os sistemas PACS são a sua integração com os sistemas RIS e HIS e a recente evolução que pretende aplicar estes sistemas à telemedicina ou mais concretamente à telerradiologia.

## 2.2. Norma DICOM

Até ao início dos anos 80, os sistemas de informação médicos de um fabricante que processavam imagens médicas digitais eram praticamente isolados dos sistemas de outros fabricantes. Eram raros os sistemas que conseguiam decodificar e utilizar imagens médicas digitais obtidas a partir de dispositivos de outro fabricante. Os departamentos de imagiologia, ao utilizar um sistema de informação para suportar o processo de aquisição e processamento de imagens médicas, ficavam “presos” a um fabricante e isto tornava o processo de digitalização da informação menos atrativo e mais arriscado.

Para colmatar este problema, foi formado um comité que envolvia a NEMA (*National Electrical Manufacturers Association*) e o ACR (*American College of Radiology*), ambos dos EUA, com o objetivo de normalizar os formatos e os protocolos de comunicação usados nos processos de imagiologia digital. Esta normalização beneficiaria toda a indústria da área pois facilitaria muito o uso de imagens médicas digitais e dos sistemas PACS em geral.

A primeira versão da norma desenvolvida chamava-se ACR-NEMA 300 e foi lançada em 1985. Desde então a conformidade dos sistemas com a norma passou a ser um fator muito importante para a comunidade médica. Esta primeira versão da norma foi fundamental para a expansão dos sistemas PACS mas rapidamente surgiram vários problemas relacionados com a mesma que precisavam de ser revistos. O comité continuou o seu trabalho para aperfeiçoamento e extensão da norma e, no ano 1988, foi lançada a segunda versão da mesma, mais robusta. Esta versão foi alvo de uma maior aceitação por parte dos fabricantes.

Devido à grande evolução que se deu ao nível dos protocolos e dispositivos de rede na altura e devido aos problemas detetados na segunda versão da norma, surgiu a necessidade de fazer uma nova revisão geral. Assim, a terceira versão foi lançada em 1993 e mudou de nome para DICOM 3.0, embora na maioria das vezes seja chamada apenas por DICOM. Esta versão dura até aos dias de hoje. Contudo, ela é revista anualmente, com acrescentos e correções de problemas detetados. Atualmente, devido às necessidades de evolução dos sistemas de informação, a norma está dividida 20 partes, sendo que algumas delas já não são utilizadas, e mais de centena e meia de suplementos.

A norma DICOM define o formato dos ficheiros, a estrutura dos diretórios usados para transferência *offline* e os protocolos de comunicação utilizados para transferir toda a informação relativa às imagens médicas através da rede informática.

### 2.2.1. Dicionário de dados

Embora tenha sido desenvolvida nos anos 80, a norma DICOM adota princípios que ainda são muito utilizados nos dias de hoje. O princípio de codificar toda a informação como sequências de elementos de dados, onde cada elemento pode ter um tamanho variável e é identificado por um nome único. Apesar das diferenças significativas entre os formatos, este princípio é verificado por exemplo no XML.

Como existe alguma variedade de dados clínicos (ex. datas, texto, números, identificadores), foi necessário encontrar formas de os codificar dentro dos elementos DICOM. A estas diferentes formas de codificação, a norma DICOM dá o nome de *Value Representations* (VRs). Existem 27 VRs diferentes definidos na parte 5 da norma DICOM [11].

A parte 6 da norma apresenta o Dicionário de Dados DICOM que contém uma lista de todos os elementos padrão. Adicionalmente a este dicionário padrão, os fabricantes DICOM podem usar dicionários privados para guardar os seus próprios atributos que não fazem parte da norma [5].

Para organizar esta lista de mais de 2000 elementos, estes foram divididos em grupos numerados (com base na similaridade dos seus conteúdos). Deste modo, cada elemento é identificado pelo seu número “(grupo, elemento)”, que também é chamado de “Tag”. Tanto os grupos como os elementos são numerados em base hexadecimal. Cada *Tag* tem associado um VR, ou seja um determinado elemento de dados é sempre codificado da mesma forma.

Os elementos dos objetos DICOM estão organizados em formato TLV (*Tag, Length, Value*), conforme a Figura 2.2. Cada elemento é constituído pela *Tag*, VR, tamanho do campo e o valor do campo codificado conforme o VR. O VR de um elemento pode ser implícito ou explícito. Isto é, pode estar declarado no próprio elemento e a aplicação que está a interpretar esse objeto DICOM pode descodificar o valor do elemento com base no VR declarado. No caso de ser um VR implícito, a aplicação terá que usar o VR definido por defeito na norma para essa *Tag*. Iremos ver mais adiante como é que a aplicação que descodifica os objetos DICOM pode saber se está perante VRs implícitos ou explícitos.

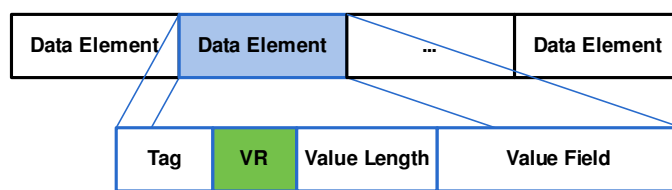


Figura 2.2: Formato TLV usado na norma DICOM.

### 2.2.2. Modelo de Informação

Antes de abordar mais detalhes da norma DICOM, convém dar uma noção básica da forma como a norma estrutura a informação. O DIM (*DICOM Information Model*) define a estrutura e organização da informação relativa à comunicação e arquivo de imagens médicas [12]. A Figura 2.3 contém uma representação das relações entre as principais estruturas do DIM.

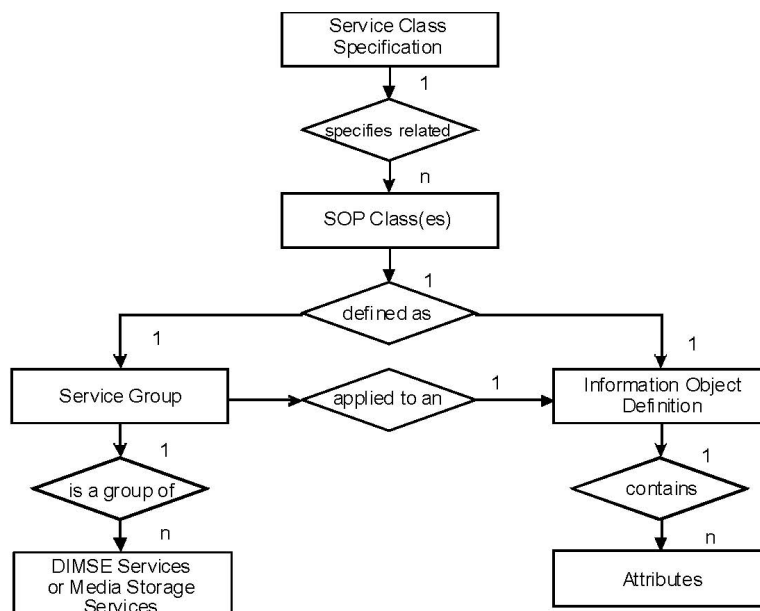


Figura 2.3: Principais estruturas do *DICOM Information Model* [12].

### Information Object Definition

Um IOD (*Information Object Definition*) é um modelo de dados que é usado para guardar informação de objetos do mundo-real. Tal como no paradigma de orientação a objetos, as características ou propriedades das entidades do mundo-real são mapeadas em atributos dos objetos, os atributos dos objetos DICOM são obtidos a partir das entidades que representam.

Um IOD não representa uma instância de um objeto do mundo-real mas sim uma classe de objetos que partilham os mesmos atributos. Um IOD que contenha apenas atributos obtidos a partir de uma entidade do mundo-real, é um IOD Normalizado (*Normalized IOD*). Por outro lado, um IOD que contenha atributos que não são obtidos a partir da entidade do mundo-real que representa mas que são obtidos a partir de entidades relacionadas, é um IOD Composto (*Composite IOD*) [12].

Para comunicação em rede, que será abordada com maior detalhe no subcapítulo 2.2.4, o DIM define serviços e conjuntos de serviços que especificam as operações ou notificações que se podem aplicar a cada IOD. Os IODs estão definidos na parte 3 da norma DICOM e os serviços e conjuntos de serviços na parte 4.

### Hierarquia de informação DICOM

Para organizar os sistemas de informação que lidam com a norma DICOM, existe a seguinte hierarquia: Paciente – Estudo – Série – Imagem (Figura 2.4). Esta hierarquia reflete o que acontece quando um paciente necessita de fazer exames imagiológicos. Quando um paciente chega ao hospital, podem ser marcados vários estudos (ex: CT, MR e XA). Cada estudo pode ter várias séries de imagens (ex: coronal, axial) e cada série tem uma ou mais imagens. Esta hierarquia permite organizar de forma lógica todas as imagens obtidas a partir de um paciente, assim como facilita a pesquisa da informação. Por exemplo: podemos obter todos os estudos CT do último ano para um determinado paciente.

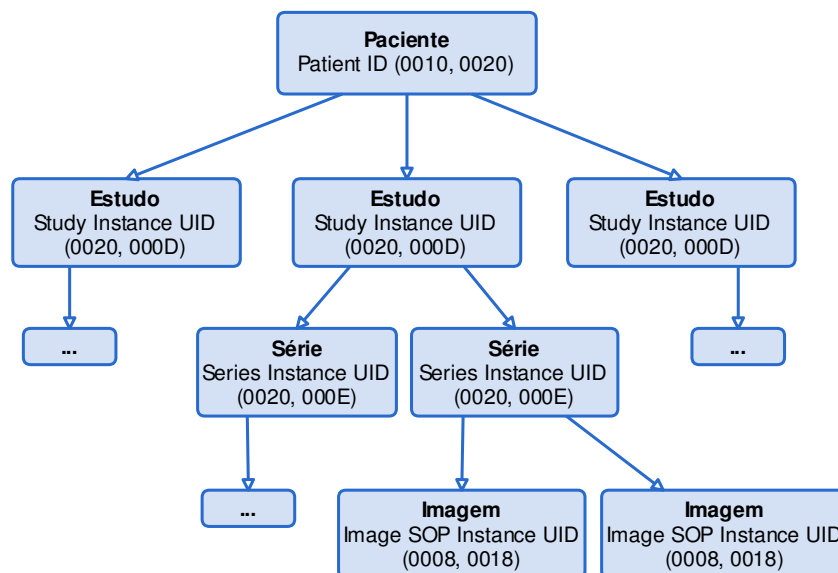


Figura 2.4: Quatro níveis da hierarquia de informação DICOM. Cada nível é identificado pelo seu elemento chave.

Cada um dos níveis da hierarquia tem um atributo que o identifica de forma única, isto é, uma espécie de chave primária. A Figura 2.4 apresenta esses atributos chave. O “Patient ID” identifica unicamente o paciente. Ao nível do exame (que também é chamado de estudo) é

usado o “Study Instance UID”. A tag “Series Instance UID” identifica a série e, por fim, a tag “Image SOP Instance UID” identifica a própria imagem.

### 2.2.3. Formato dos ficheiros

Uma imagem médica digital, além da informação relativa aos pixels da imagem, tem também associados meta-dados acerca dos pacientes, profissionais de saúde, instituição médica, dispositivo de aquisição, informação relativa ao exame e outros tipos de informação igualmente importante do ponto de vista clínico (como por exemplo a dose de radiação aplicada ao paciente ou o tempo de exposição).

Existe a possibilidade de guardar essa informação das imagem médica em ficheiros de forma normalizada. Isto é, apesar de ser possível transferir essa informação entre as várias entidades do sistema através da rede informática (conforme iremos ver no subcapítulo seguinte), existe também a possibilidade de a transferir e guardar manualmente da mesma forma que transferimos e guardamos os ficheiros no dia-a-dia. Isto para além de as próprias entidades de arquivo dos sistemas PACS recorrerem, na maior parte das vezes, à persistência das imagens DICOM na forma de ficheiros.

Um ficheiro DICOM contém um preâmbulo de 128 bytes seguido dos caracteres “DICM”, a meta-informação sobre o ficheiro e a imagem nele contida e, por fim, a informação relativa aos pixels da imagem digital, conforme podemos ver na Figura 2.5. A zona de meta-informação contém um conjunto variável de campos de acordo com a modalidade do estudo a que a imagem diz respeito, contudo alguns campos têm obrigatoriamente que estar presentes em todos os ficheiros DICOM.

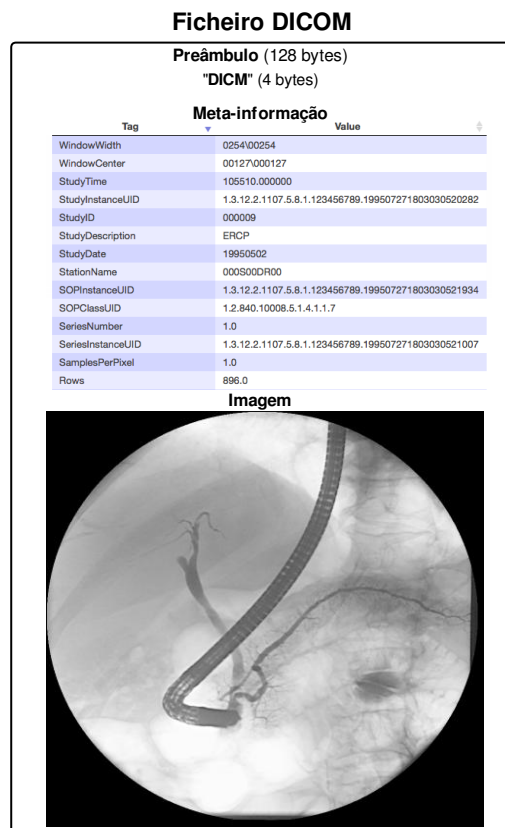


Figura 2.5: Estrutura conceptual de um ficheiro DICOM. No cabeçalho ilustra alguns dos elementos de informação contidos no ficheiro.

Cada ficheiro contém um objeto DICOM. Os ficheiros DICOM seguem o formato TLV apresentado anteriormente. O primeiro grupo de meta informação do ficheiro (grupo 0002) tem VR explícito. Este grupo contém alguma informação sobre o conteúdo do ficheiro, tal como a modalidade da imagem, o identificador da imagem (*Image SOP Instance UID*) e um elemento chamado “Transfer Syntax UID” (0002, 0010) que identifica se a codificação dos restantes grupos de elementos tem VR implícito ou explícito.

Um codificador ou decodificador de objetos DICOM (ficheiros ou mensagens na rede) precisa de conhecer o IOD que corresponde ao objeto que pretende processar. Tal como foi dito em 2.2.2, um IOD representa uma classe de objetos que partilha os mesmos atributos. Assim, apenas conhecendo o IOD correspondente ao tipo de objeto que se pretende codificar/decodificar é que é possível a uma aplicação saber que elementos existem, quais são obrigatórios e quais são opcionais. É importante referir que os IODs também estão organizados de forma hierárquica e um determinado IOD pode ser constituído por vários outros IODs. Por exemplo, o IOD “Patient Information Object Definition” é utilizado nos IODs das várias modalidades.

#### 2.2.4. Protocolo de Rede

Os sistemas de informação médicos não são aplicações e dispositivos que funcionam isoladamente. Existe uma grande necessidade de comunicação entre todas as aplicações e dispositivos que fazem parte dos sistemas de informação para ser possível trocar dados e executar ações dentro do sistema. É por isso que a norma DICOM também define o protocolo que deve ser usado para transmissão e execução remota de comandos em sistemas PACS.

O protocolo de comunicações definido na norma é um protocolo da camada de aplicação que funciona sobre o protocolo TCP/IP (Transmission Control Protocol / Internet Protocol). Assim, o protocolo DICOM tira partido de todas as vantagens que o TCP oferece, nomeadamente: gestão da conexão, ordenação de pacotes, controlo de fluxo, notificação de receção e retransmissão automática em caso de perda de pacotes ou erro de transmissão.

No TCP/IP a comunicação é feita ponto-a-ponto e cada um dos intervenientes na comunicação é identificado pelo par (IP, porta). Ao nível DICOM, os intervenientes na comunicação chamam-se *Application Entities* (AE). As AEs são dispositivos ou aplicações que “falam” DICOM como por exemplo: uma aplicação de uma estação de trabalho para visualizar imagens médicas, um servidor de arquivo de imagens ou um dispositivo de aquisição de imagens (modalidade). Cada uma destas entidades tem associado um *Application Entity Title* (AETitle) que é uma *string* que pode ter até 16 caracteres e que identifica a entidade DICOM. Desta forma as AEs podem ser endereçadas através do seu AETitle pois cada AETitle irá corresponder a um par (IP, porta).

A porta por defeito do protocolo de rede DICOM é a porta 104 mas qualquer porta livre pode ser usada, dependendo da configuração das aplicações e dispositivos. Isto permite, por exemplo, que um computador com várias aplicações DICOM possa executá-las todas em simultâneo. Cada aplicação terá, obviamente, um AETitle diferente.

O protocolo de rede DICOM encontra-se dividido em 2 níveis. Ao nível mais baixo temos o *DICOM Upper Layer Protocol* (DICOM UL), que define o protocolo de gestão das associações entre as entidades na rede e o formato dos pacotes que será usado para transmitir os objetos DICOM. A alto-nível temos os *DICOM Message Service Elements* (DIMSE) que corre sobre o DICOM UL e define os serviços e comandos que podem ser prestados/usados pelas entidades de rede. Estes serviços e comandos de alto-nível são construídos sobre o DICOM UL.

## **DICOM UL**

Este protocolo de baixo nível foi definido para criar um mecanismo de associação entre quaisquer duas entidades/aplicações DICOM (AEs) de forma a que elas possam comunicar e transferir informação imagiológica entre si. Para isso, o DICOM UL define o formato dos pacotes e também a ordem pela qual eles devem ser enviados [5].

Do ponto de vista da comunicação TCP/IP, o DICOM UL pode ser entendido como uma extensão ao TCP. O TCP é um protocolo da camada de transporte que lida com o estabelecimento e gestão de sessões ponto-a-ponto. As associações entre duas AEs são estabelecidas após uma fase de negociação de forma a que as AEs saibam as funcionalidades uma da outra, incluindo os formatos e serviços suportados.

O estabelecimento das sessões DICOM é feito antes de qualquer transação de informação entre as AEs. Acontece imediatamente após o famoso *TCP handshake* que estabelece as sessões TCP. Durante este estabelecimento da sessão DICOM, as AEs trocam informação sobre as suas funcionalidades, acordam sobre o tipo de informação que vai ser transferida e os parâmetros que vão usar para a sua comunicação daí em diante, como por exemplo: se vão declarar os VRs implícita ou explicitamente e a codificação que irá ser utilizada na transferência.

Todas as comunicações DICOM são feitas sobre um conjunto de estruturas de informação que definem o formato dos pacotes transferidos na rede. Estas estruturas são normalmente chamadas “*Protocol Data Units*” (PDUs). Existem 7 tipos de PDUs definidos na norma:

1. **A-Associate-RQ:** Faz um pedido de associação
2. **A-Associate-AC:** Aceita um pedido de associação (responde a um A-Associate-RQ)
3. **A-Associate-RJ:** Rejeita um pedido de associação (responde a um A-Associate-RQ)
4. **P-Data-TF:** Transfere blocos de informação DICOM
5. **A-Release-RQ:** Faz um pedido de término de associação
6. **A-Release-RP:** Responde a um pedido de término de associação
7. **A-Abort:** Aborta uma associação em qualquer altura da sessão

A estrutura e informação contida em cada um destes PDUs poderia ser detalhada, pois foi alvo de estudo e trabalho no desenvolvimento do sensor de rede, que será apresentado no capítulo 3. Apesar disso, a estrutura dos PDUs não será detalhada neste documento pois não se pretende fazer uma exposição exaustiva, byte a byte, do protocolo de rede pois tal resultaria numa narrativa muito técnica e densa.

A Figura 2.6 apresenta a comunicação de PDUs entre duas AEs. No caso de haver uma rejeição (A-Associate-RJ) ou falha (A-Abort) durante a sessão, a associação é interrompida imediatamente e não será transmitido mais nenhum PDU entre as duas AEs nessa sessão de comunicação.



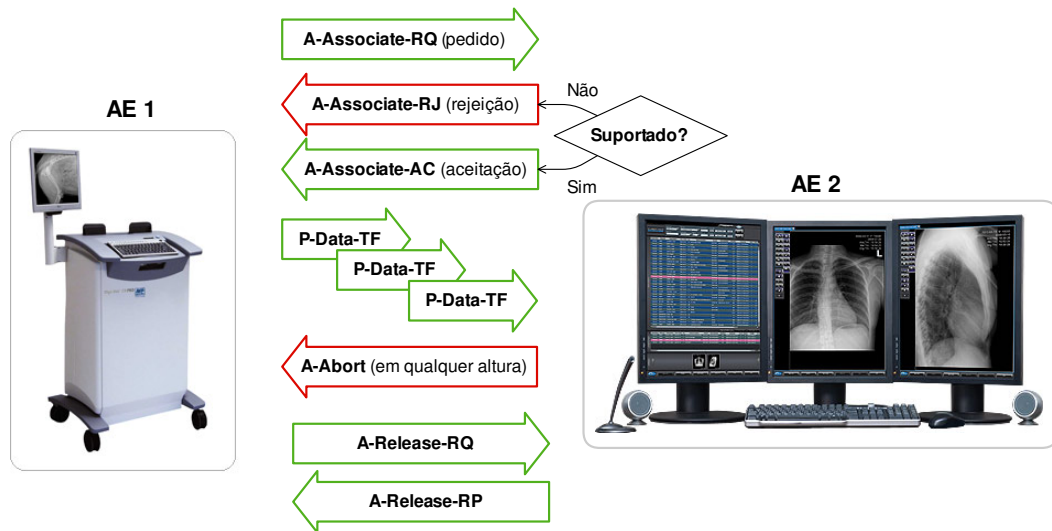


Figura 2.6: Fluxo normal dos DICOM PDUs. As setas vermelhas correspondem a falhas e as verdes correspondem ao fluxo de associações bem sucedidas (adaptado de [5]).

### Serviços DIMSE

As DICOM AEs fornecem serviços umas às outras usando uma lógica cliente/servidor. Na norma DICOM, a uma AE que utilize serviços de outra é dado o nome de "Service Class User" (SCU). Por sua vez, uma AE que forneça um serviço é um "Service Class Provider" (SCP). Por exemplo, um dispositivo de aquisição produz imagens de uma modalidade (SCU) e envia-as para um servidor (SCP) que fornece um serviço de armazenamento. Uma única entidade/aplicação DICOM pode assumir o papel de SCU e SCP (Figura 2.7) em diferentes associações.

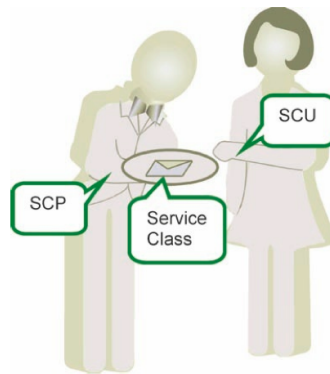


Figura 2.7: Analogia do modelo SCU-SCP [5].

Cada um dos serviços está associado com um ou mais comandos DIMSE (Tabela 2.1). A Tabela 2.1 apresenta os principais serviços DICOM e os respectivos comandos.

Os comandos DIMSE são enviados dentro de PDUs P-Data-TF, depois de negociada e estabelecida a associação entre as duas AEs. A maioria dos comandos usam 2 PDUs. O primeiro com a identificação, estado, ID da mensagem e o segundo com um IOD relativo ao comando contendo os dados que se pretendem processar na outra AE (por exemplo os parâmetros de uma pesquisa). Uma *SOP Class* é definida pela união de um IOD com um conjunto de serviços DIMSE. A identificação das *SOP Classes* é usada pelas aplicações DICOM (AEs) para estabelecer e acordar um conjunto que funcionalidades que poderão ser usadas durante a sua interação [12].

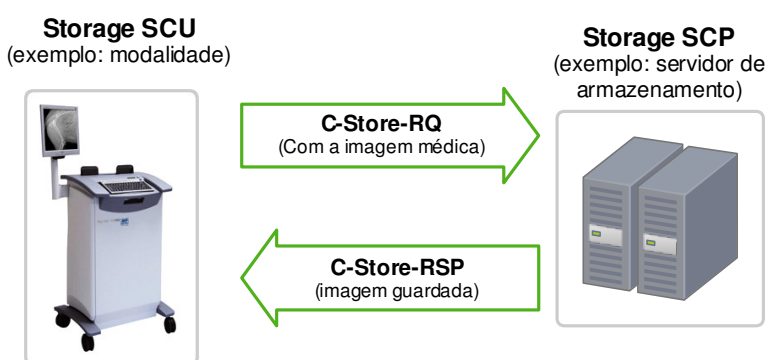
O serviço de verificação permite a um SCU verificar se consegue estabelecer comunicação com um SCP. Permite verificar se um SCP está a funcionar corretamente.

**Tabela 2.1: Principais serviços DIMSE.**

<b>Serviço</b>	<b>Descrição</b>	<b>Comandos DIMSE</b>
Verification	“Ping” do DICOM. Verifica o estado de um AE.	C-ECHO
Storage	Envio de imagens (“Push”) para o repositório.	C-STORE
Query/Retrieve	Pesquisa no repositório, Move/Get de imagens.	C-FIND, C-MOVE, C-GET
Worklist Management	Obter a lista de trabalho.	C-FIND
Basic Print	Impressão de imagens.	N-CREATE, N-SET, N-DELETE, N-ACTION
Storage Commitment	Pedido de salvaguarda de dados.	N-ACTION, N-EVENT-REPORT
Modality Performed Procedure Step (MPPS)	Envio de informação do exame para o RIS.	N-CREATE, N-SET

O serviço de armazenamento permite que um SCU guarde imagens (e outros tipos de dados DICOM) num servidor de armazenamento, usando o comando C-STORE (Figura 2.8). Uma vez que os diferentes tipos de objetos DICOM (IODs diferentes) implicam diferentes tipos de processamento, a norma DICOM atribui uma *SOP Class* a cada modalidade ou tipo de dados (por exemplo: *CT Image Storage SOP*, *MR Image Storage SOP* e *Ultrasound Image Storage SOP*). No serviço de armazenamento, fora o tipo de *SOP Class* ser específico para cada modalidade tendo todos um UID diferente, tudo o resto é igual para todas as modalidades.

A Figura 2.8 apresenta o processo de transferência de apenas uma imagem. Para cada imagem que se pretende transferir, o SCU envia uma mensagem C-Store-RQ contendo o objeto DICOM que pretende transferir. Depois de receber o objeto, o *Storage SCP* responde com uma mensagem C-Store-RSP indicando se a transferência teve sucesso ou não.



**Figura 2.8: Serviço de armazenamento DICOM (C-STORE).**

O serviço de *Query/Retrieve* faz uso de dois comandos. O comando C-FIND permite que um SCU (ex: uma estação de trabalho) envie uma *query* para um servidor de armazenamento

(SCP) e obtenha as respostas correspondentes à *query* enviada (Figura 2.9). O segundo comando usado no fluxo normal do serviço pode ser um C-GET ou um C-MOVE.

A Figura 2.9 apresenta o fluxo normal de um comando C-FIND. A pesquisa efectuada procura saber todos os estudos da modalidade CR cujo nome do paciente comece pela letra T. No caso apresentado são retornados apenas 2 estudos (Tania e Tiago). Note-se que é enviada uma mensagem C-FIND-RSP para cada um dos resultados obtidos e uma última mensagem sem nenhum resultado para confirmar que a pesquisa foi terminada com sucesso.

É importante referir que as pesquisas com o comando C-FIND podem ser feitas a qualquer nível da hierarquia que vimos na Figura 2.4, isto é, ao nível do Paciente, Estudo, Série ou imagem. Um dos parâmetros da pesquisa é exatamente o nível da hierarquia ao qual a pesquisa deve ser feita.

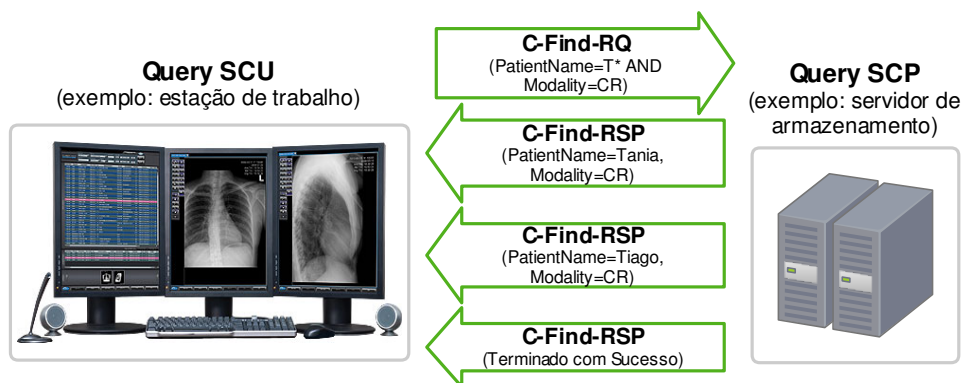


Figura 2.9: Serviço de pesquisa DICOM (C-FIND).

O segundo comando usado no serviço de *Query/Retrieve* é usado para transferir as imagens médicas de uma AE para outra. Normalmente são imagens referenciadas nos resultados da pesquisa efectuada anteriormente, embora isto não seja obrigatório.

A Figura 2.10 apresenta o fluxo do comando C-GET. Neste caso a estação de trabalho pretende transferir todas as imagens do paciente com o ID "123456" e são retornadas duas imagens. Note-se que as imagens são transferidas usando o comando C-STORE, que vimos anteriormente. A transferência das imagens com C-STORE pode ser entendido como uma suboperação do C-GET.

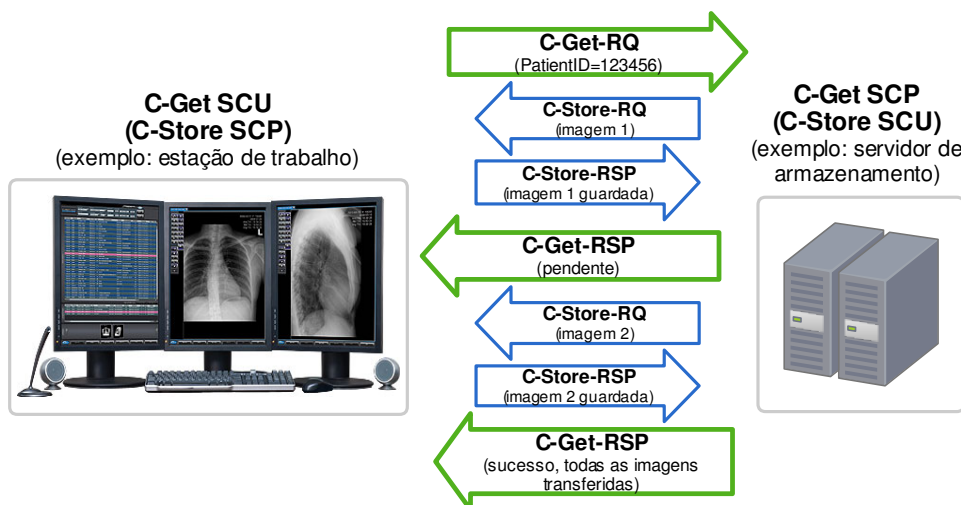


Figura 2.10: Serviço de recuperação DICOM (C-GET).

Após cada imagem transferida, é enviada uma mensagem C-Get-RSP para indicar se vão ser transferidas mais imagens ou se terminou o processo. O comando C-MOVE é muito semelhante ao comando C-GET, tendo apenas duas diferenças. A primeira é: a mensagem inicial de *request* do C-MOVE contém um parâmetro extra que indica ao C-Move SCP qual é o AETitle da AE para onde deve transferir as imagens. Assim, se o AETitle for conhecido pelo C-Move SCP, as imagens serão reencaminhadas para a terceira AE. A segunda diferença é que o C-MOVE exige sempre o estabelecimento de duas associações DICOM, mesmo que o AETitle de destino seja o da AE que fez o pedido. Ou seja, no caso do C-GET, todas as mensagens que vemos na imagem usam apenas uma associação. Se fosse um C-MOVE as mensagens a verde seriam transferidas na primeira associação e as mensagens a azul seriam transferidas numa associação estabelecida posteriormente.

O serviço de *Worklist Management* possibilita ao SCU obter a lista de trabalho que lhe foi atribuída. Por exemplo a lista de estudos que foram marcados para esse dia. Este serviço funciona com base em comandos C-FIND.

Os restantes serviços da Tabela 2.1 não serão abordados aqui pois não foram utilizados no trabalho desenvolvido.

### 2.3. HL7

Ao falar de sistemas de informação médicos e nas normas mais utilizadas, é necessário fazer referência às normas desenvolvida pelo HL7 (*Health Level Seven*). O HL7 é uma organização internacional, fundada em 1987 e sem fins lucrativos, que se dedica a desenvolver um conjunto de normas para intercâmbio, integração, partilha e transferência de informações médicas que suportam a prática, gestão, entrega e avaliação de serviços de saúde [13]. O nome “*Level Seven*” é uma referência à sétima camada do modelo de comunicações OSI, que também é chamada de “camada de aplicação”. Este nome indica portanto que o HL7 se foca em protocolos da camada de aplicação para o cenário médico.

O principal objetivo ao criar várias normas para serem utilizadas em ambiente médico em todas as áreas mencionada é que elas sejam usadas pela vasta maioria dos sistemas médicos de forma a promover a interoperabilidade entre sistemas de diferentes fabricantes, sem ambiguidades entre os vários sistemas.

Embora as normas e protocolos definidos pelo HL7 sejam muito interessantes para usar em ambientes médicos, possuem uma diferença muito significativa em relação à norma DICOM, apresentada anteriormente. Os standards do HL7 não são standards abertos, isto é, todo o “material” do HL7 está protegido por leis de *copyright* e propriedade intelectual [13]. Na prática, isto quer dizer que para ter acesso a todos os detalhes das normas e mesmo para desenvolver aplicações que os usem, é necessário ser membro da organização.

### 2.4. Ecocardiografia

A ecocardiografia é uma das várias modalidades de imagiologia que são usadas para estudar o coração. A imagiologia cardíaca é hoje de grande importância ao nível de diagnóstico, planeamento e tratamento de patologias. Como caso de estudo do sistema apresentado, foi desenvolvido um sensor especializado para obter informação a partir de um repositório de relatórios de ecocardiogramas.

A ecocardiografia usa técnicas de ultrassonografia para obter imagens do coração. Tradicionalmente são imagens bidimensionais mas, devido aos avanços científicos, existem

hoje técnicas de ultrassonografia 3D em tempo-real (também denominada por ultrassonografia 4D) [14].

Os princípios físicos usados na ecocardiografia vêm dos estudos feitos às propriedades das ondas sonoras. Em 1877, Jobin Strutt publicou “A Teoria do Som” onde eram apresentados os princípios dos ultrassons. As ondas sonoras utilizadas na ecocardiografia podem variar entre os 2 e os 15 MHz e são, por isso, imperceptíveis ao ouvido humano (o limite superior da audição humana é aproximadamente 20 KHz) [14]. Como o som é uma onda mecânica não se propaga no vácuo, ao contrário do que acontece com as ondas electromagnéticas. É necessária a existência de um meio físico para permitir a propagação das ondas sonoras. A velocidade dessa propagação varia conforme a inércia e elasticidade do meio onde está a ser propagada.

As ondas ultrassônicas não são transmitidas com 100% de eficiência através dos tecidos do corpo humano. A reflexão, difusão e absorção são alguns dos fatores que provocam atenuação das ondas transmitidas. Os ultrassons podem ter diferentes utilidades no campo da medicina, desde a terapêutica ao diagnóstico através da obtenção de imagens que são geradas a partir da reflexão sonora [15]. Estes ecos sonoros são provocados pelas diferenças de impedância acústica entre os diferentes tecidos do corpo humano.

A frequência de emissão dos ultrassons interfere com o nível de detalhe (resolução) que é possível produzir nas imagens e o poder de penetração desse mesmo feixe. Quanto maior for a frequência, mais curto é o comprimento de onda e maior é a energia do som [15]. Frequências mais altas são utilizadas para obter maior detalhe de imagens mas limitam o poder de visualização aos órgãos superficiais. Frequências mais baixas não permitem alcançar tanto nível de detalhe mas permitem analisar órgãos a maior profundidade.

Os equipamentos de ultrassonografia que suportam frequências mais elevadas requerem maior precisão e poder de processamento e, por isso, tendem a ser mais dispendiosos. A frequência usada para realizar ecocardiogramas de rotina em adultos utiliza frequências a rondar os 3 MHz pois o seu coração está localizado mais profundamente no tórax. Em crianças, as frequências utilizadas são normalmente superiores.

Um ecocardiograma é um exame complementar de diagnóstico que usa ultrassons para obter imagens do coração e dos vasos sanguíneos que lhe estão próximos. O aparelho que realiza ecocardiogramas chama-se ecógrafo ou ecodardiógrafo (Figura 2.11). Este aparelho contém um transdutor que emite e regista os sinais ultrassónicos. O aparelho consegue saber a distância a que estão as fronteiras que refletem os sinais sonoros medindo o tempo de viagem entre a onda sonora emitida e o seu respetivo eco.



**Figura 2.11: Ecógrafo moderno da Siemens.**

Tipicamente os ecocardiogramas são sequências de imagens bidimensionais com uma forma aproximadamente triangular, dando uma aparência de movimento em tempo real (Figura 2.12). Existem vários tipos de ecocardiogramas [16]:

- **Ecocardiograma transtorácico em repouso (ETT):** É o tipo mais comum de ecocardiogramas. Permite obter informação sobre o miocárdio, válvulas cardíacas e o fluxo de sangue no interior das cavidades do coração. Neste exame, o transdutor é colocado no peito do paciente em direção ao coração e, dependendo da posição do transdutor, podem ser obtidos diferentes tipos de imagens.
- **Ecocardiograma transesofágico (ETE):** Neste tipo de ecocardiograma é utilizada uma pequena sonda com transdutor de frequências mais elevadas. Esta pequena sonda é colocada no esôfago do paciente, imediatamente atrás do coração. Como a sonda é colocada perto do coração, a interferência é muito menor e pode ser utilizada uma frequência mais elevada, as imagens irão ter maior detalhe do que no ecocardiograma transtorácico.
- **Ecocardiograma de sobrecarga farmacológica (stress):** É um ecocardiograma em que são administrados medicamentos durante o exame que provocam o esforço cardíaco.
- **Ecocardiograma de esforço:** É um tipo de ecocardiograma semelhante ao ecocardiograma transtorácico mas que é realizado durante uma prova de esforço do paciente.

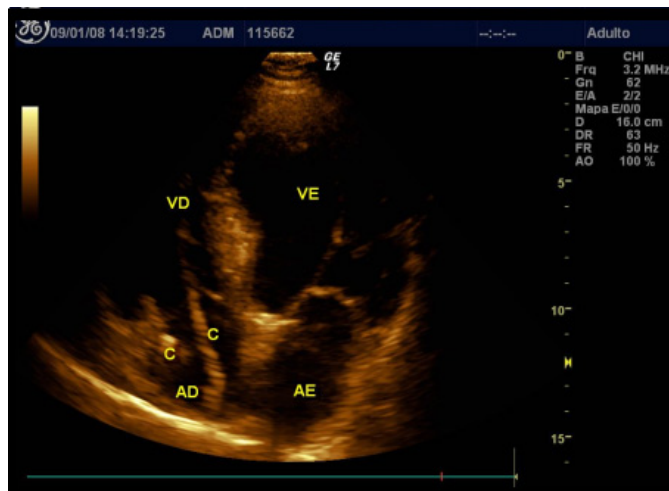


Figura 2.12: Exemplo de uma imagem de um ecocardiograma transtorácico [17].

O efeito de Doppler é verificado quando os objetos que refletem as ondas sonoras se encontram em movimento. Ao contrário do caso em que os objetos se encontram parados, quando se encontram em movimento, a frequência da onda sonora refletiva vai ser diferente da frequência da onda sonora inicial [15].

$$F_R = F_0 + \Delta F$$

A diferença de frequência  $\Delta F$  vai ser positiva se o objecto refletor se estiver a aproximar da fonte e negativa caso se esteja a afastar. A equação seguinte permite relacionar a diferença de frequência com a velocidade do objeto [15]:

$$\Delta F = \frac{2v}{c} \cos \theta$$

onde:

$v$  é a velocidade do objeto em movimento

$c$  é a velocidade de propagação dos ultrassons nos tecidos (constante 1540 m/s)

$\theta$  é o ângulo entre o eixo do feixe e o eixo sobre o qual o objeto se desloca

Na prática médica, o Doppler pode ser utilizado para detetar o movimento da corrente sanguínea. Uma cor indica que o sangue se move na direção da sonda e outra cor indica que ele se afasta da sonda. Esta técnica de análise do efeito de Doppler é útil na ecocardiografia porque é eficaz na deteção de fluxos sanguíneos anormais que podem ser provocados por lesões nas válvulas cardíacas. Dada a sua importância, esta função está hoje disponível em grande parte dos ecógrafos [15].

As imagens médicas obtidas com recurso a ecógrafos são hoje em dia uma das modalidades (US) de imagens médicas digitais mais utilizadas, cuja informação produzida e tem sido arquivada com recurso aos sistemas PACS e à norma DICOM, falados anteriormente.

## 2.5. Dicoogle

O Dicoogle é uma aplicação *open-source* desenvolvida no grupo de bioinformática da Universidade de Aveiro. O Dicoogle é um servidor de armazenamento imagens médicas DICOM que, além das funcionalidades típicas deste tipo de componentes do sistema PACS, dá aos utilizadores liberdade para pesquisar sobre todos os campos de meta informação contida nas imagens médicas digitais. Os componentes tradicionais restringem os campos passíveis de pesquisas a um conjunto limitado. É uma aplicação de fácil instalação e está disponível para *download* livremente, conforme se pode ver em [7].

Esta aplicação já foi tema de várias publicações científicas e artigos de opinião devido às potencialidades e aspetos distintivos que possui. Este servidor de armazenamento funciona em múltiplas plataformas, utiliza um motor de indexação, chamado Lucene [18,19] para indexar os conteúdos das imagens DICOM, providencia vários serviços DICOM (Armazenamento, Query/Retrieve) e mecanismos de comunicação *peer-to-peer* para pesquisar em outros nós (instâncias do Dicoogle) tanto numa rede local como na WAN [20–22].

Na Figura 2.13 podemos ver a sua arquitetura de componentes de alto-nível. Conforme apresentado, o Dicoogle tem uma arquitetura modular baseada em plugins. Trata-se de uma ferramenta importante para cumprir os objetivos desta dissertação pois permite indexar todo o repositório de imagens médicas existente numa instituição médica. Depois da indexação, esse repositório pode ser analisado através de *web services* REST. A escolha do Dicoogle como sensor do repositório de imagens médicas foi tomada graças à sua flexibilidade ao permitir indexar todos os campos existentes nas imagens médicas [20] e ao facto de ser *open-source* [20–22] permitindo assim adicionar métodos aos *web services* já existentes e também à familiaridade que tinha com a aplicação.

Os *web services* disponibilizados pelo Dicoogle devolvem mensagens em formato XML e permitem pesquisar imagens, ver todos o conteúdo de todos os campos indexados de uma determinada imagem e fazer *download* da própria imagem. No âmbito deste trabalho foi necessário implementar alguns métodos adicionais aos *web services* disponibilizados. No subcapítulo 3.5 serão detalhados os métodos existentes, bem como os novos métodos desenvolvidos.

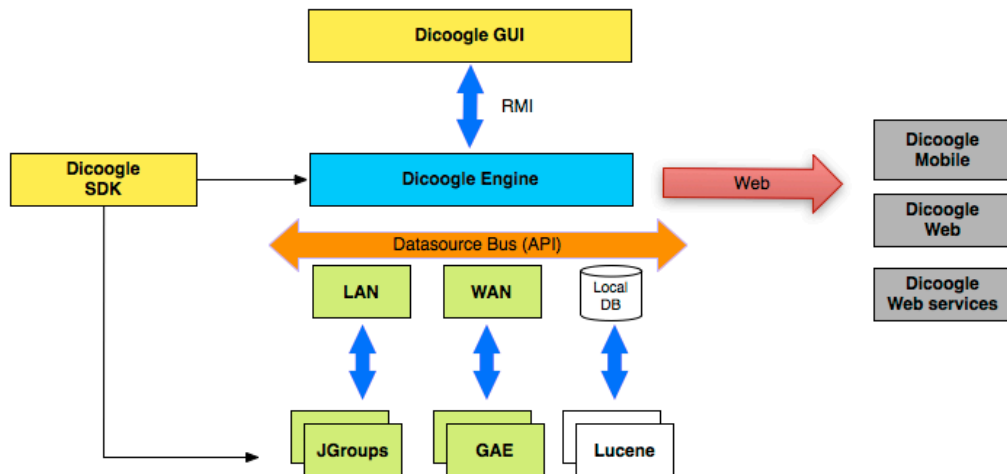


Figura 2.13: Arquitetura de alto-nível do Dicoogle [7].

## 2.6. Monitorização de fluxos de trabalho em instituições médicas

Monitorizar os fluxos de trabalho de uma instituição médica e extrair conhecimento a partir da sua informação digital, não é uma tendência nova. Muitas vezes, os fabricantes de sistemas PACS oferecem soluções que permitem fazer grande parte das coisas mencionadas antes. Contudo, na maioria dos casos, estas soluções são proprietárias. Isto significa que estas soluções oferecidas não são interoperáveis com os componentes de sistemas de outras empresas.

Existe uma solução para analisar o desempenho de sistemas PACS, chamada PACSPulse [23]. É uma ferramenta *open-source* que fornece uma interface de utilizador baseada em páginas Web. Esta ferramenta permite aos utilizadores detetar os pontos de congestionamento da rede do PACS. A principal métrica de desempenho usada por esta ferramenta a taxa de transferência dos estudos, de e para cada componente do sistema PACS. Esta métrica pode ser muito útil para o administrador de rede pois permite-lhe saber quais os dispositivos, ou ligações de rede, que estão sobrecarregados. É indiscutivelmente uma ferramenta útil no processo de gestão e dimensionamento de uma rede PACS já em ambiente de produção.

Para recolher as métricas, o PACSPulse usa os *Logs* de utilização dos componentes do sistema PACS. Estes *Logs* são transferidos a cada duas horas, via FTP (File Transfer Protocol), para um motor de *parsing*. Por fim, a informação processada fica disponível para os utilizadores consultarem na aplicação Web. O sistema PACSPulse não é um sistema tempo-real, uma vez que recolhe e processa a informação periodicamente. Este sistema apenas está disponível para componentes do sistema PACS que gerem *Logs* de utilização. Para incorporar novos formatos de *Log* no sistema, é necessário escrever um script PERL (*Practical Extraction and Report Language*) para fazer *parse* da informação contida no *Log*.

Também existe um sistema gráfico de *Dashboards* que tem como objetivo ajudar os administradores dos departamentos de imagiologia nos processos de inteligência empresarial (*business intelligence*) [24]. Este sistema também apresenta uma interface de utilizador baseada em páginas Web. Ao contrário do sistema apresentado anteriormente, este sistema fornece indicadores de desempenho mais ligados a inteligência empresarial (ex. percentagem de pacientes que esperam mais que uma hora para que o exame termine, taxa de repetição dos exames, qualidade das imagens médicas e assim por diante). Este sistema obtém informação a partir de um componente que recebe informação de protocolos do HL7



(*Health Level Seven*), a partir de um arquivo DICOM que transmite e recebe mensagens DICOM MPPS (*Modality Performed Procedure Step*), de um servidor RIS e de outras ferramentas internas utilizadas naquela instituição.

No artigo [25] é apresentado um sistema chamado "*Imaging Exam Time Monitor*". Este também é um sistema com interface Web que fornece aos utilizadores 5 métricas de eficiência que são obtidas diretamente a partir dos campos presentes nas imagens DICOM. Estas métricas são: a duração média dos exames, taxa de utilização dos aparelhos médicos, tempo entre-pacientes, intervalo de tempo inter-séries. O sistema é capaz de gerar relatórios sobre cada uma das métricas e enviar notificações por e-mail ou SMS para alertar sobre a inatividade de um aparelho por um determinado período de tempo. O artigo não especifica o mecanismo de transmissão que é usado para transmitir os ficheiros em formato DICOM do sistema PACS para o analisador usado pelo *Imaging Exam Time Monitor*. Os dois principais objetivos deste sistema são: demonstrar o valor da informação que está acessível no cabeçalho das imagens médicas em formato DICOM e propor um conjunto de métricas para avaliação da produtividade ao nível dos aparelhos de imagiologia [25].

Em [26] é apresentado um sistema capaz de analisar os campos das imagens DICOM que estão relacionadas com a dose de radiação utilizada no exame. Este sistema lida com campos proprietários nas imagens DICOM, usando uma base de dados que guarda as relações entre os dispositivos PACS e os campos proprietários usados por cada um. Se a dose de radiação exceder um determinado limiar ou se existir um novo dispositivo PACS na rede que não foi registado no sistema, é enviado um alerta para uma lista de e-mails, para telemóveis por SMS e para os pagers internos dos responsáveis pelo sistema. Este sistema também permite gerar relatórios parametrizáveis através de uma interface Web. O sistema analisa os dados recolhidos periodicamente não sendo assim um sistema de tempo-real. Este sistema pode ser uma ferramenta valiosa para assegurar a qualidade de serviço, relativamente às doses de radiação aplicadas, num departamento de imagiologia.

Além dos sistemas apresentados anteriormente existem ainda outros sistemas que nos permitem analisar redes de imagem médica a um nível mais baixo. O *DICOMETRIX* [27] é uma ferramenta que permite obter algumas métricas relacionadas com o desempenho da rede de imagens médicas. A Figura 2.14 apresenta a sua janela principal. Nesta aplicação temos acesso a duas métricas: Imagens médicas Por Segundo (IPS) e taxa de transmissão em KB/s para um determinado computador na rede médica. Existe uma versão servidor que permite agregar no servidor a informação de todos os computadores que estão a executar a aplicação.



Figura 2.14: Janela principal do DICOMETRIX [27].

O DVtk DICOM Network Analyzer [28], o DICOM Detective [29] (Figura 2.15) e o Wireshark com o plugin do protocolo DICOM [30] (Figura 2.16) são ferramentas que nos permitem analisar os pacotes DICOM e associações DICOM que circulam na rede de imagens médicas.

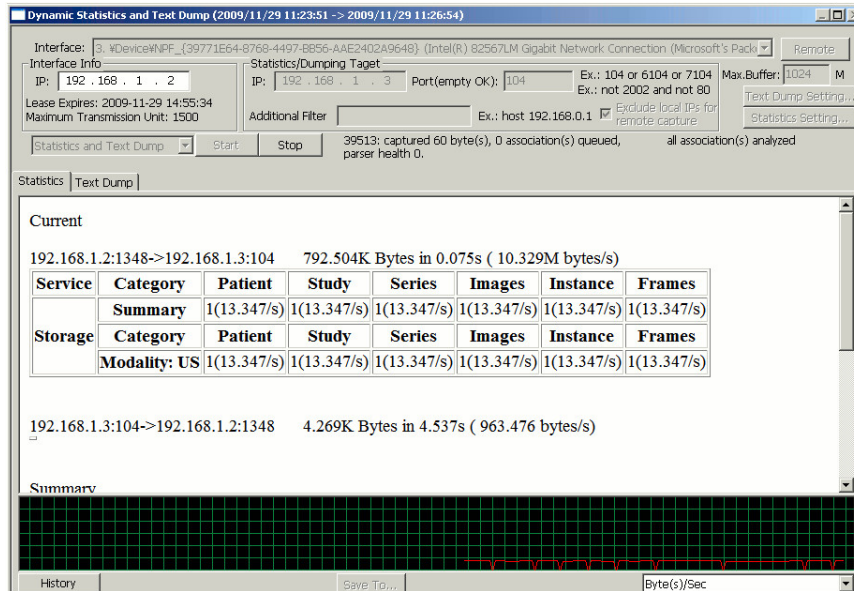


Figura 2.15: Janela principal do DICOM Detective [29].

Estas ferramentas são boas para “chegar aos bytes” das mensagens DICOM, ou seja, ver o conteúdo exato das mensagens e até descobrir a origem de problemas de conectividade. Ainda assim, elas não podem fazer muito para ajudar os profissionais da área a fazer uma análise de alto-nível sobre o que acontece na rede de imagens médicas.

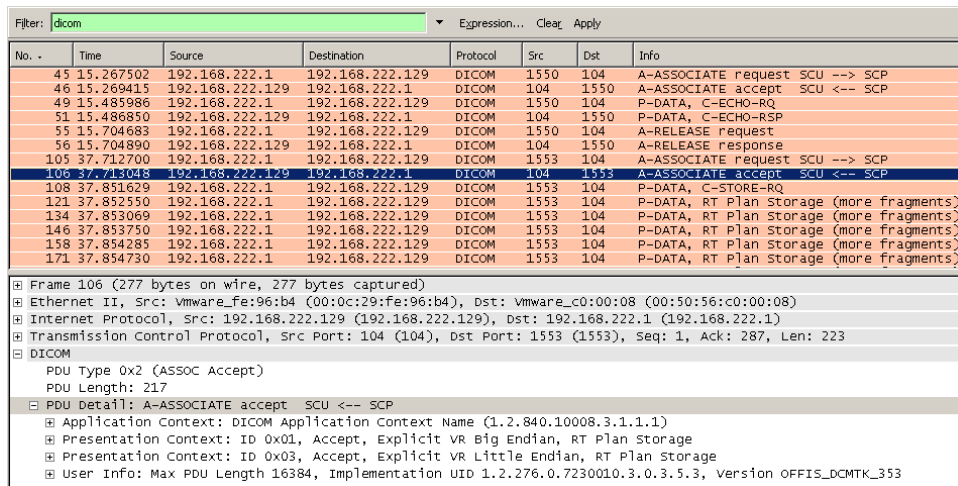


Figura 2.16: Wireshark a apresentar os pacotes DICOM [30].

## 3 Conceptualização da plataforma

No capítulo anterior foi apresentado o panorama dos sistemas de informação usados na saúde, detalhando a área da imagiologia médica que é o foco desta dissertação. Foram apresentados os principais protocolos envolvidos e alguns sistemas usados para analisar a atividade de departamentos de imagiologia em instituições médicas. Esses conceitos são importantes para entender os objetivos e trabalho realizado nesta dissertação.

### 3.1. Os desafios

O principal desafio deste trabalho é permitir a extração de informação relevante para a prestação de serviços médicos num departamento de imagiologia. Para isso serão usadas 3 fontes de informação distintas, o repositório de imagens médicas, o repositório de relatórios de ecocardiogramas e o tráfego DICOM na rede informática. Para cada tipo de fonte, será utilizado um sensor específico.

Um dos primeiros desafios surge a partir da localização dessas mesmas fontes de informação. A informação que se pretende obter não está localizada num ponto específico mas está dispersa por várias máquinas da instituição.

A diversidade de formatos entre as fontes e a falta de normalização no caso dos relatórios de ecocardiogramas vêm acrescentar complexidade à integração da informação num único sistema. As imagens médicas digitais, seguindo a norma DICOM, têm os seus meta dados organizados em conjuntos de campos. Cada campo é identificado por uma *tag*. Deve ser possível relacionar a informação obtida nas várias fontes. Para isso, devem ser escolhidos modelos de informação para os outros sensores que deem aos utilizadores uma noção de uniformidade e conexão entre a informação proveniente das várias fontes. Os utilizadores não devem ter a percepção de que o sistema integra 3 fontes de informação e não que estão a utilizar 3 sistemas diferentes.

A quantidade de informação que se pode extrair a partir destes 3 tipos de fontes de informação médica é enorme. Uma parte já se pode obter a partir dos sistemas atuais usados nas instituições e, por isso, embora seja um desafio interessante ao nível técnico não é de muita relevância na prática pois não traz nada de novo face ao que já existe. Saber o que é “informação relevante”, para os profissionais de saúde que trabalham diariamente com os sistemas e têm a definir estratégias para otimizar o seu desempenho, exige contacto com a sua realidade e ouvir as suas necessidades.

Devido à multiplicidade de estações de trabalho existentes hoje em dia em qualquer instituição de saúde, é importante que a informação obtida pelo sistema desenvolvido possa ser visualizada em qualquer computador dentro da instituição. Por exemplo: na tentativa diária de optimização dos serviços ou numa reunião de planeamento do departamento, deve ser possível extrair informação do sistema com um portátil ligado à rede interna da instituição.

### 3.2. Requisitos funcionais

#### *Registo das mensagens DICOM na rede*

O sistema deve ser capaz de registar as mensagens DICOM que são transmitidas em determinados pontos da rede da instituição médica. O sensor de rede será o único elemento a

recolher e registar informação. Nos restantes dois casos, essa informação já existe e tem sido guardada durante os últimos anos.

A informação deve ser registada de forma estruturada para que seja possível analisá-la posteriormente.

#### ***Envio de relatórios de eventos anormais na rede***

Dada a quantidade de informação DICOM que circula na rede das instituições, o programa deve ser capaz de analisar essa informação, os comandos, as imagens médicas e enviar relatórios de eventos em que determinadas condições se verifiquem.

Os utilizadores deverão ter a possibilidade de definir e editar as condições de teste de eventos. Os relatórios/alertas deverão ser gerados pelo sistema em tempo real, isto é, o sistema deve processar os dados obtidos a partir da rede logo após o momento em que eles são recolhidos. O processamento não deve ser feito apenas periodicamente.

O mecanismo de transmissão predefinido dos relatórios/alertas será o email.

#### ***Pesquisa integrada entre todas as fontes***

Os utilizadores do sistema devem conseguir pesquisar sobre a informação das diversas fontes de forma integrada. Deve existir uma interface única que integre as pesquisas sobre as 3 fontes de informação. Além disto, nos casos em que for possível, o utilizador deve ter a possibilidade de relacionar os resultados obtidos a partir de uma fonte com os dados disponíveis em outras fontes.

Por exemplo: Ao obter uma imagem médica como resultado de uma pesquisa, o utilizador deve ser capaz de ver quando é que esta foi transferida através da rede e obter o relatório do exame, caso a imagem seja de um ecocardiograma.

#### ***Obtenção de indicadores de desempenho***

Através do contacto com profissionais de saúde, surgiram vários indicadores de desempenho que serão úteis para avaliar e decidir estratégias para melhorar a eficiência dos serviços.

O sistema deve ser capaz de calcular e apresentar esses indicadores em tempo útil aos seus utilizadores.

#### ***Exportação de dados***

O sistema desenvolvido deve ser capaz de exportar dados relativos às pesquisas e indicadores de desempenho em formatos compatíveis com o Excel ou programas semelhantes. Este requisito é importante pois permitirá aos utilizadores fazerem análises que não foram inicialmente planeadas.

### **3.3. Requisitos não funcionais**

De seguida serão definidos alguns requisitos fundamentais que um sistema desta natureza deve cumprir. No subcapítulo 3.4 será apresentada a arquitetura geral do sistema desenvolvido, que pretende cumprir os requisitos aqui apontados.

#### ***Arquitetura distribuída***

Para extrair e processar informação que se encontra dispersa pela instituição, a arquitetura do sistema desenvolvido deve ser distribuída. Além disso, pode existir múltipla

instanciação de cada um dos sensores. Os repositórios podem estar distribuídos por várias máquinas e pode existir mais que um ponto de captura de informação na rede.

#### ***Sistema multiplataforma***

Todos os componentes do sistema devem ser independentes do sistema operativo usado na máquina em que são instanciados. Devido à variedade de sistemas de informação existentes em ambientes médicos, é importante que a solução desenvolvida não restrinja a sua instanciação a uma plataforma ou sistema de ficheiros.

#### **3.3.1. Segurança**

A informação médica apresenta um carácter especialmente sensível. Ao usar este tipo de informação torna-se imperativo garantir a sua segurança nos seus variados aspetos.

#### ***Confidencialidade nas comunicações***

Num sistema distribuído há a necessidade de usar a rede informática para transmitir informação e executar operações remotamente. As comunicações devem ser confidenciais, impedindo assim que um atacante consiga ter acesso à informação que está a ser transmitida.

Este requisito pode parecer incoerente, uma vez que uma das fontes de informação do sistema apresentado é informação que circula na rede da instituição. No entanto, este requisito passa a fazer sentido se tivermos em conta que o sistema distribuído pode ser mais abrangente do que o sistema PACS. Isto é, as comunicações do PACS, não confidenciais, podem ser facilmente circunscritas a uma secção da rede da instituição. Já ao nível do sistema apresentado nesta dissertação, não deve ser imposta nenhuma restrição quando à sua abrangência ao nível da rede.

#### ***Autenticação dos componentes e utilizadores***

Dado que o sistema distribuído vai ter vários componentes (sensores e agregador), é necessário que cada um dos componentes e utilizadores do sistema sejam autenticados. Isto evitará que um atacante consiga usar o sistema para obter informações privilegiadas às quais não deveria ter acesso.

#### ***Anonimização dos identificadores***

Para analisar os fluxos de trabalho de uma instituição médica e extrair informação que ajude no planeamento e tomada de decisões não relacionadas com casos de pacientes específicos, não é necessário visualizar os identificadores dos pacientes e exames.

A anonimização de dados clínicos é uma área complexa, que tem sido tema de várias discussões no meio científico. O artigo [31] apresenta uma técnica que permitiu reconhecer um número considerável de pacientes (27,6%) usando uma técnica de reconstrução 3D de imagens a partir de exames CT (*Computed Tomography*), disponível em algumas aplicações médicas de visualização. Após as reconstruções 3D, usaram as imagens reconstruídas e fotos reais dos pacientes na aplicação Picasa da Google e conseguiram relacionar uma quantidade considerável de imagens reconstruídas com os pacientes reais. O Picasa é uma aplicação gratuita e permite fazer reconhecimento facial em fotos.

O exemplo anterior demonstra bem o cuidado que se deve ter ao lidar com informação médica. Contudo, este trabalho não procura apresentar soluções para problemas da complexidade daquele que foi apresentado. Para o caso específico apresentado, os utilizadores podem escolher não usar as imagens propriamente ditas e apenas usar a meta

informação das imagens, impedindo assim qualquer tipo de identificação do paciente através dos pixels das imagens.

Os principais campos de meta informação que podem identificar um paciente ou um exame são: o nome do paciente, o ID do paciente e o *accession number* que identifica os procedimentos médicos. O mecanismo de anonimização usado deve ser capaz de manter a relação entre dados do mesmo paciente sem permitir relacionar os dados com o paciente real. Alguns dos estudos que se pretendem realizar podem exigir que seja possível relacionar informações relativas a um mesmo paciente ou exame.

### 3.4. Arquitetura geral do sistema

A arquitetura do sistema concebido procura cumprir todos os requisitos apontados até aqui. A Figura 3.1 apresenta o diagrama de arquitetura de alto nível. O sistema desenvolvido tem uma arquitetura orientada a serviços (SOA). Os vários sensores disponibilizarão as suas funcionalidades através de serviços. Os serviços disponibilizados pelos sensores serão orquestrados por uma aplicação Web chamada “Medical Analysis Web”. Esta aplicação irá interagir simultaneamente com os vários serviços e com os utilizadores.

Usar uma arquitetura orientada a serviços torna possível a reutilização dos sensores noutros sistemas. Também permite que a instanciação do sistema seja flexível, isto é, podem não ser utilizados todos os tipos de sensores ou podem haver múltiplas instâncias de cada um dos sensores.

Os vários sensores irão ter a informação armazenada localmente ao invés de a enviar para um ponto centralizado. A principal desvantagem de utilizar esta abordagem prende-se com o facto de ser necessário aceder aos sensores pela rede, cada vez que se pretende obter informação sobre uma das fontes. O processo de apresentação da informação vai ser sempre mais lento e o sistema será sensível a possíveis falhas na rede da instituição.

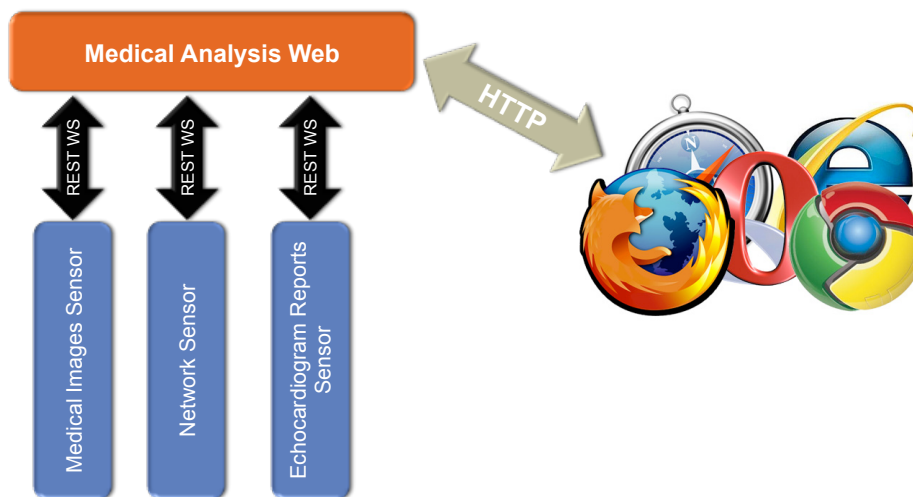


Figura 3.1: Arquitetura de alto nível do sistema.

Contudo, as vantagens de usar esta abordagem são mais significativas e importantes. Enviar os dados recolhidos para um ponto centralizado iria aumentar a complexidade da solução, uma vez que no caso dos repositórios de imagens e relatórios médicos isso iria criar redundância de informação. Os repositórios de imagens médicas originais continuariam a existir paralelamente. Além disso, cada atualização dos repositórios ou recolha de informação do protocolo de rede DICOM, iria necessitar de usar a rede da instituição para atualizar o repositório centralizado. Isto poderia provocar um congestionamento na rede da instituição e

iria contra a filosofia do SOA. Um serviço deve ser autónomo e deve fazer sentido existir independentemente do sistema onde se utiliza. Com a informação guardada localmente em cada um dos sensores, estes podem até ser utilizados por vários sistemas diferentes em simultâneo.

Uma vantagem clara de usar uma aplicação Web para orquestrar os serviços e agregar a informação é que os utilizadores não precisam de instalar uma aplicação específica nas várias estações de trabalho que usam para interagirem com o sistema. Qualquer *Web browser* recente será capaz de interagir com o sistema desenvolvido. Além disso, dependendo das restrições impostas pelo administrador de rede da instituição, será possível aceder à plataforma a partir de qualquer ponto da rede ou até mesmo fora da instituição.

Conforme referido anteriormente, o sensor de imagens médicas usado será o Dicoogle, cuja arquitetura de alto nível já foi apresentada. A arquitetura dos restantes componentes do sistema será apresentada com maior detalhe nos capítulos correspondentes.

#### 3.4.1. Segurança das comunicações

Uma das prioridades deste sistema distribuído é ter um nível de segurança adequado à importância dos dados que utiliza.

Para trazer confidencialidade às comunicações com os sensores, os *web services* disponibilizados usam HTTPS. O SSL pode garantir a autenticidade dos sensores através dos seus certificados. Para isso deve ser gerado um certificado diferente para cada um dos sensores. Num ambiente real, o servidor Web também deverá usar HTTPS.

Quanto à autenticação dos clientes dos *web services*, o método escolhido foi o *HTTP Basic Authentication* [32]. Uma vez que o canal de comunicação é seguro, não existe risco das credenciais de acesso serem interceptadas por atacantes [33].

Os utilizadores do Medical Analysis Web são autenticados por um nome de utilizador e uma password. Os utilizadores poderão pertencer a um de 2 grupos: os utilizadores normais e os administradores. As contas dos novos utilizadores terão que ser ativadas por um dos administradores. Só após a ativação é que os utilizadores podem fazer *login* no site.

#### 3.4.2. Anonimização de dados clínicos

Tal como foi apresentado no subcapítulo dos requisitos, é necessário garantir que os pacientes não podem ser identificados no processo de utilização deste sistema de análise dos fluxos de trabalho. Todos os sensores lidam com informação capaz de identificar os pacientes.

Os campos a anonimizar serão o nome do paciente, o ID do paciente e o *accession number* do procedimento clínico. Para manter a relação entre informações de um mesmo paciente e de um mesmo exame, foi adotada uma estratégia de pseudo-anonimização. Esta estratégia consiste na substituição dos nomes e IDs por outros nomes e IDs, que não correspondem aos reais. Durante o processo de recolha de informação, é mantida a relação entre o nome ou ID real e o nome ou ID fictício. Assim, o mesmo nome ou ID na realidade será sempre traduzido no mesmo nome ou ID fictício.

Após o processo de recolha de dados, pode-se descartar essa informação que permite relacionar com os dados reais e assim impossibilitar a relação direta com os pacientes reais. Este processo de pseudo-anonimização é feito partilhando uma biblioteca JAVA chamada "AnonymousPatientData" entre os vários sensores. Esta biblioteca contém uma tabela que faz a relação entre os campos identificadores os campos pseudo-anonimizados. Esta tabela é guardada e cifrada numa base de dados persistente.

No entanto, para que fosse possível relacionar informação proveniente dos vários sensores, foi necessário ir além de uma pseudo-anonimização local. A anonimização feita entre os vários sensores deve ser coerente de forma a que, por exemplo, as imagens médicas de um exame e o seu respetivo relatório possam ser relacionadas.

Dada esta necessidade, foi desenvolvido um método de anonimização que exige a escolha de um sensor *Master*, que irá anonimizar os dados localmente e os restantes sensores (*Slaves*) irão usar um serviço de anonimização prestado pelo *Master* (Figura 3.2).

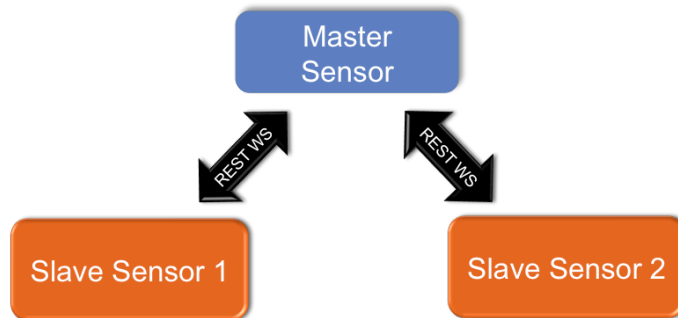


Figura 3.2: Arquitetura de anonimização.

O administrador que instanciar o sistema, deve decidir qual é o *Master* de anonimização e configurar todos os outros sensores para anonimizarem a informação remotamente. Dependendo da configuração definida, um sensor pode anonimizar a informação local ou remotamente.

A Figura 3.3 apresenta um exemplo de configuração de anonimização remota. No caso apresentado, o sensor de rede foi configurado para anonimizar a informação remotamente e além disso, usar autenticação para aceder aos *Web services* do *Master*. O *Master* está a correr na máquina com o IP 192.168.2.6, na porta 9999 e exige autenticação.

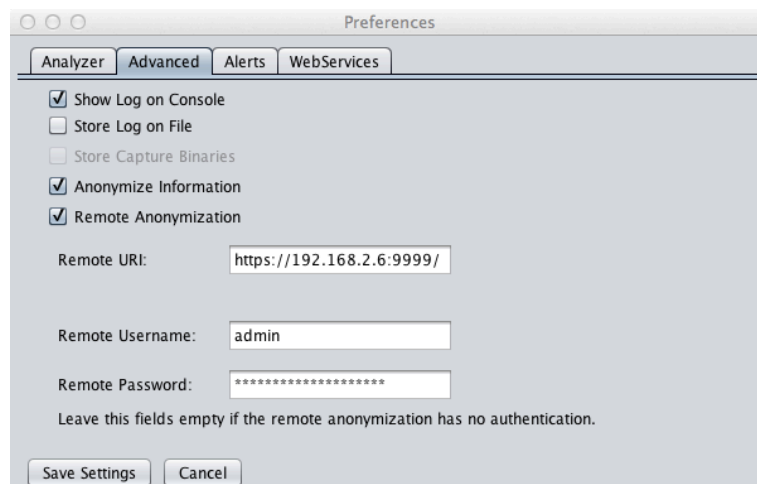


Figura 3.3: Preferências de Anonimização do sensor de rede.

### Serviço de anonimização

O serviço de anonimização é comum entre o sensor de rede e o sensor de relatórios de ecocardiogramas. A Tabela 3.1 apresenta os métodos disponibilizados pelo serviço de anonimização. Os dados de entrada e de saída dos métodos são texto simples (MIME "text/plain").



Tabela 3.1: Métodos do *web service* de anonimização.

<i>Método</i>	<i>Dados de entrada</i>	<i>Método HTTP</i>	<i>Descrição</i>
/Anonymize/save	-	GET	Guarda as correspondências usadas recentemente
/Anonymize/getName	Nome real do paciente	POST	Retorna o pseudônimo do nome do paciente
/Anonymize/getPatientID	ID real do paciente	POST	Retorna o ID do paciente anonimizado
/Anonymize/getAccessionNumber	<i>Accession Number</i> do exame	POST	Retorna o <i>Accession Number</i> anonimizado

### 3.5. Adaptações ao Dicoogle

O Dicoogle foi apresentado no subcapítulo 2.5 porque cumpre os requisitos necessários para ser utilizado como sensor de imagens médicas. Neste subcapítulo serão apresentados os *web services* que o Dicoogle disponibilizava e os novos serviços que tiveram que ser desenvolvidos (numa base *open-source*) para que este fosse integrado no sistema desenvolvido.

A Tabela 3.2 apresenta os *web services* que já existiam no Dicoogle. Todos os *web services* disponibilizados pelo Dicoogle devolvem informação em formato XML (MIME “application/xml”).

Tabela 3.2: *Web services* previamente existentes no Dicoogle.

<i>Método</i>	<i>Dados de entrada</i>	<i>Método HTTP</i>	<i>Descrição</i>
/dim?q={query}	<i>Query</i> de pesquisa	GET	Executa uma pesquisa sobre o repositório e obtém a lista de resultados.
/file?uid={uid}	UID da imagem médica	GET	Permite fazer o download de uma imagem médica
/dump?uid={uid}	UID da imagem médica	GET	Obtém todas as <i>tags</i> indexadas para uma imagem médica.

Todos os serviços da tabela anterior são utilizados pelo sistema desenvolvido. Contudo, estes serviços não eram suficientes para cumprir todos os requisitos do sistema, relativamente ao sensor de imagens médicas.

A Figura 3.4 apresenta a estrutura XML da lista de resultados retornados pelo serviço de pesquisa do Dicoogle. No exemplo da figura são retornadas duas imagens que satisfazem os critérios de pesquisa.

```

<DIM>
<Patient name="patient 1" id="1">
  <Study date="20060727" id="1.2.276.0.200607270934002006276091">
    <Serie modality="CT" id="1.3.12.2.1107.5.1.4.54472.30000006072707295196800007542">
      <Image path="DICOM/file1.dcm" uid="1.3.12.2.1107.5.1.4.54472.30000006072707295196800007545" />
    </Serie>
    <Serie modality="CT" id="1.3.12.2.1107.5.1.4.54472.30000006072707295196800006906">
      <Image path="DICOM/file2.dcm" uid="1.3.12.2.1107.5.1.4.54472.30000006072707295196800007459" />
    </Serie>
  </Study>
</Patient>
</DIM>

```

Figura 3.4: Exemplo de uma lista com dois resultados em resposta a uma pesquisa no Dicoogle.

A Tabela 3.3 descreve os novos serviços do Dicoogle que foram desenvolvidos no âmbito deste trabalho. Os novos serviços desenvolvidos para o Dicoogle foram enviados como contributo e estão acessíveis a todos os programadores que trabalham no desenvolvimento da aplicação. Assim, poderão ser incluídos nas próximas versões oficiais.

Tabela 3.3: Serviços desenvolvidos no Dicoogle para o sistema desenvolvido.

<i>Método</i>	<i>Parâmetros de entrada</i>	<i>Método HTTP</i>	<i>Descrição</i>
/tags	-	GET	Obtém a lista de todas as <i>tags</i> DICOM
/image?uid={uid}&height={height}	UID da imagem e altura pretendida	GET	Gera uma imagem JPEG a partir de uma imagem médica
/enumField?field={field}&type={type}	Nome do campo e tipo do campo ( <i>string</i> ou <i>float</i> )	GET	Enumera todos os valores que um determinado campo tem em todo o repositório
/countResults?q={query}	<i>Query</i> de pesquisa	GET	Conta o número de resultados obtidos como resposta a uma determinada <i>query</i>
/examTime?action={action}	Ação de pré-processamento	GET	Usado para fazer pré-processamento relativo à duração e número de imagens dos exames

### 3.5.1. Serviço de pré-processamento

Alguns dos indicadores e métricas que se pretendem obter com este trabalho exigem um grande tempo de processamento (cerca de 42 minutos com a máquina descrita no subcapítulo 7.2) pois envolvem todos os dados indexados neste sensor de imagens médicas.

O serviço de pré-processamento, referido na última linha da Tabela 3.3, serve para analisar todo o repositório indexado de forma a extrair para cada exame indexado, o seu tempo total de duração e o número de imagens médicas digitais extraídas. Os resultados são armazenados num ficheiro CSV que será utilizado mais tarde para extrair as métricas pretendidas. A razão de criar este serviço de pré-processamento prende-se com o facto de não ser possível fazer este processamento em relativamente pouco tempo.

No capítulo de apresentação dos resultados iremos ver alguns dos resultados obtidos, dependentes da utilização deste serviço. Usando este recurso, é possível executar algumas ações sobre o sensor:

- **getState:** retorna o estado atual do serviço. O estado pode ser: EMPTY, RUNNING ou READY.
- **startCalc:** esta ação cria uma *thread* de processamento que vai produzir o ficheiro CSV com os resultados.
- **stopCalc:** interrompe a *thread* de processamento.
- **percentage:** retorna a percentagem de conclusão do processamento.
- **download:** permite fazer o *download* do ficheiro CSV, caso o processamento já tenha sido concluído.



## 4 Sensor de rede

Conforme foi apresentado no capítulo 2, a norma DICOM é a norma mais importante na área da imagiologia médica. O seu protocolo de rede possibilita que várias aplicações e dispositivos do PACS comuniquem entre si, mesmo sendo desenvolvidos por empresas diferentes. Embora existam componentes do sistema PACS que possam usar protocolos proprietários para comunicar entre si, a grande maioria usa a norma DICOM.

Ter um sensor de rede, que analise e registre os fluxos de informação na rede da instituição, permite que os vários eventos sejam registados e seja extraída informação relevante, contida nos pacotes transferidos. O facto de ter um analisador de rede permite que o sistema tenha acesso de forma transparente e interoperável, aos dados que são guardados, em sistemas de informação proprietários. Para além disso, alguma da informação transferida não é registada, nem mesmo nesses sistemas proprietários e pode ser registada pelo sensor de rede. A informação extraída, quando combinada com outras fontes de informação, pode tornar-se muito útil para analisar os fluxos de trabalho de um departamento de imagiologia e ajudar a promover uma melhoria contínua dos serviços prestados e uma otimização dos recursos.

### 4.1. Componentes usados

O sensor de rede foi desenvolvido de raiz usando a linguagem de programação Java. A Figura 4.1 apresenta os principais componentes de *software* usados pelo sensor de rede.

Para capturar os pacotes que circulam na rede foi utilizada a biblioteca Jpcap [34]. A captura é feita em modo promíscuo, isto é, todos os pacotes que chegam à interface de rede são enviados para o sensor, mesmo que a origem e o destino não sejam a máquina em que o sensor está a ser executado. Todos os pacotes TCP capturados são testados de forma a verificar se são pacotes DICOM ou não. Sempre que um pacote DICOM é detetado, os seus conteúdos são extraídos, segundo as estruturas definidas no *DICOM Upper Layer Protocol* (PDUs), usando classes desenvolvidas para o efeito (Figura 4.3). Depois de descodificar os PDUs, a aplicação passa a analisar a informação do protocolo ao nível dos serviços (*DICOM Message Service Elements*), usando para o efeito algumas funcionalidades da biblioteca DCM4CHE2 [35].

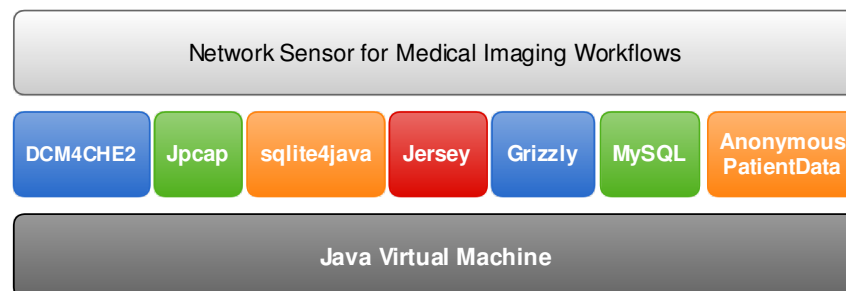


Figura 4.1: Principais componentes do sensor de rede.

Para guardar a informação extraída pelo sensor, pode ser utilizada uma base de dados SQLite através da biblioteca *sqlite4java* [36] ou uma base de dados MySQL [37]. O objetivo ao dar a possibilidade de serem utilizados dois motores de bases de dados diferentes (não em simultâneo) é dar ao maior flexibilidade ao utilizador na instanciação do sensor de rede. Isto é, usando o motor de SQLite facilita-se a instanciação do sensor pois não é preciso ter uma

instalação de um motor de base de dados MySQL no sistema, não tendo assim qualquer dependência externa. Por outro lado, a solução usando MySQL o sensor terá notoriamente maior desempenho do que usando SQLite, para ser usado em ambientes de produção com elevados níveis de utilização [38,39]. O anexo 10.1 contém o modelo físico da base de dados do sensor de rede.

Dada a arquitetura do sensor, substituir o motor de base de dados não é uma tarefa árdua pois todas as *queries* à base de dados estão centralizadas numa única classe. Esta classe implementa todos os métodos de acesso à base de dados por parte de todas as restantes classes da aplicação. Assim, para utilizar outro motor de base de dados, é apenas necessário desenvolver uma nova implementação da mesma interface.

O Jersey é uma implementação da API JAX-RS para usar e implementar *web services* RESTful [40]. O Grizzly [41] é um pequeno servidor Web que executa os *web services* criados pelo Jersey. Os *web services* fornecidos por estes dois componentes são a principal interface de comunicação com o exterior, isto é, através deles as aplicações podem aceder à informação obtida através da rede.

A biblioteca AnonymousPatientData, que foi falada no subcapítulo 3.4.2, é uma biblioteca que permite substituir IDs e nomes por pseudónimos a partir de uma lista interna nomes. Caso o nome esteja a ser traduzido pela primeira vez, a seleção do novo nome a utilizar na substituição é aleatória, caso um nome já tenha sido traduzido anteriormente, a tradução é repetida para preservar a relação entre os nomes originais.

## 4.2. Reconstrução dos fluxos TCP

Conforme foi referido anteriormente, o protocolo de rede DICOM funciona sobre TCP. O protocolo TCP permite que as aplicações se abstraíam da forma como os dados são divididos em sucessivos pacotes e apenas se limitem a ler e a escrever bytes, de e para o canal de comunicação.

Como foi dito em 2.2.4, o TCP garante a ordenação dos pacotes na recepção e a retransmissão no caso de haverem pacotes perdidos. As aplicações que usam TCP não têm portanto que implementar mecanismos de ordenação dos dados recebidos nem de espera por pacotes perdidos que irão entretanto ser retransmitidos. Este mecanismo de gestão de fluxos implementado pelo TCP permite que os programadores se abstraíam de tudo isso durante o desenvolvimento das suas aplicações.

Ao nível da camada de rede do modelo TCP/IP, o IP é o protocolo responsável pelo endereçamento e encaminhamento dos pacotes. Cada tipo de *link layer* (segunda camada do modelo TCP/IP) tem normalmente um MTU (*Maximum Transmission Unit*) associado. Caso um pacote IP seja maior que o MTU da rede, esse pacote irá ser fragmentado, ou seja, dividido em vários pacotes de forma a poder ser transmitido naquela rede. Nas redes Ethernet habituais, o MTU é de 1500 bytes. No entanto, noutros tipos de rede, ou em redes Ethernet com outros protocolos de encapsulamento antes do IP (exemplo: PPPoE), o MTU tende a ter um valor mais baixo. Por isso, um pacote grande que seja transmitido através de vários tipos de redes pode vir a ser fragmentado.

Ao capturar pacotes diretamente nas interfaces de rede, não há intermediação da *stack* TCP/IP, o que se obtém são pacotes com todos os bytes, sem qualquer pré-processamento. Os pacotes capturados podem estar fragmentados e, em grande parte dos casos, desordenados. O RFC 815 descreve um algoritmo de reagrupamento dos pacotes IP fragmentados [42].

Mesmo em redes bem dimensionadas é muito provável ocorrerem perdas e retransmissões de pacotes TCP. Isto acontece devido aos mecanismos de controlo de fluxo do

TCP, *Slow Start* e *Congestion Avoidance* [43], que tentam manter a transmissão de dados próxima da largura de banda total disponível, sem provocar congestionamento na rede.

Dado que o protocolo de rede DICOM funciona sobre o TCP, foi necessário desenvolver uma solução para reconstruir os fluxos TCP, isto é, desfragmentar/reagrupar os pacotes, reordená-los e esperar por retransmissões, caso existam falhas.

Na Figura 4.2 podemos ver as duas classes responsáveis por reconstruir os fluxos da sessão TCP. Existe uma instância da classe *BuildTCPSession* por cada fluxo de dados TCP, isto é, duas instâncias por cada sessão TCP ativa, uma para cada um dos sentidos. Sempre que um pacote de um fluxo é recebido, é enviado para esta classe (método *addPacket*) que coloca o pacote numa fila ordenada segundo o *Sequence Number* do pacote (usando a classe *TCPPacketComparator*). Por fim, através do método *getNextPacket*, é possível obter os pacotes reconstruídos e ordenados.

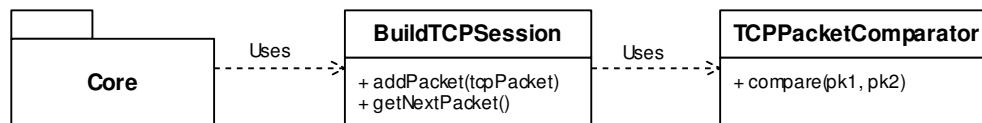


Figura 4.2: Classes do *package* TCP.

### 4.3. Arquitetura

A arquitetura do sensor de rede está dividida em 7 *packages*:

- **Core:** Contém a classe principal, as classes que guardam e gerem as definições da aplicação e outras classes que são responsáveis por usar uma interface de rede para capturar pacotes TCP. Esta captura é feita através de *threads*, permitindo assim que a aplicação possa capturar em múltiplas interfaces de rede em simultâneo.
- **DICOM:** Contém as classes responsáveis por fazer descodificar os pacotes TCP e verificar se são ou não pacotes DICOM. Também contém classes para extrair toda a informação relevante dos comandos DIMSE e dos objetos DICOM dentro desses comandos. Este *package* divide-se em vários *sub-packages*: **UL**, **UL.Items**, **DIMSE** e **ProcessObjects**.
- **Database:** Contém uma classe abstrata (DBManager) que contém a interface das classes de acesso à base de dados e implementa os métodos comuns a todas essas classes que não dependem diretamente do motor de base de dados. Também contém as implementações para SQLite e MySQL.
- **GUI (Graphical User Interface):** O sensor tem uma interface gráfica simples que permite aos utilizadores gerir as configurações, iniciar e parar as capturas. Enquanto o sensor está a ser executado não é necessário ter nenhuma janela visível na área de trabalho pois existe uma classe que cria um ícone na bandeja do sistema (*system tray*), permitindo esconder todas as janelas e ainda assim recuperar o controlo da aplicação através da interface gráfica do sistema.
- **Security:** Este *package* inclui os componentes que vão implementar os mecanismos de segurança da aplicação. Neste *package* está uma classe que verifica as credenciais de autenticação *HTTP Basic Authentication* dos *web services*, outra classe para anonimização da informação (local ou remota) e os

ficheiros KeyStore e TrustStore, criados pelo keytool [44], que gerem os certificados usados e verificados pelo SSL.

- **TCP:** Contém as classes usadas para reconstruir as sessões TCP e transformar os bytes recebidos num fluxo de bytes contínuo, conforme foi referido no subcapítulo 4.2.
- **WebServices:** Neste *package* estão as classes que implementam os métodos dos *web services* REST que o sensor fornece. Os métodos implementados serão detalhados no subcapítulo 4.6.

Para fazer *parse* dos pacotes a baixo nível, segundo o protocolo DICOM UL, foram implementadas várias classes, uma para cada tipo diferente de PDU, além de classes abstratas que lidam com as partes que são semelhantes entre os vários PDUs. A Figura 4.3 apresenta um diagrama com as classes envolvidas na descodificação de PDUs.

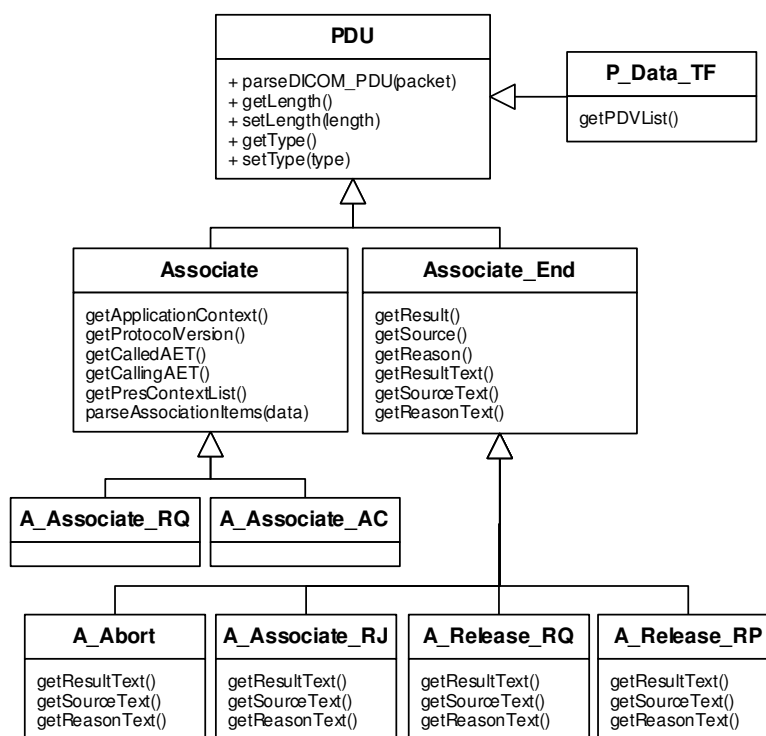


Figura 4.3: Diagrama de Classes (*package* DICOM.UL).

Conforme foi dito no capítulo de apresentação da norma DICOM (subcapítulo 2.2), o protocolo de rede DICOM não obriga a que seja usada sempre a porta 104. Logo, não é possível usar um filtro de pacotes por porta e têm que ser analisados todos os pacotes TCP capturados. Para detectar pacotes DICOM é necessário começar por verificar o valor do primeiro byte dos dados contidos no pacote TCP. Se este byte tiver um valor entre 0x01 e 0x07 pode ser um PDU DICOM, então os dados do pacote são enviados para o construtor da classe correspondente (uma das 7 classes da Figura 4.3 que não são estendidas). Essas classes começam a fazer *parse* dos pacotes byte a byte e verificam se eles têm a estrutura definida na norma, caso não tenham, as classes lançam uma exceção descartando a possibilidade de se tratarem de pacotes DICOM.

Quando o processo de análise do protocolo DICOM UL termina, a aplicação já sabe de que tipo são os PDUs capturados na rede. Sempre que o PDU é um P-Data-TF, a aplicação precisa de analisar também o conteúdo do objeto DICOM que este contém. Este objeto pode ser um comando (DIMSE) ou pode conter a informação que se pretende transferir entre AEs.



Existem várias classes que cooperam para realizar esta tarefa de análise do conteúdo dos objetos DICOM (Figura 4.4). O sensor desenvolvido consegue lidar com múltiplas associações em simultâneo.

A classe *ParseDICOMObject* recebe os PDUs e tem como principal função gerir a lista de associações ativas (Figura 4.4). Além disso, quando se pretende pseudo-anonimizar a informação, esta classe usa métodos da classe de pseudo-anonimização do *package* "Security" para substituir os campos de identificação do paciente e do exame. Se o PDU é um P-Data-TF, esta classe usa a classe *DicomInputStream* da biblioteca DCM4CHE2 para ler o objeto DICOM encapsulado no PDU e enviá-lo para a instância da classe *Association* correspondente. Uma associação DICOM é identificada da mesma forma que uma sessão TCP, ou seja, através dos seguintes valores: IP de origem, porta de origem, IP de destino e porta de destino. Para identificar facilmente a associação DICOM é utilizada a classe *AssociationKey*.

A anexo 10.2 apresenta mais alguns detalhes do processo de captura e descodificação de pacotes. A implementação do sensor de rede desenvolvida processa os comandos C-ECHO, C-FIND, C-STORE, C-MOVE e C-GET. Estes comandos permitem extrair as métricas identificadas como requisitos do sistema. Todos os outros comandos não são, por enquanto, processados e por isso são ignorados quando capturados a partir da rede.

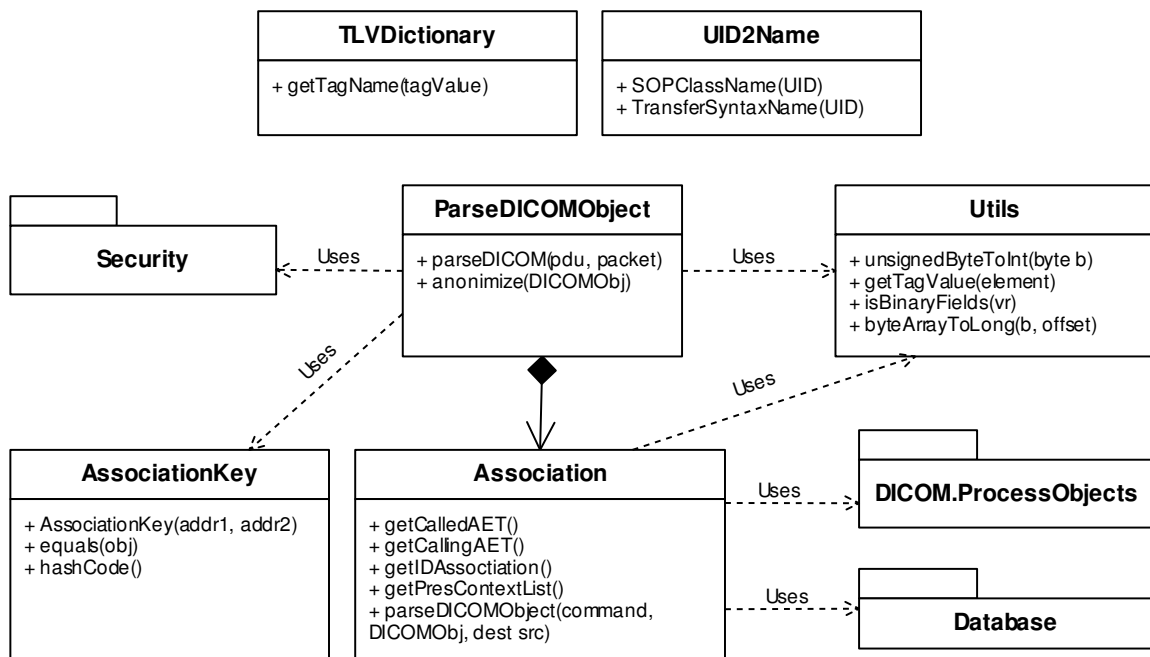


Figura 4.4: Diagrama de Classes (*package* DICOM).

A interface gráfica do sensor (Figura 4.5) permite-nos editar configurações, iniciar ou parar capturas em interfaces de rede, iniciar e parar o servidor Web e descodificar ficheiros binários com o conteúdo de pacotes DICOM. Através da Figura 4.5 podemos ainda ver que existiam 7 associações DICOM ativas no momento e era feita apenas uma captura (apenas uma interface de rede usada).

A interface gráfica não permite fazer pesquisas ou extrair indicadores estatísticos, uma vez que isso é feito a partir dos *web services*, que serão detalhados no subcapítulo 4.6.

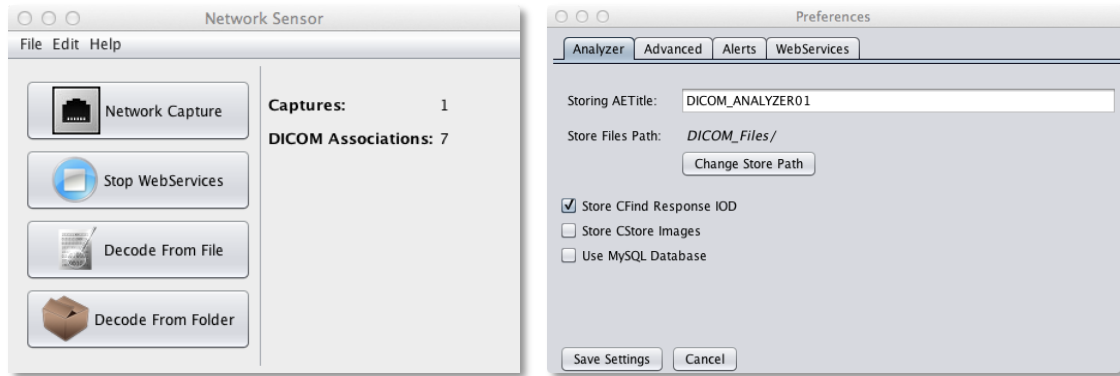


Figura 4.5: Janelas principal (à esquerda) e de preferências (à direita) do sensor de rede.

#### 4.4. Redes de switches

Hoje em dia, as redes de computadores utilizam essencialmente *switches* em vez de *hubs*, para fazer a interligação entre computadores e outros dispositivos na rede. Os *switches* aumentam o desempenho da rede, eliminam a colisão de pacotes e trazem alguma segurança às comunicações, uma vez que os pacotes *unicast* não são enviados para todos os nós da rede mas apenas para a porta que leva ao nó de destino. Apesar disto, a segurança de um canal de comunicação nunca deve depender do uso de *switches* na rede, dado que existem algumas formas de contornar esta barreira.

Este encaminhamento ao nível da camada de rede é feito usando uma tabela com todos os endereços MAC (*Media Access Control*) conhecidos e a respetiva porta de encaminhamento. Quando um pacote chega ao *switch*, este lê o endereço MAC de destino (que identifica o nó de destino) e, se estiver na tabela, envia apenas pela porta correspondente. Se não conhecer o endereço, então envia o pacote por todas as suas portas. É preciso ter em conta que o que pretendemos, com o sensor de rede, capturar pacotes da rede e este mecanismo dos *switches* traz algumas dificuldades, apesar de ter um impacto muito positivo no desempenho das redes de comunicações.

Existem essencialmente duas formas de contornar esta situação. A primeira consiste em colocar o sensor a executar na máquina que executa o servidor de armazenamento DICOM, isto é, Sistema de Arquivo PACS. Esta máquina irá intermediar quase todas as comunicações DICOM. A segunda alternativa é colocar o sensor a executar em outra máquina ligada ao mesmo *switch* e usar uma técnica chamada “*port mirroring*” [45] para enviar uma cópia de todo o tráfego da porta do servidor de armazenamento para a porta do sensor de rede. Alguns fabricantes têm nomes próprios para esta técnica: a Cisco chama-a “*Switched Port Analyzer*” (SPAN) e a 3Com “*Roving Analysis Port*” (RAP). Obviamente, esta segunda alternativa exige que se tenha acesso à configuração do *switch*. Devemos ainda ter em conta que o uso desta técnica pode ter um impacto negativo no desempenho do *switch*. Atualmente, dependendo da arquitetura e configuração usada pelo *switch*, esse impacto pode ser muito reduzido [45].

#### 4.5. Alertas e extração de imagens transmitidas

Para fornecer alertas e permitir efetuar cópias de segurança das imagens transmitidas, foi criada uma interface chamada *IProcessDICOMObject* que tem diversos métodos usados para processar cada um dos objetos DICOM capturados (Figura 4.6). Foram desenvolvidas duas implementações desta interface, a classe *DICOMObject2File* e *DICOMObjectAlert*. A

primeira permite guardar objetos DICOM, normalmente imagens, em ficheiros **.dcm** com a estrutura definida na norma. Esta funcionalidade pode ser útil para fazer backups de imagens médicas usadas recentemente, sem alterar nada no sistema PACS.

A segunda implementação da interface *IProcessDICOMObject*, usa a classe *CoreConditions* para obter uma lista de condições definidas através dos *web services* e, caso alguma das condições se verifique, é despoletado um alerta para um email definido nas configurações da aplicação.

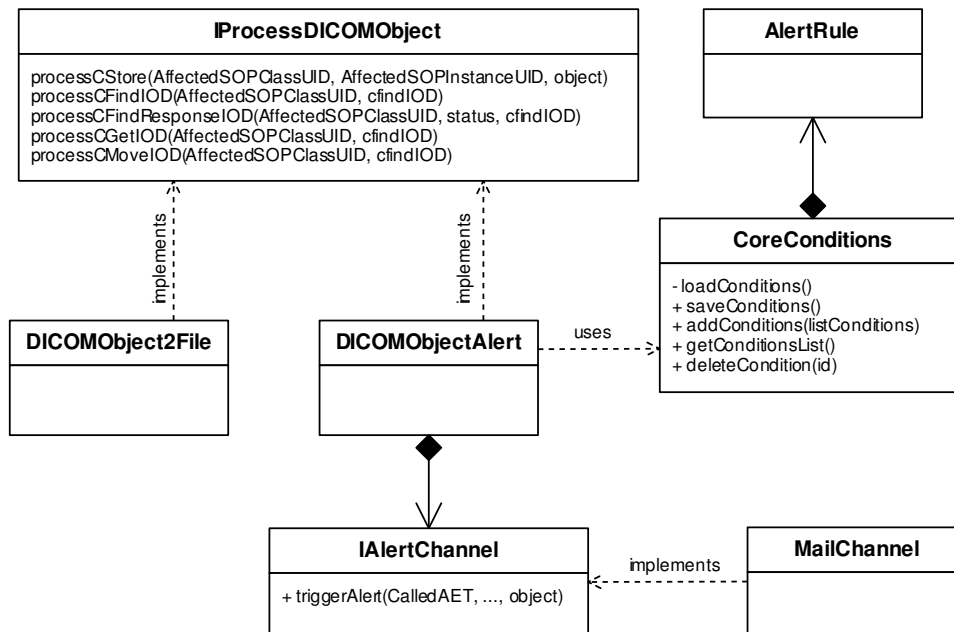


Figura 4.6: Diagrama de classes do *package* DICOM.ProcessObjects.

As regras para despoletar alertas podem ser visualizadas, adicionadas e removidas através dos *web services*, usando os métodos que serão apresentados no subcapítulo seguinte (4.6.1). A Figura 4.7 apresenta um exemplo de uma regra de alerta que pode ser adicionada à lista do serviço. Uma regra está obrigatoriamente associada com um único comando DICOM: C-STORE, C-FIND, C-GET ou C-MOVE e pode ter uma lista de condições. Cada condição da lista está associada com uma *tag* DICOM e deve ter uma das seguintes funções: LESS, LESS\_OR\_EQUAL, GREATER, GREATER\_OR\_EQUAL, EQUAL, NOT\_EQUAL, LIKE, NOT\_LIKE, BETWEEN, NOT\_BETWEEN, EXISTS ou NOT\_EXISTS. É importante referir que as condições de uma regra são agrupadas usando uma conjunção (operação lógica AND). Caso seja desejável usar uma disjunção lógica (operação OR), devem ser feitas duas regras diferentes para o mesmo comando.

Existem funções que só podem ser utilizadas se a *tag* tiver um valor numérico (ex. função BETWEEN) e outras funções adequadas para processar *strings* (ex. LIKE). Dependendo da função escolhida, a condição pode usar 1 ou 2 valores. Ao inserir novas regras pelos *web services*, elas são validadas para garantir a que têm uma estrutura correta.

Usando o exemplo da Figura 4.7, se num comando C-STORE for transferida uma imagem na qual se verificam as duas condições, o nome do paciente começa com as letras "Pa" e o seu ID está entre 10000 e 20000, é enviado um email para a conta definida nas configurações da aplicação. Este trata-se de um exemplo demonstrativo. No entanto, muitos outros cenários poderiam ser explorados com esta funcionalidade. Por exemplo, monitorizar a exposição ao raio X em função do peso do paciente e do protocolo clínico.

```

<alertRules>
  <AlertRule>
    <id>1</id>
    <command>CSTORE</command>
    <conditions>
      <tag>PatientName</tag>
      <value>Pa*</value>
      <function>LIKE</function>
    </conditions>
    <conditions>
      <tag>PatientID</tag>
      <value>10000</value>
      <value2>20000</value2>
      <function>BETWEEN</function>
    </conditions>
  </AlertRule>
</alertRules>

```

Figura 4.7: Exemplo de uma regra de alerta com duas condições, em formato XML.

## 4.6. Web services

O sensor de rede disponibiliza uma série de serviços REST para que seja possível interagir com este elemento a partir do exterior. Existem 4 grupos de serviços disponibilizados pelo sensor de rede: anonimização, gestão de alertas, gestão da aplicação e pesquisa de informação.

No subcapítulo 3.4.2 já foi apresentado o serviço de anonimização, que é comum ao sensor de rede e de relatórios de ecocardiogramas. Assim, nesta secção serão apenas apresentados os restantes serviços.

Os métodos dos *web services* desenvolvidos no sensor de rede podem consumir e produzir dados em formato XML ou JSON. Para evitar o uso de *parsers*, foi adoptada uma tecnologia do Java suportada pelo Jersey, o JAXB [46]. Através do JAXB é possível mapear XML ou JSON em classes de objetos Java, devidamente anotadas. Assim, os clientes dos *web services*, apenas têm que definir o tipo MIME do campo “Content-type” do pedido HTTP como “application/xml” ou “application/json” para poderem enviar os dados nesse formato. No caso da resposta, por defeito os dados são devolvidos em formato XML. Se o cliente definir o campo “Accept” do pedido HTTP com o tipo MIME relativo a XML ou JSON, os dados serão devolvidos no formato pretendido. Do ponto de vista da implementação, o que é necessário fazer para suportar estas alternativas é criar a estrutura de objetos pretendida, tendo as classes desses objetos as anotações necessárias.

### 4.6.1. Gestão de alertas

O serviço de gestão de alertas é composto por 3 métodos (Tabela 4.1). Estes métodos permitem ver, adicionar e remover regras de alerta do sensor. Ao adicionar ou remover regras da lista, as alterações têm efeito imediato sobre o sensor.

Tabela 4.1: Métodos do serviço de gestão de alertas.

<b>Método</b>	<b>Parâmetros de entrada</b>	<b>Método HTTP</b>	<b>Descrição</b>
/ManageAlerts/rules	-	GET	Obtém a lista de regras que estão a ser utilizadas atualmente

/ManageAlerts/rules	Lista de regras em XML ou JSON	POST	Adiciona regras à lista atual
/ManageAlerts/rules/{id}	ID da regra	DELETE	Apaga a regra com o ID enviado da lista atual

A Figura 4.7, apresentada anteriormente, apresenta o resultado de uma invocação do método “/ManageAlerts/rules” usando um HTTP GET. No caso apresentado, o sensor tinha apenas uma regra, com duas condições, na sua lista.

#### 4.6.2. Gestão da aplicação

O serviço de gestão da aplicação tem 4 métodos que permitem ver e editar as definições da aplicação, assim como ver e definir as interfaces de rede disponíveis na máquina onde o sensor está a ser executado (Tabela 4.2). Existem apenas dois estados possíveis para as interfaces de rede: a capturar ou em espera.

**Tabela 4.2: Métodos do serviço de gestão da aplicação.**

<i>Método</i>	<i>Parâmetros de entrada</i>	<i>Método HTTP</i>	<i>Descrição</i>
/ManageApp/settings	-	GET	Devolve uma estrutura com as definições do sensor
/ManageApp/settings	Estrutura com as definições do sensor	POST	Guarda as novas definições do sensor
/ManageApp/interfaces	-	GET	Retorna uma lista de todas as interfaces de rede e o seu estado atual
/ManageApp/interfaces	Lista das interfaces com o novo estado pretendido	POST	Atualiza o estado das interfaces de rede

A Figura 4.8 apresenta um exemplo de uma estrutura em formato JSON enviada para o sensor de rede para atualizar as suas definições, usando o segundo método da Tabela 4.2. O primeiro método da tabela retorna uma estrutura semelhante com as definições que estão atualmente a ser utilizadas pelo sensor de rede.

O penúltimo método da Tabela 4.2 permite ver o estado das interfaces de rede. A Figura 4.9 apresenta um possível resultado da invocação deste método. Segundo o exemplo da Figura 4.9, a máquina onde o sensor está a correr tem 4 interfaces de rede e o sensor está a capturar pacotes em duas delas em simultâneo: “en0 - Ethernet” e “en1 - Ethernet”.

```

{
  "saveCFindResponseIOD": "true",
  "logOnConsole": "true",
  "logOnFile": "false",
  "webServicesOnStartup": "true",
  "WSport": "9999",
  "storingAETitle": "DICOM_ANALYZER01",
  "PathDICOMFiles": "DICOM_Files/",
  "WSUsername": "admin",
  "WSPasswordHash": "",
  "WSAuthentication": "false",
  "AnonymizeInfo": "false",
  "RemoteAnonymization": "false",
  "RemoteUsername": "admin",
  "RemotePasswordHash": "",
  "RemoteAnonBaseURI": "https://localhost:9999/",
  "storeCaptureBinaries": "false",
  "storeCStoreImages": "false",
  "mailSettings": {
    "useSSL": "true",
    "smtpHost": "subdomain.SMTPHOST.com",
    "smtpPort": "465",
    "requireAuthentication": "true",
    "username": "Username",
    "password": "Password",
    "from": "from@SMTPHOST.com",
    "to": "to@SMTPHOST.com",
    "alertByMail": "false"
  },
  "useMySQLDatabase": "false",
  "MySQLDatabaseUser": "root",
  "MySQLDatabasePassword": ""
}

```

Figura 4.8: Mensagem HTTP POST com as definições do sensor de rede em formato JSON.

```

{
  "interfaceState": [{
    "id": "0",
    "interfaceName": "en0 - Ethernet",
    "capturing": "true"
  }, {
    "id": "1",
    "interfaceName": "fw0 - Apple IP-over-IEEE 1394",
    "capturing": "false"
  }, {
    "id": "2",
    "interfaceName": "en1 - Ethernet",
    "capturing": "true"
  }, {
    "id": "3",
    "interfaceName": "lo0 - BSD loopback",
    "capturing": "false"
  }
  ]
}

```

Figura 4.9: Estado atual das interfaces de rede em formato JSON.

Por sua vez, o último método permite definir o estado das interfaces, isto é, definir se o sensor deve capturar ou não, através das respectivas interfaces. Este método da tabela recebe uma lista de interfaces com o novo estado, conforme a Figura 4.10 apresenta. Ao invocar o método com o conteúdo da figura, a interface de rede com ID=2 vai parar a captura e a interface com ID=3 vai iniciar a captura, caso não estivesse a capturar previamente.

```

{
  "interfaceState": [{
    "id": "2",
    "capturing": "false"
  }, {
    "id": "3",
    "capturing": "true"
  }
  ]
}

```

Figura 4.10: Mensagem HTTP POST usada para definir o estado das interfaces de rede em formato JSON.

### 4.6.3. Pesquisa de informação

O serviço de pesquisa de informação é o mais importante do sensor de rede pois disponibiliza uma forma de pesquisar sobre a informação recolhida. A Tabela 4.3 contém a listagem de todos os métodos disponibilizados pelo serviço de pesquisa de informação.

Tabela 4.3: Métodos do serviço de pesquisa de informação.

<i>Método</i>	<i>Método HTTP</i>	<i>Descrição</i>
/Search/getAllTags	GET	Retorna a lista todas os campos indexados pelo sensor
/Search/AE	POST	Pesquisa de <i>Application Entities</i>
/Search/Assoc	POST	Pesquisa de associações
/Search/CStore	POST	Pesquisa sobre as mensagens C-STORE registadas
/Search/CFIND/Req	POST	Pesquisa mensagens C-FIND Request
/Search/CFIND/Resp	POST	Pesquisa sobre as respostas aos comandos C-FIND
/Search/CFIND/ReqIOD	POST	Pesquisa <i>tags</i> DICOM contidas nas mensagens C-FIND Request
/Search/CFIND/RespIOD	POST	Pesquisa <i>tags</i> DICOM nas respostas aos comandos C-FIND
/Search/CMove	POST	Pesquisa sobre os comandos C-MOVE
/Search/CMove/IOD	POST	Pesquisa <i>tags</i> DICOM contidas nos comandos C-MOVE
/Search/CGet	POST	Pesquisa sobre os comandos C-GET
/Search/CGet/IOD	POST	Pesquisa <i>tags</i> DICOM contidas nos comandos C-GET

Todos estes métodos recebem como parâmetro de entrada uma estrutura que contém uma lista de elementos, cada um tem obrigatoriamente um valor e, opcionalmente, uma condição associada (Figura 4.11). A condição pode ser uma das seguintes "=", ">", "<", ">=", "<=", "<>" ou "LIKE". Caso não seja indicada nenhuma condição para um elemento, a condição por defeito é "LIKE".

```

<rTags>
  <rTag>
    <tag>Time</tag>
    <value>1338983*</value>
  </rTag>
  <rTag>
    <tag>MoveOriginatorApplicationEntityTitle</tag>
    <value>Dicoogle</value>
  </rTag>
  <rTag>
    <tag>MessageID</tag>
    <value>10</value>
    <cond>&gt;</cond>
  </rTag>
</rTags>

```

Figura 4.11: Exemplo de estrutura enviada para os métodos de pesquisa, em formato XML.

Na Figura 4.11 podemos ver um exemplo do conteúdo enviado para o método “/Search/CStore”. O cliente está a procurar todos os registos de comandos C-STORE que tenham ocorrido num instante com *timestamp* começado por “1338993”, cuja origem do comando tenha sido a AE chamada “Dicoogle” e que o ID da mensagem seja superior a 10 (“&gt;” é o caractere ‘>’ codificado).

A Figura 4.12 apresenta um possível resultado da invocação do método usando o conteúdo do exemplo anterior. Através deste resultado conseguimos, por exemplo, obter o UID que identifica a imagem transferida. Com o UID da imagem médica, podemos usar os *web services* do Dicoogle para ver a imagem ou os seus meta-dados.

```
<cStores>
  <CStore>
    <IDCStore>2</IDCStore>
    <RefIDAssociation>8</RefIDAssociation>
    <Time>1338993585</Time>
    <AffectedSOPClassUID>1.2.840.10008.5.1.4.1.1.2</AffectedSOPClassUID>
    <AffectedSOPInstanceUID>1.3.12.2.1107.5.1.4.54472.300000 ...</AffectedSOPInstanceUID>
    <MessageID>21</MessageID>
    <Direction>1</Direction>
    <MoveOriginatorApplicationEntityTitle>DICOOGLE</MoveOriginatorApplicationEntityTitle>
    <MoveOriginatorMessageID>1</MoveOriginatorMessageID>
    <Status>0</Status>
  </CStore>
</cStores>
```

**Figura 4.12: Retorno de um registo de um comando C-STORE como resposta a uma pesquisa.**

Os restantes métodos do serviço de pesquisa funcionam de forma semelhante aos exemplos aqui apresentados. Obviamente cada um dos métodos devolve os resultados com uma estrutura diferente. Esta depende do tipo de dados armazenados para esse comando (podemos ver no anexo 10.1 que apresenta o modelo da base de dados) e também das *tags* enviadas nos comandos DICOM.



## 5 Sensor de relatórios de ecocardiogramas

Como caso de estudo e uma vez que já dispúnhamos de acesso ao repositório de imagens DICOM de um laboratório de ecocardiografia, foi desenvolvido um sensor de relatórios de ecocardiogramas. Os relatórios aos quais tivemos acesso não seguem nenhuma estrutura normalizada.

O sensor foi desenvolvido para um caso específico, não funcionando assim automaticamente com todos os sistemas deste género, uma vez que se trata de um sistema proprietário. Contudo, desde que os sistemas médicos usem o princípio de codificar a informação como sequências de elementos, o custo de adaptar o sensor para esses novos formatos não será elevado.

O formato dos relatórios será apresentado no subcapítulo seguinte. O acesso a esses relatórios é feito através de uma ligação ODBC, apenas com permissão de leitura, diretamente à base de dados do sistema. Uma alternativa a este método que parece fazer sentido é usar interfaces dos protocolos HL7, no entanto não foi concedido acesso por esta via, provavelmente porque o sistema (feito à medida da instituição) não disponibiliza este tipo de interfaces. Mesmo que essa possibilidade fosse dada, devido às desvantagens apresentadas no subcapítulo 2.3, dificilmente teria sido a abordagem adotada.

### 5.1. Formato dos relatórios

A base de dados do sistema de arquivo de relatórios, continha vários tipos de relatórios, além dos ecocardiogramas. Contudo, apenas foram analisados os relatórios de ecocardiogramas. Conforme foi apresentado em 2.4, existem várias formas de realizar este tipo de estudo, pelo que podemos ter 3 tipos de relatórios: Transtorácico, Transesofágico e Sobrecarga. Cada um destes tipos tem uma estrutura diferente.

A informação de cada um dos relatórios não é apenas um simples registo de uma tabela da base de dados mas está dispersa pelos registos de 4 tabelas diferentes. Desta forma, foi necessário criar uma *query* capaz de reunir a informação de todas as tabelas necessárias. Para que não houvesse o risco de o processo de recolha dos relatórios afetar o desempenho ou mesmo o funcionamento do sistema de informação da instituição, foi criada uma base de dados MySQL com uma estrutura semelhante à original (Oracle) para onde se exportaram temporariamente os dados. Assim, durante o período de desenvolvimento e de testes de desempenho não houve o risco de interferir com o funcionamento do sistema de informação da instituição que, em momentos de emergência, pode ser de importância vital para os pacientes.

Após a obtenção de uma vista dos relatórios completos, um dos campos desse registo, em formato texto, é na verdade uma estrutura XML (Figura 5.1). Mais, cada um dos tipos de ecocardiogramas tem uma estrutura XML diferente. Este sistema de registo de relatórios em formato digital arquiva os relatórios de ecocardiogramas desde 1999.

Um ecocardiograma Transtorácico gera um relatório com cerca de 80 campos de dados. Um Transesofágico gera cerca de 60 campos e, por fim, um ecocardiograma de sobrecarga gera à volta de 140 campos de informação. Parte destes campos são comuns entre os vários tipos de relatórios. A Figura 5.1 apresenta parte da estrutura XML de um relatório de um ecocardiograma Transtorácico. No entanto, o conteúdo exato de cada um dos tipos de relatórios não será detalhado neste documento pois tal não é necessário para expressar o trabalho desenvolvido.

```

<TransthoracicXML xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="htt
<DtCri xsi:nil="true" /><DtAct xsi:nil="true" />
<ObjectState>Unchanged</ObjectState>
<ObjectStateBackup>Unchanged</ObjectStateBackup>
<SimpleEpisodeDefinitionEpisodeType>Exame</SimpleEpisodeDefinitionEpisodeType>
<SimpleEpisodeDefinitionEpisodeId>507289</SimpleEpisodeDefinitionEpisodeId>
<ExamId>40550</ExamId>
<ExamDate>2011-10-25T11:25:47.0987118+01:00</ExamDate>
<ExamProv>Internamentos</ExamProv>
<RelatId>497912</RelatId>
<ExamReqServ>UNID CUID INT CORONARIAS (UCIC)</ExamReqServ>
<ExamReqDoctorAlt />
<Relat> Câmaras cardíacas de dimensões normais.\n Discreta hipertrofia do segmento méd
<MotId>59</MotId>
<MotDescr>Enfarte do miocárdio</MotDescr>

```

Figura 5.1: Porção da estrutura XML de um relatório antes de 2009.

### Relatórios pós 2009

Em 2009, o sistema informático da instituição foi alvo de uma reestruturação. O formato desse campo XML mudou, dando origem a 3 estruturas diferentes, outra para cada tipo de relatório de ecocardiograma. Na verdade a estrutura nova foi colocada em formato texto dentro da tag “Relat” do XML. O campo “Relat” é um campo de texto livre onde inicialmente os profissionais podiam descrever as suas observações de forma não estruturada.

Após a reestruturação do sistema informático foi tomada a opção, quanto a mim errada, de não compatibilizar os novos relatórios com o XML existente mas sim colocar um texto semiestruturado dentro do campo “Relat” com a informação recolhida durante o exame (Figura 5.2). Embora se tenha verificado uma diminuição do número de campos nos relatórios pós 2009, praticamente todos os campos dos novos relatórios estão presentes na estrutura antiga.

```

Nome: AAAAA AAAAA AAAAA AAAAA AAAAA NSC: 9999999
Data de Nascimento: 04-05-1942 (69 anos) Sexo: F
Altura: 1,59 m IMC: 32,44
Peso: 82 kg Sup. Corporal: 1,84 m²

Indicação para exame: Valvulopatia Mitral, Estenose.
Qualidade do exame: Regular.

Medições Modo M e 2D Doppler
Diam Ao 30 mm Ref: <40 Insuficiência Tipo Válvula
Diam AE 52 mm Ref: <40 V.Ao. Mínima Nativa
Área AE (4C) 28 cm² Ref: <18 V.M. Grau II/IV Nativa
Septo IV 7 mm Ref: <11 V.T. Grau I/IV Nativa
Parede Posterior 7 mm Ref: <11 V.P. Grau I/IV Nativa
VE d, Modo M 53 mm Ref: <56-58 Grad máx/méd Área
VE s, Modo M 30 mm Ref: <40 (mmHg) (cm2)
F.Encurt. 43 % Ref: >30 V.M. 11/4 2,0
F.E. (4C) 59 % Ref: >50 Fluxo Pulmonar: Normal.
Massa VE (ASE) 127 g Ref: <162 Derrame Pericárdio: Não.
Índice de Massa 69 g/m² Ref: <124
Vol VE (d, 4C) 100 ml Ref: <105

Análise Segmentar
Sem alterações da contractilidade segmentar.

Relatório
Dilatação moderada da AE (28 cm2). Espessura normal das paredes ventriculares.
Válvula mitral com padrão reumático de estenose: ligeiro espessamento dos
folhetos, com redução da sua motilidade, especialmente do posterior;
insuficiência mitral de grau moderado (grau II- III/IV); área funcional de 2 cm2
e anatomica de 1,8cm2; gradientes máximo e médio de 11 e 4 mmHg.
Ligeiro espessamento das cúspides da válvula aórtica, mantendo boa amplitude de
abertura sistólica; IAo mínima. IT ligeira; PSP calculada de 36 mmHg + PVC.
Fluxo pulmonar normal.

```

Figura 5.2: Porção do conteúdo do campo "Relat" com texto semiestruturado.

### 5.1.1. Processo de recolha dos relatórios

Após estudar a estrutura dos relatórios, foi preciso escolher um método para permitir que o sensor conseguisse usar toda a informação do repositório em tempo útil. Uma das abordagens poderia ser replicar as tabelas da base de dados para outro motor de base de dados e criar os índices mais adequados para a utilização que se queria dar aos dados. Apesar de evitar redundância de informação, usar o motor de base de dados original (Oracle) estava fora de questão, uma vez que isso implicaria sobrecarregar o sistema médico sempre que se estivesse a consultar informação de forma exaustiva com o sistema desenvolvido. Além disso implicaria, no mínimo, alterações aos índices utilizados o que poderia causar anomalias no sistema já existente.

Usar informação redundante tornou-se assim a melhor via a seguir. A alternativa a usar um motor de base de dados, era usar um motor de indexação. Hoje em dia existem motores de base de dados que permitem indexar estruturas XML [47,48]. Contudo, devido à complexidade da estrutura dos relatórios, à existência de vários campos de texto livre e também devido ao facto de os motores de bases de dados relacionais de utilização livre não permitirem indexar XML, a opção de usar um motor de base de dados para guardar uma cópia dos relatórios foi descartada.

A alternativa escolhida foi então usar um motor de indexação que fosse de utilização livre. O motor de indexação escolhido será apresentado no subcapítulo seguinte. Na Figura 5.3 podemos ver os 4 passos do processo de recolha e indexação dos relatórios. Para extrair os campos a partir do texto dos 3 tipos de relatórios pós 2009 foram criadas várias expressões regulares. Desta forma o campo “Relat” do XML não é todo indexado como texto mas apenas a parte que corresponde ao texto escrito pelo médico. No subcapítulo 5.3.1 iremos ver a arquitetura da solução encontrada para compatibilizar as estruturas pré e pós 2009.



Figura 5.3: Processo de recolha e indexação dos relatórios.

## 5.2. Componentes usados

À semelhança do sensor de rede, o sensor de relatórios de ecocardiogramas também é desenvolvida em Java. A Figura 5.4 apresenta o diagrama de componentes do sensor de rede.

Os componentes Jersey e Grizzly, que foram apresentados no capítulo relativo ao sensor de rede, são usados mais uma vez para criar e fornecer os *web services* que servirão de interface entre o sensor de relatórios e o exterior.

O componente MySQL, também usado no sensor anterior, é o componente utilizado no processo de indexação para obter a informação relativa aos relatórios. Conforme foi explicado anteriormente, para o processo de indexação foi criada uma base de dados MySQL temporária com uma estrutura semelhante à original, de forma a não se correr o risco de interferir com o funcionamento do sistema de informação.

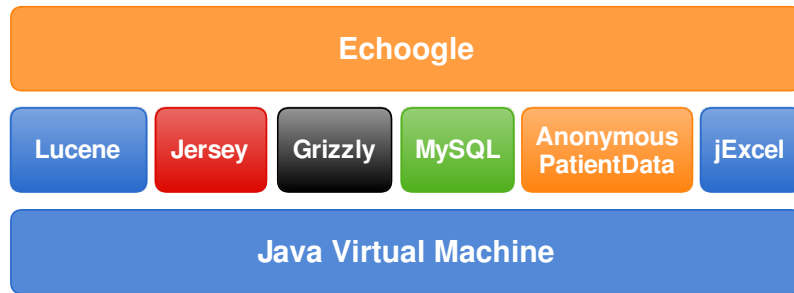


Figura 5.4: Diagrama de componentes do sensor de relatórios.

O motor de indexação escolhido para indexar todos os campos relativos aos relatórios de ecocardiogramas foi o Lucene [19]. O Lucene é uma biblioteca originalmente desenvolvida em Java (já foi implementada em várias outras linguagens) que foi adotada pela fundação Apache e pode ser utilizada para dotar uma aplicação de capacidades de indexação e pesquisa [18]. Usar a biblioteca Lucene diretamente no código fonte da aplicação não é, por vezes, a solução mais simples. Existe uma ferramenta chamada Apache Solr [49] que usa o Lucene internamente e permite aos programadores usarem o Lucene a partir de qualquer linguagem atual sem terem que lidar com os pormenores de integração do Lucene nas suas aplicações. Para isso disponibiliza uma API *REST-like* que suporta tanto XML como JSON. O Solr funciona como um servidor de indexação/pesquisa autónomo usando um servidor de aplicações como o Tomcat [50], por exemplo.

A opção por usar a biblioteca Lucene diretamente na aplicação e não usar o Solr foi tomada porque: usando o Lucene diretamente podemos ter acesso a todas as suas potencialidade e não há restrição apenas às funcionalidades que o Solr oferece. Além disso, esta opção, apesar de aumentar a complexidade da implementação, diminui a complexidade da instanciação de todo o sistema distribuído. Usando o Solr como solução para indexação, criar-se-ia mais um componente distinto dos restantes no sistema distribuído.

A biblioteca AnonymousPatientData é usada com o mesmo objetivo e da mesma forma que foi apresentada no capítulo referente ao sensor de rede.

O jExcel [51] é uma biblioteca para Java que implementa uma API que permite ler e escrever folhas de cálculo em formatos compatíveis com o Microsoft Excel e aplicações similares. Esta biblioteca é usada especialmente para o serviço de localização, conforme iremos ver no subcapítulo 5.4.2.

### 5.3. Arquitetura

O sensor de relatórios de ecocardiogramas está organizado em 9 *packages*. A lista seguinte apresenta o nome dos *packages* e uma pequena descrição de cada um deles:

- **Core:** Este *package* contém a classe principal da aplicação, a classe com as definições, uma lista de *stopwords* a usar na análise dos campos de texto dos relatórios e uma classe para carregar essa lista para o analisador do Lucene.
- **GUI:** Contém as classes que implementam a interface gráfica do sensor. A interface gráfica do sensor é minimalista. O principal objetivo do sensor é disponibilizar uma boa interface ao nível dos *web services* para o resto do sistema. Mais uma vez, este sensor cria um ícone na bandeja do sistema (*system tray*) e permite que não seja necessário visualizar nenhuma janela enquanto o sensor está a ser executado, deixando visível apenas o ícone na bandeja.

- **Database:** Este *package* contém apenas duas classes que interagem com a base de dados MySQL. Uma delas que permite importar para a base de dados temporária MySQL os ficheiros exportados a partir do Oracle. A outra classe, "TGetInfo" permite criar uma *thread* para obter os exames a partir da base de dados e colocá-los na fila de indexação.
- **Export:** Este *package* contém as classes responsáveis por realizar o pré-processamento necessário para extrair algumas métricas ou dados que não se conseguem extrair em tempo-real.
- **Index:** As classes relativas ao processo de indexação estão localizadas neste *package* ou nos seus *subpackages*, **index.documents** e **index.fields**. Estas classes usam o Lucene para indexar os campos obtidos a partir dos relatórios e também efetuar pesquisas sobre esse índice.
- **Location:** Contém a classe usada para ler e guardar a informação relativa aos distritos, concelhos e freguesias portuguesas necessárias para o serviço de localização (subcapítulo 5.4.2). Contém também o ficheiro *.xls* com a informação necessária que foi obtido a partir do Portal das Finanças português [52].
- **Security:** O conteúdo deste *package* é semelhante ao *package* "Security" que foi apresentado no sensor de rede, no subcapítulo 4.3, pois serve exatamente o mesmo propósito.
- **WebServices:** Contém as classes que implementam os *web services* do sensor. Os *web services* do sensor de relatórios serão detalhados no subcapítulo 5.4.
- **XML:** Contém as classes que, através de anotações do JAXB, servem para transformar a estrutura XML dos relatórios em objetos Java sem recorrer a *parsers*, como foi descrito no subcapítulo 5.1. Além disso contém a classe "ParseRegexPatterns" que têm todas as expressões regulares usadas para extrair os campos a partir do texto dos relatórios posteriores a 2009.

Embora só tenha que ser feito na totalidade na primeira utilização do sensor, a indexação dos relatórios é um processo exaustivo e demora um tempo considerável a concluir, conforme será apresentado no capítulo 7. Este é um processo que envolve grandes quantidades I/O (*Input/Output*) e também de processamento. Para tirar partido da capacidade de processamento disponível atualmente, principalmente nos sistemas *multi-core* mas também devido ao tempo de espera de I/O, o mecanismo de indexação foi desenvolvido usando *multi-threading*.

Na Figura 5.5 podemos ver um diagrama com as classes usadas durante a indexação de relatórios com a exceção das classes do *package* XML, que constam da Figura 5.7. Durante o processo de indexação são usadas duas *threads*.

A *thread* "TGetInfo" obtém os exames a partir da base de dados e coloca-os na fila de indexação (primeiro passo da Figura 5.3). A *thread* "TIndex" usa os exames colocados na fila, extrai todos os campos e indexa-os usando o Lucene (restantes passos da Figura 5.3). A classe "IndexQueue" é a que contém a fila de documentos à espera de serem indexados. Esta classe tem todos os seus métodos sincronizados, implementando assim um monitor, para que não existam problemas de concorrência que ponham em causa a consistência da informação. A segunda *thread* exige mais capacidade de processamento e também muito esforço de I/O para escrever o índice no disco. Para que a fila de indexação não cresça indefinidamente no caso da primeira *thread* obter os exames mais rapidamente do que a segunda os consegue processar e indexar, foi criada a noção de *indexing step* (passo de indexação).

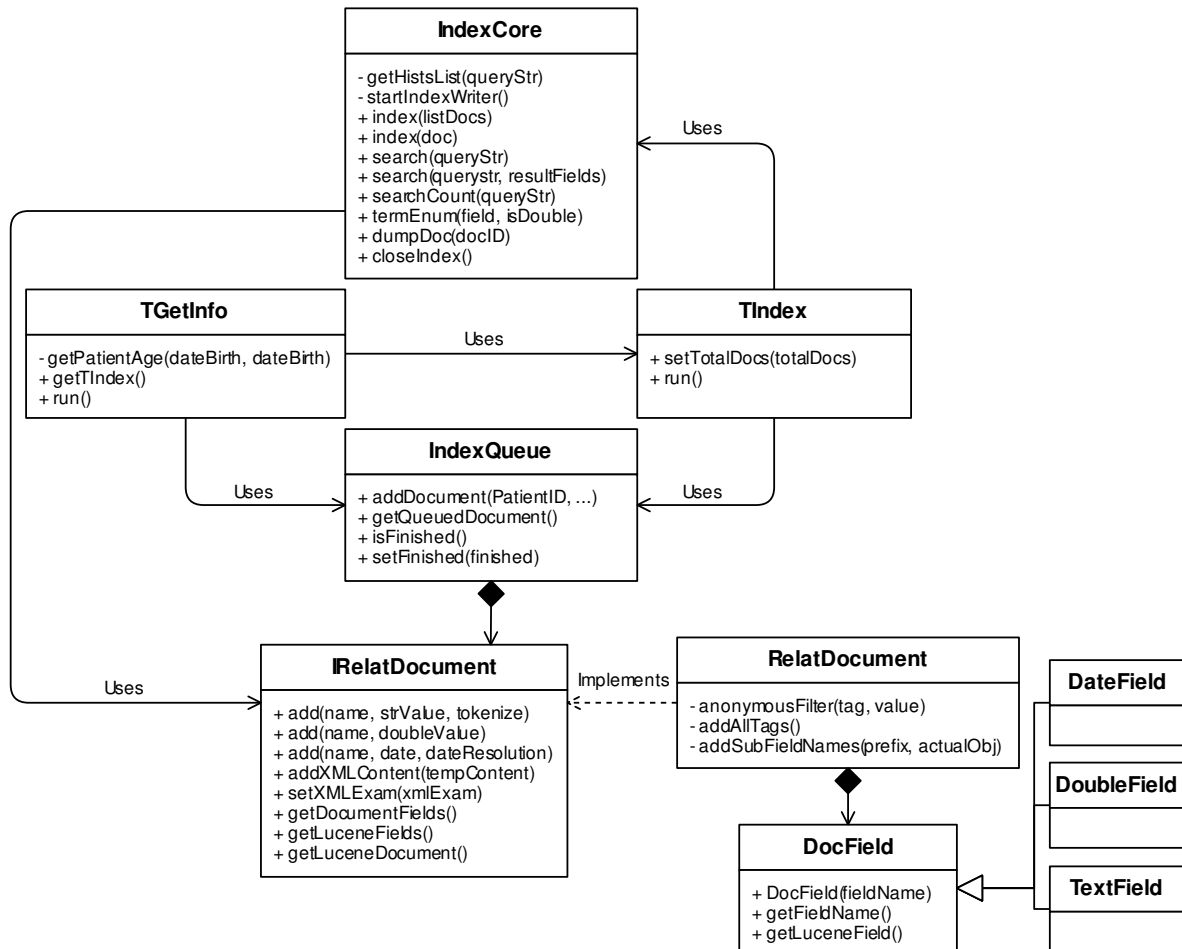


Figura 5.5: Diagrama de classes com as classes usadas na indexação de relatórios com omissão das classes usadas para obter o conteúdo das estruturas XML (package “XML”).

O *indexing step* influencia duas coisas no processo de indexação. A primeira é o tamanho máximo da fila de indexação e a segunda é a frequência com que é chamado o método *commit* da classe “IndexWriter” do Lucene. A fila de indexação terá no máximo três vezes o número do *indexing step*. Caso a fila chegue a este limite, a *thread* “TGetInfo” ficará suspensa no método *addDocument* da classe “IndexQueue” até que a fila volte a ter espaço disponível. O método *commit* é invocado periodicamente para tornar as alterações feitas ao índice criado pelo Lucene persistentes, libertando assim a memória utilizada temporariamente. Este método é muito exaustivo e por isso não deve ser chamado sempre que um novo relatório é indexado. O valor do *indexing step* pode ser definido através das definições da aplicação (Figura 5.6) e, desta forma, o utilizador poderá escolher entre consumir menos memória ou aumentar o desempenho durante a indexação dos relatórios.

Os campos dos relatórios são traduzidos em apenas 3 tipos diferentes a serem indexados: data, texto ou *double* para todo o tipo de informação numérica. Os campos de texto passam por um processo de atomização (separação em *tokens*) onde são também removidas as *stopwords* que constam de uma lista predefina. O utilizador pode escolher guardar o conteúdo dos campos de texto, se quiser aceder a todo o conteúdo dos relatórios apenas através do índice, ou apenas indexá-los não guardando o seu conteúdo exato.

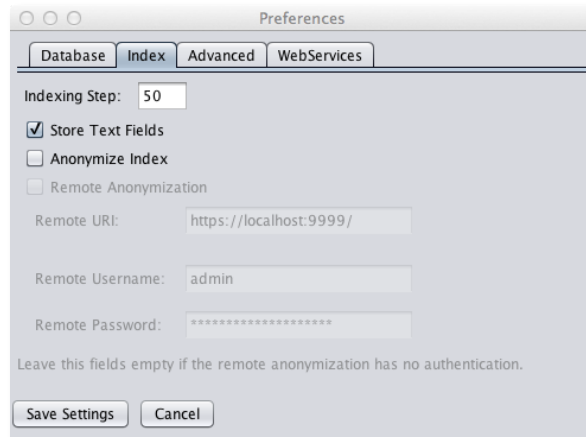


Figura 5.6: Preferências de indexação do sensor de rede.

### 5.3.1. Normalização e processamento das estruturas dos relatórios

Conforme já foi referido anteriormente, em 2009 a estrutura dos 3 tipos de relatórios mudou. Assim, para cumprir os objetivos do trabalho foi necessário normalizar os novos relatórios na estrutura antiga pois esta é mais adequada para indexar com base em campos. Em 5.1.1 foi referido que foram criadas diversas expressões regulares para obter a informação relativa aos campos do relatório a partir do seu texto semiestruturado. Essas expressões regulares foram todas concentradas numa única classe, de forma a que pudessem ser reutilizadas em todas as classes que obtêm os campos do relatório a partir do texto semiestruturado (Figura 5.7).

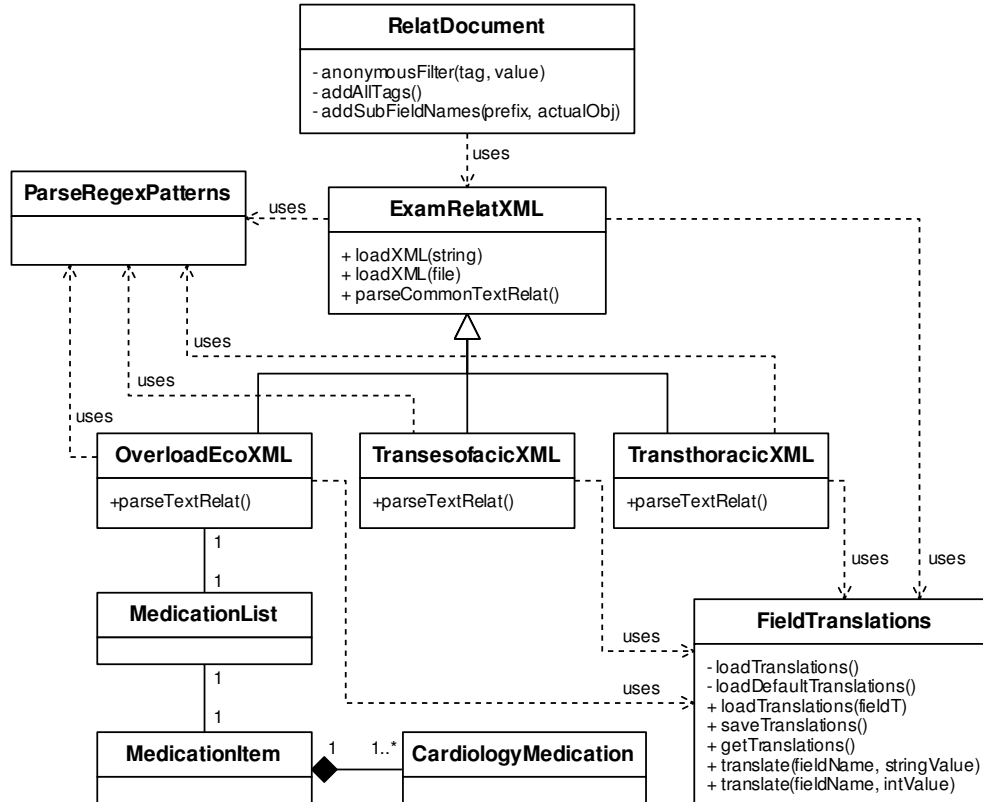


Figura 5.7: Diagrama de classes do *package* XML com adição das classes RelatDocument e FieldTranslations.

Na estrutura antiga, muitos dos campos do relatório em XML eram representados por um número. Esse número correspondia a uma *string* que era apresentada na interface gráfica do sistema de registo dos relatórios. Para que os utilizadores do sistema desenvolvido possam ter acesso ao significado dos campos, foi necessário guardar essas correspondências entre nomes e números. Para isso foi criada a classe *FieldTranslations* (Figura 5.7) que consegue ler e escrever essas traduções a partir de um ficheiro XML na raiz da aplicação. Também é possível ver e alterar essas correspondências através dos *web services* do sensor, como iremos ver mais adiante em 5.4.

Na nova estrutura pós 2009, vários campos do relatório presentes no texto não estão no seu formato codificado, isto é, no relatório apenas está o texto correspondente ao valor numérico dos campos. Desta forma é necessário traduzir esses nomes para os números correspondentes de forma a uniformizar os relatórios pré e pós 2009, usando a classe *FieldTranslations*.

#### 5.4. Web services

À semelhança do sensor de rede apresentado anteriormente, este sensor de relatórios de ecocardiogramas também disponibiliza vários *web services* REST. Estes *web services* também são capazes de manipular estruturas XML e JSON como parâmetros de entrada ou de retorno, dependendo dos cabeçalhos do pedido HTTP.

Este sensor também disponibiliza o serviço de pseudo-anonimização para que possa ser usado como *master* de anonimização, tal como foi descrito no subcapítulo 3.4.2.

##### 5.4.1. Serviço de gestão do sensor

O serviço de gestão do sensor tem métodos para ver e editar as definições do sensor de relatórios, para obter e alterar as correspondências entre valores numéricos e textuais dos campos e também para traduzir os valores dos campos com base na lista de correspondências do sensor. A listagem dos métodos deste serviço encontra-se na Tabela 5.1.

Tabela 5.1: Métodos do serviço de gestão do sensor.

<i>Método</i>	<i>Parâmetros de entrada</i>	<i>Método HTTP</i>	<i>Descrição</i>
/ManageApp/settings	-	GET	Devolve uma estrutura com as definições do sensor
/ManageApp/settings	Estrutura com as definições do sensor	POST	Guarda as novas definições do sensor
/ManageApp/translations	-	GET	Obtém as correspondências entre valores textuais e numéricos dos campos
/ManageApp/translations	Estrutura com a correspondência de valores	POST	Guarda as novas correspondências entre valores
/ManageApp/translations/{fieldName}	Nome do campo	GET	Obtém as correspondências de um único campo
/ManageApp/translate2Int	Nome e valor	POST	Retorna o valor numérico



	textual do campo		que corresponde ao valor textual
/ManageApp/translate2Str	Nome e valor numérico do campo	POST	Retorna o valor textual que corresponde ao valor numérico enviado

A Figura 5.8 apresenta um exemplo da estrutura que é retornada e aceite pelos dois métodos “/ManageApp/translations” usando HTTP GET e POST. Essas traduções são importante para que a indexação dos relatórios pré e pós 2009 seja feita uniformemente e para que na interface gráfica do sistema desenvolvido os utilizadores tenham acesso à correspondência dos valores dos campos.

```

{
  "listTranslations": [{
    "fieldName": "TriCalcif",
    "translations": [{
      "key": "",
      "value": "0"
    }, {
      "key": "Não",
      "value": "1"
    }, {
      "key": "Moderado",
      "value": "3"
    }, {
      "key": "Marcado",
      "value": "4"
    }, {
      "key": "Ligeiro",
      "value": "2"
    }
  ]
},
...
]
}

```

Figura 5.8: Parte da estrutura com a lista de correspondências entre os valores numéricos e textuais dos campos em formato JSON.

#### 5.4.2. Serviço de localização

Para fazer certos estudos demográficos sobre os pacientes da instituição, é necessário obter a sua proveniência. Nos casos em que tal estava registado no sistema original, é possível obter o distrito, concelho e freguesia do paciente ao qual um relatório corresponde. O distrito, concelho e freguesia encontram-se registados pelo seu código e não pelo nome. Para identificar uma freguesia são usados seis dígitos, dois para identificar o distrito, dois para identificar o concelho e os restantes dois identificam a freguesia.

O portal das finanças [52] contém um ficheiro .xls que tem as traduções entre estes códigos e os nomes. Este ficheiro foi então utilizado para criar este serviço de obtenção da localização com o métodos identificados na Tabela 5.2.

Tabela 5.2: Métodos do serviço de localização.

<i>Método</i>	<i>Parâmetros de entrada</i>	<i>Método HTTP</i>	<i>Descrição</i>
/location/dist/{cod}	Código com pelo menos dois dígitos para identificar o distrito	GET	Retorna o nome do distrito

/location/conc/{cod}	Código com pelo menos quatro dígitos para identificar o concelho	GET	Retorna o nome do concelho
/location/freg/{cod}	Código com pelo menos seis dígitos para identificar a freguesia	GET	Retorna o nome da freguesia
/location/listDistritos	-	GET	Retorna uma lista com os nomes e códigos de todos os distritos portugueses
/location/listConcelhos/{cod}	Código com pelo menos dois dígitos para identificar o distrito	GET	Retorna uma lista com os nomes e códigos dos concelhos do distrito
/location/listFreguesias/{cod}	Código com pelo menos quatro dígitos para identificar o concelho	GET	Retorna uma lista com os nomes e códigos das freguesias do concelho

A Figura 5.9 apresenta o resultado da invocação do método “/location/listDistritos”. No capítulo de apresentação dos resultados, iremos perceber a importância deste serviço para realizar estudos demográficos.

```

{
  "Item": [
    {
      "name": "AVEIRO",
      "value": "01"
    },
    {
      "name": "BEJA",
      "value": "02"
    },
    {
      "name": "BRAGA",
      "value": "03"
    },
    {
      "name": "BRAGANÇA",
      "value": "04"
    },
    {
      "name": "CASTELO BRANCO",
      "value": "05"
    },
    ...
  ]
}

```

Figura 5.9: Parte inicial da lista de distritos portugueses em formato JSON.

### 5.4.3. Serviço de pesquisa

Este é o serviço mais importante do sensor de relatórios de ecocardiogramas (Tabela 5.3). Existem dois tipos de *queries* que podem ser feitas ao indexador. Podem ser enviadas *queries* em texto livre sem uma estrutura pré definida. Para refinar os resultados o utilizador pode também enviar *queries* avançadas que têm uma sintaxe própria. No anexo 10.3 podemos ver alguns exemplos dos dois tipos de *queries*.

A lista de resultados retornada nos métodos de pesquisa não contém o valor de todos os campos indexados dos relatórios. Para cada resultado são retornados apenas os seguintes campos: Nome do Paciente, ID do Paciente, Data do Exame, Tipo do Exame, ID do Relatório e *Accession Number*. Estes campos são suficientes para construir a interface de apresentação de resultados (conforme iremos ver mais adiante) e foram escolhidos para acelerar o processo

de retorno de resultados à pesquisa. Caso, para um resultado específico, o utilizador queira ver o valor de todos os campos indexados, pode utilizar o método “/Search/dump” do serviço de pesquisa, conforme podemos ver na Tabela 5.3.

**Tabela 5.3: Métodos do serviço de pesquisa do sensor de relatórios.**

<b>Método</b>	<b>Parâmetros de entrada</b>	<b>Método HTTP</b>	<b>Descrição</b>
/Search/getAllTags	-	GET	Retorna a lista todas os campos indexados pelo sensor
/Search/search/{query}	Query de pesquisa	GET	Retorna a lista de resultados
/Search/search	Query de pesquisa	POST	Retorna a lista de resultados
/Search/advSearch/{query}	Query avançada de pesquisa	GET	Retorna a lista de resultados
/Search/advSearch	Query avançada de pesquisa	POST	Retorna a lista de resultados
/Search/searchCount/{query}	Query de pesquisa	GET	Retorna o número de resultados obtidos como resposta à query
/Search/searchCount	Query de pesquisa	POST	Retorna o número de resultados obtidos como resposta à query
/Search/advSearchCount/{query}	Query avançada de pesquisa	GET	Retorna o número de resultados obtidos como resposta à query avançada
/Search/advSearchCount	Query avançada de pesquisa	POST	Retorna o número de resultados obtidos como resposta à query avançada
/Search/dump/{docID}	ID do relatório	GET	Retorna o valor de todos os campos indexados para um relatório
/Search/enumTerm/{term}/{isNumber}	Nome do campo e indicação se é um campo numérico ou não	GET	Enumera todos os valores indexados para o campo indicado

Na Figura 5.10 podemos ver a estrutura que é retornada em resposta às pesquisas em formato XML. No exemplo apresentado o índice criado estava anonimizado. Daí o ID do paciente e o *Accession Number* terem um formato tão longo e o nome do paciente não ter aparente sentido.

```

<Results>
  <nResults>1</nResults>
  <patient PatientID="0afelb21-fe74-425a-a0d1-59a91494cdb7" PatientName="SHELLEY Paul LOLA">
    <examDate ExamDate="20070124">
      <type ContentType="ECO SOBRECARGA">
        <exam ResultID="68392" AccessionNumber="339f629a-7292-4130-9c3b-66546662db4e"/>
      </type>
    </examDate>
  </patient>
</Results>

```

Figura 5.10: Retorno do método de pesquisa com um único resultado em formato XML.

A resposta da Figura 5.10 foi obtida em resposta à *query* avançada “PatientName:Paul AND NOT ContentType:ETT” enviada para o método “/Search/advSearch” deste serviço de pesquisa. A estrutura retornada pelos métodos de pesquisa está organizada de forma hierárquica: Paciente – Data do Exame – Tipo de Exame – Exame. Desta forma, se houverem múltiplos resultados com alguns destes campos em comum, a estrutura irá refletir isso, facilitando assim o processo de tratamento dos resultados e evitando redundância de dados na estrutura.

#### 5.4.4. Serviço de pré-processamento

À semelhança do que foi apresentado em 3.5.1, o serviço de pré-processamento do sensor de relatórios de ecocardiogramas tem como objetivo efetuar os cálculos necessários para suportar estudos que relacionam dados vindos de outros sensores. Não é possível realizar estes estudos em tempo real porque os cálculos demoram um tempo considerável (cerca de 45 minutos com as condições descritas no subcapítulo 7.2).

A Tabela 5.4 contém a listagem de métodos do serviço de pré-processamento. O ficheiro em formato CSV será posteriormente utilizado pela aplicação Web para que os utilizadores possam efetuar os estudos pretendidos relacionando com dados vindos de outros sensores.

Tabela 5.4: Métodos do serviço de pré-processamento do sensor de relatórios de ecocardiogramas.

Método	Parâmetros de entrada	Método HTTP	Descrição
/Export/getState	-	GET	Retorna o estado atual do módulo de pré-processamento
/Export/percentage	-	GET	Retorna a percentagem de conclusão do processamento
/Export/startCalc	-	GET	Inicia os cálculos de pré-processamento
/Export/stopCalc	-	GET	Interrompe os cálculos de pré-processamento
/Export/download	-	GET	Retorna o ficheiro CSV com o resultado do pré-processamento

## 6 Medical Analysis Web

Nos capítulos anteriores foram apresentados vários sensores de informação médica, a sua arquitetura e interface baseada em *web services*. Neste capítulo será apresentada a aplicação Web, “Medical Analysis Web”, que integra os dados recolhidos pelos vários sensores e faz interface com o utilizador final.

### 6.1. Visão geral

Esta aplicação dá aos utilizadores a possibilidade de pesquisar sobre as várias fontes de informação, utilizando para o efeito uma interface única. Várias métricas sobre os serviços prestados, obtidas a partir da informação contida nos sensores, são também disponibilizadas nesta aplicação.

Existem dois papéis de utilizadores no site, os utilizadores normais e os administradores. A Figura 6.1 apresenta o diagrama com os casos de utilização da aplicação Web desenvolvida relacionados com a pesquisa de informação nos vários tipos de sensores. Além de pesquisar nos vários sensores, os utilizadores e administradores podem ver uma lista de todos os campos de um resultado, exportar esses resultados e, no caso de serem resultados relativos a imagens médicas, pode pré-visualizar e descarregar as imagens médicas relativas aos resultados. É importante notar que os sensores (atores do lado direito do diagrama) podem não ser todos instanciados ou podem existir múltiplas instâncias de cada tipo.

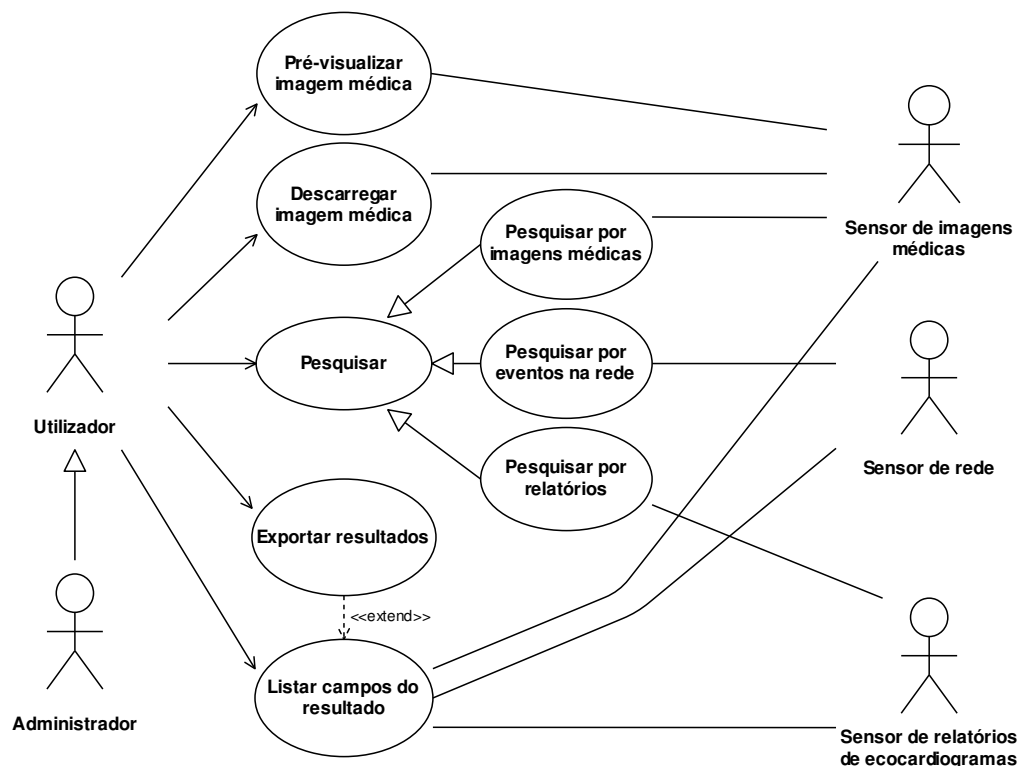


Figura 6.1: Casos de utilização relacionados com a pesquisa de informação nos sensores.

No diagrama da Figura 6.2 podemos ver os casos de utilização relacionados com a as estatísticas e métricas extraídas pela aplicação Web desenvolvida. As métricas extraídas pela

aplicação serão detalhadas no subcapítulo 6.3.4 e no capítulo de apresentação dos resultados deste trabalho. A maioria das métricas e indicadores extraídos são apresentados em forma de gráficos mas, para cada métrica, o utilizador pode de escolher exportar uma folha de cálculo com os dados para os tratar externamente.

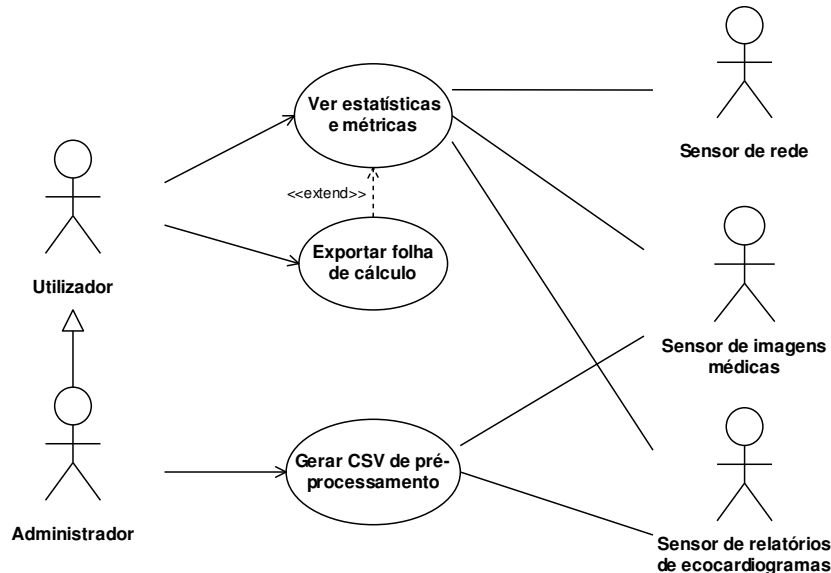


Figura 6.2: Casos de utilização relacionados com as estatísticas e métricas extraídas pela aplicação.

Os casos de utilização relacionados com a gestão e utilização das contas de utilizadores estão representados no diagrama da Figura 6.3. Estes são os casos de utilização mais gerais pois são comuns a muitas outras aplicações Web. Uma vez que esta aplicação lida com dados sensíveis, todas as contas têm que ser ativadas por um administrador, antes de poderem ser utilizadas.

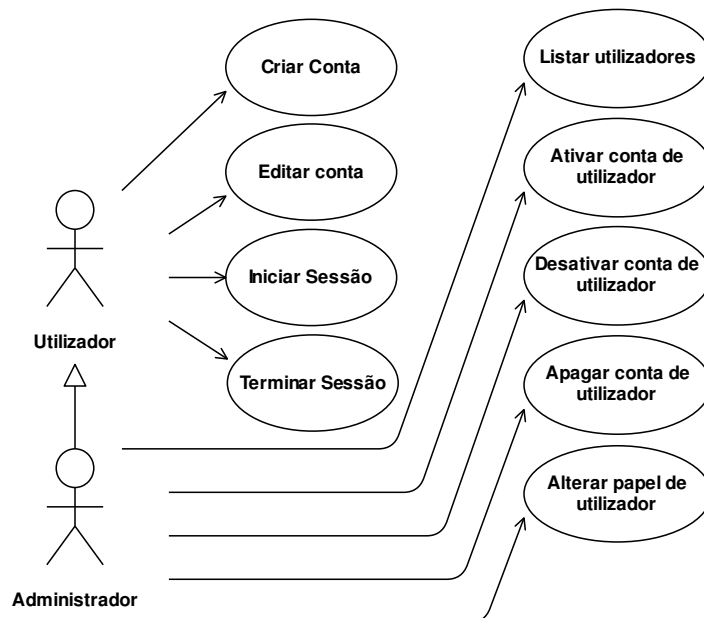


Figura 6.3: Casos de utilização relacionados com as contas de utilizadores.

A Figura 6.4 apresenta o diagrama com os restantes casos de utilização, que estão apenas disponíveis aos administradores da aplicação. Estes estão relacionados com a gestão

de preferências do Medical Analysis Web, preferências dos sensores, regras para alerta no sensor de rede e com a gestão da lista dos sensores a utilizar.

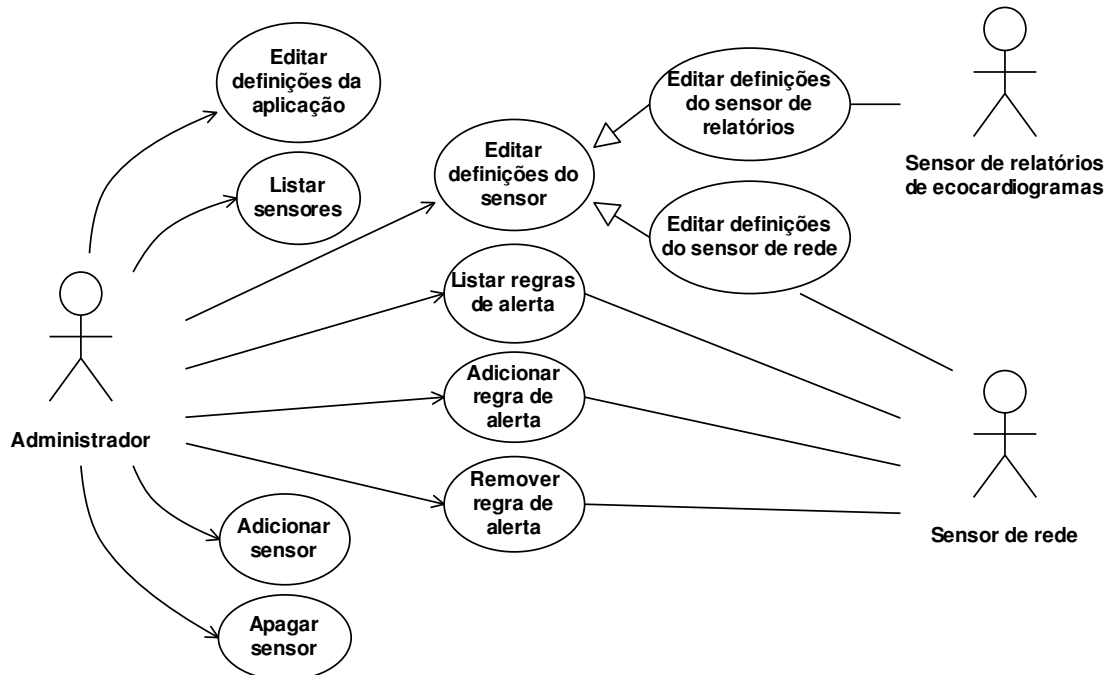


Figura 6.4: Casos de utilização apenas disponíveis aos administradores.

## 6.2. Arquitetura

Uma aplicação ou site Web necessita de ser colocada num servidor Web para que possa ser acessível aos utilizadores da World Wide Web. Hoje em dia, grande parte dos sites Web gera conteúdos dinamicamente. Os utilizadores podem modificar ou acrescentar informação à medida que vão utilizando a aplicação Web. Tornou-se assim vulgar ligar a aplicação Web a um motor de Base de Dados.

A aplicação Web apresentada neste capítulo foi desenvolvida utilizando maioritariamente a linguagem PHP. O PHP é uma linguagem, do lado do servidor, muito usada para desenvolvimento de aplicações Web. O servidor Web escolhido foi o Apache, visto ser o mais utilizado à escala mundial [53], ser de utilização livre, suportar PHP, e ter versões para os sistemas operativos mais populares na atualidade, não limitando assim a plataforma que possa vir a ser usada para alojar a aplicação Web. O motor de base de dados escolhido foi o MySQL, porque também é largamente usado, de livre utilização e tem versões para a maioria dos sistemas operativos. A Figura 6.5 apresenta os mecanismos de comunicação utilizados pela aplicação Web desenvolvida, o servidor Web e o motor de base de dados escolhidos. O servidor Web e o servidor de base de dados podem estar alojados na mesma máquina física ou em máquinas distintas.

Para acelerar o desenvolvimento desta aplicação, foi tomada a opção de utilizar diversas ferramentas *open-source* permitindo assim concentrar esforços na análise dos dados obtidos pelos sensores. Desta forma, foi possível concentrar mais esforço naqueles que eram os principais objetivos da aplicação Web.

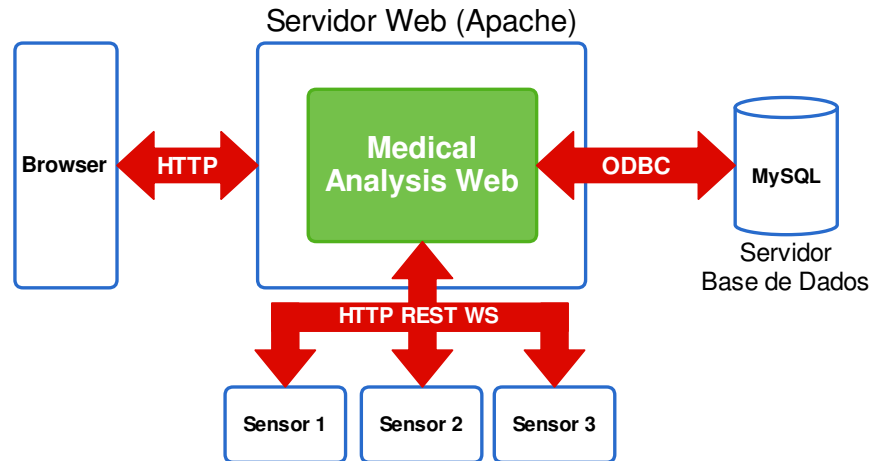


Figura 6.5: Mecanismos de comunicação do Medical Analysis Web.

### 6.2.1. CakePHP

Frequentemente, são utilizadas *frameworks* de *software* no sentido de acelerar o desenvolvimento das aplicações e ajudar a garantir a sua robustez. Uma *framework* é uma peça de *software* que implementa funcionalidades genéricas que podem ser reutilizadas, configuradas e adaptadas para construir assim *software* específico. Usar *frameworks open-source* pode ajudar a garantir a robustez de uma aplicação. Se a base de utilizadores dessas mesma *framework* for alargada, a deteção e correção de falhas será feita em curtos espaços de tempo, diminuindo a probabilidade de existirem falhas críticas nas funcionalidades genéricas da aplicação.

A *framework* PHP escolhida para desenvolver a aplicação Web foi o CakePHP [54]. O CakePHP é uma das *frameworks* PHP mais utilizadas, com uma grande base de pessoas envolvidas no seu desenvolvimento e com um grande número de utilizadores [55]. O CakePHP baseia-se numa arquitetura MVC (Model-View-Controller) que permite isolar os 3 grandes componentes de uma aplicação: dados, lógica e interface de utilizador (Figura 6.6). Através deste padrão MVC, é possível alterar completamente o modo como os dados são apresentados sem que isso exija qualquer alteração à lógica implementada para processar os pedidos HTTP que chegam ao servidor.

A Figura 6.6 apresenta o ciclo de processamento típico de um pedido no CakePHP. O cliente é habitualmente um *browser*. O pedido é recebido pelo *Dispatcher* que, consoante o URL, encaminha o pedido para o *Controller* correspondente. Tipicamente um site ou aplicação Web tem vários controladores. O *Controller* processa o pedido e interage com o *Model* no sentido de obter, validar ou atualizar os dados e, finalmente, retorna o resultado para uma *View*. O resultado enviado para as *Views* é um conjunto de variáveis que vão ser aplicadas de forma a retornar para o utilizador um documento no formato esperado. O documento produzido pela *View* é então entregue ao cliente do pedido. A camada de visualização (conjunto das *Views*) não está limitada a produzir documentos em formato HTML, pode produzir XML ou documentos com os mais variados formatos, dependendo da implementação feita.

No CakePHP, cada uma das 3 camadas do MVC é formada por conjuntos de classes PHP. Por exemplo, para gerir os utilizadores de um site é necessário desenvolver um controlador, um modelo para interagir com a Base de Dados que contém a informação dos utilizadores e um conjunto de *Views* para registar, editar o perfil, iniciar a sessão e listar os utilizadores. O CakePHP têm suporte nativo para os motores de base de dados mais comuns (MySQL, Postgres, SQLite, Microsoft SQL Server, Oracle) mas é possível construir novos *drivers* para



persistência de dados, como por exemplo para diretórios LDAP (*Lightweight Directory Access Protocol*).

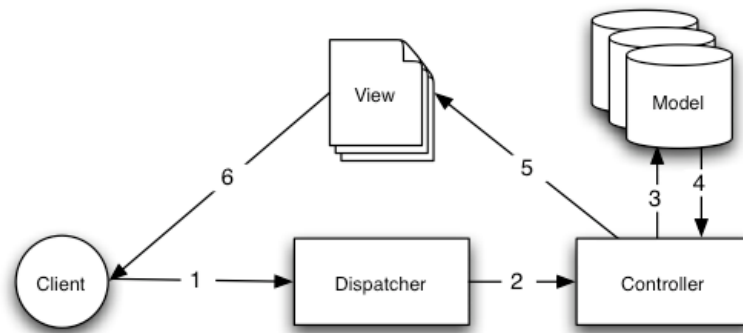


Figura 6.6: Arquitetura MVC usada pelo CakePHP [54].

Como foi referido anteriormente, o motor de base de dados escolhido foi o MySQL. A base de dados criada para fazer a persistência dos dados necessários aos modelos seguiu regras de nomenclatura do CakePHP para facilitar a criação desses mesmos modelos. A Figura 6.7 apresenta o modelo físico da base de dados criada. Conforme podemos ver, a base de dados é minimalista, uma vez que grande parte dos da informação utilizada pela aplicação Web está alojada nos sensores e não na base de dados da aplicação Web.

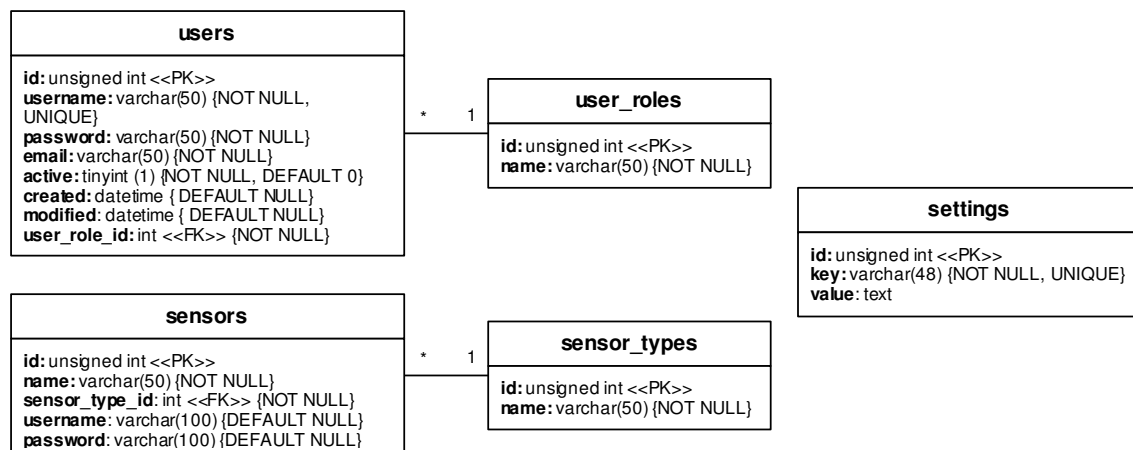


Figura 6.7: Modelo físico da Base de Dados do Medical Analysis Web.

Para desenvolver aplicações utilizando o CakePHP é necessário passar por um processo de adaptação e aprendizagem que limita um pouco a produtividade. No entanto, após ultrapassar essa fase, a produtividade aumenta porque permite aos programadores dedicarem-se a coisas específicas da sua aplicação ao invés de perderem tempo a fazer mais coisas básicas e genéricas.

### 6.2.2. AJAX

O desenvolvimento Web tem evoluído nos últimos anos no sentido de não ser preciso recarregar a totalidade das páginas sempre que o utilizador executa uma ação na interface. Isto é possível graças a uma tecnologia chamada "AJAX", que é suportada por todos os browsers atuais. Esta tecnologia permite construir aplicações Web que conseguem enviar e receber dados do servidor sem recarregar a página, atualizando apenas as partes necessárias (Figura 6.8). Ao desenvolver esta aplicação houve o cuidado de utilizar esta tecnologia para evitar recarregamentos das páginas em todas as ações dos utilizadores.

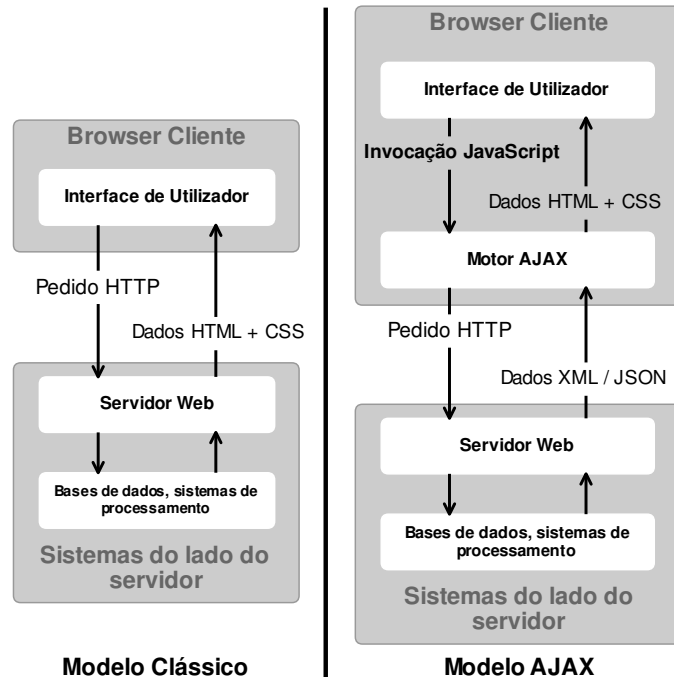


Figura 6.8: Modelos de desenvolvimento de aplicações Web. À esquerda o modelo clássico e à direita o novo modelo AJAX (adaptado de [56]).

### 6.2.3. Outros componentes usados

Além da *framework* apresentada anteriormente foram utilizados outros componentes, que serão aqui apresentados. A apresentação dos componentes utilizados será feita em duas partes. A primeira parte apresentará os componentes, além do CakePHP, que são utilizados do lado do servidor e a segunda aqueles que são carregados para o browser do cliente e executados lá.

#### Lado do Servidor

Os componentes utilizados do lado do servidor são:

- **PHPExcel** [57]: é uma biblioteca PHP que permite ler e escrever folhas de cálculo nos formatos CSV, XLS, XLSX, entre outros. É a biblioteca usada para implementar a funcionalidade de exportação de resultados de pesquisa e também dos dados usados para criar os gráficos apresentados na página.
- **PHPMailer** [58]: é uma biblioteca PHP para transferência de emails. Embora o PHP já tenha algum suporte para enviar emails nativamente, esta biblioteca acrescenta funcionalidades e vem dar maior flexibilidade às aplicações que pretendem transferir emails. No caso da aplicação desenvolvida, é utilizada para enviar os emails de confirmação de ativação de conta por parte do administrador do sistema e alertar o administrador sempre que um novo utilizador se regista.

#### Lado do Cliente

Do lado do cliente, são usados os seguintes componentes:

- **jQuery** [59]: é uma biblioteca JavaScript muito usada atualmente. Ela simplifica a interação com a informação contida no documento HTML, manipulação de eventos e interações AJAX com os servidores. Esta biblioteca

suporta diversos *plugins* para estender as suas funcionalidades. Alguns dos próximos componentes apresentados são *plugins* do jQuery.

- **jQuery File Tree** [60]: como o nome indica, o jQuery File Tree é um *plugin* para o jQuery. Este permite criar um elemento na interface gráfica que possibilita a navegação no sistema de ficheiros do servidor utilizando um esquema hierárquico em árvore. Para ser possível navegar pelo sistema de ficheiros do servidor, este *plugin* depende de scripts conectores que executem do lado do servidor. Existem scripts desenvolvidos para as linguagens mais usadas do lado do servidor. Contudo, o interesse por este *plugin* não teve nada a ver com a navegação pelo sistema de ficheiros do servidor mas sim com a sua capacidade de navegação hierárquica em árvore. Foi desenvolvido um script conector para PHP que permite navegar entre os resultados de uma pesquisa (Figura 6.17). Iremos ver mais detalhes no subcapítulo 6.3.3 que apresenta a interface gráfica da aplicação Web.
- **DataTables** [61]: o DataTables também é um *plugin* para o jQuery que permite criar tabelas que suportam diversas interações com o utilizador como, paginação de tamanho variável, ordenação de colunas, filtragem de dados, carregamento dados por AJAX, entre outras coisas. Na aplicação desenvolvida foi usado para apresentar todos os campos relativos a um resultado de pesquisa (Figura 6.18), como poderemos no subcapítulo 6.3.3.
- **Twitter Bootstrap** [62]: o Bootstrap é uma biblioteca JavaScript em conjunto com alguns CSS que são usados para ajudar a desenvolver a interface gráfica de um site Web e implementar pequenas funcionalidades também elas relacionadas com a interface de utilizador. O Bootstrap foi desenvolvido, principalmente, por dois elementos da equipa do Twitter, um engenheiro e um designer. Construir uma interface gráfica Web de raiz que seja atraente e tenha uma boa usabilidade não é uma tarefa fácil para alguém que não tem formação específica em áreas de design. Assim, tomei a opção de usar este componente para acelerar o desenvolvimento dessa interface e alcançar melhores níveis de usabilidade. Suportar todos os *browsers* populares exige um grande esforço quando se desenvolve uma aplicação Web. O Bootstrap também ajuda nesta tarefa. Ele suporta todos os *browsers* atuais, alguns mais antigos mas ainda muito utilizados atualmente e a maioria dos *browsers* para dispositivos móveis.
- **jQuery UI** [63]: o jQuery UI é outro *plugin* para jQuery que, de certa forma, é semelhante ao Bootstrap. Contudo, não é direcionado para a construção de interfaces gráficas completas, como o Bootstrap, mas sim para criar pequenas animações, efeitos e *widgets* gráficas. Alguns *widgets* e efeitos gráficos que foram necessários são implementados apenas pelo jQuery UI.
- **Google Chart Tools** [64]: é uma API JavaScript fornecida pela Google para desenhar gráficos permitindo pequenas animações de interação. Esta API foi usada para criar os gráficos que são apresentados na aplicação Web desenvolvida.
- **Google Maps JavaScript API** [65]: é uma API que permite desenhar mapas a partir do JavaScript e interagir com eles. Esta API foi usada para apresentar ao utilizador as estatísticas sobre a origem dos pacientes da instituição de saúde, conforme veremos adiante.

#### 6.2.4. Comunicação com os sensores

A importância da aplicação Web desenvolvida vem da sua capacidade em interagir com os sensores apresentados anteriormente, criando uma interface única e integrada para que os utilizadores tirem partido da informação recolhida pelos sensores.

Existiam duas abordagens possíveis para a interação com os sensores. A primeira era comunicar com os *web services* dos sensores diretamente a partir do browser do utilizador. A segunda era utilizar o servidor Web como intermediário entre o browser do utilizador e os sensores. Cada uma das abordagens tem vantagens e desvantagens.

A primeira abordagem liberta o servidor Web de cargas desnecessárias, tornando-o assim mais disponível para atender mais utilizadores em simultâneo. As desvantagens desta abordagem são:

- A maioria dos browsers implementam uma política de segurança chamada "Same Origin Policy" [66], que dificulta o acesso a recursos de outras origens a partir da página carregada no browser do cliente. Contudo, existem várias formas de contornar isto.
- Como vimos anteriormente, os acessos aos *web services* dos sensores podem ser autenticados. Logo, usar esta abordagem implicaria passar as credenciais de acesso aos sensores que exigem autenticação para todos os clientes que usassem esses sensores. Isto provocaria uma falha de segurança, uma vez que, após obterem essas credenciais, os utilizadores conseguiriam aceder diretamente aos sensores sem usar o site desenvolvido, que autentica os utilizadores.
- Exige que as máquinas dos utilizadores sejam capazes de processar grandes quantidades de dados a através de JavaScript, para apresentação dos resultados e cálculos das métricas.
- Os clientes precisam de estar na mesma rede física dos sensores ou então precisam que a *firewall* da rede dos sensores conceda acesso aos sensores (Figura 6.9).

A segunda abordagem não padece das desvantagens apresentadas para a primeira. É mais segura pois controla todos os acessos aos sensores. Esta abordagem exige mais recursos da parte do servidor Web. Contudo, as máquinas dos utilizadores não precisam de ter grandes recursos computacionais para aceder a toda a informação obtida pelos sensores. Se o servidor Web tiver acesso aos sensores, o que em princípio deverá acontecer em todas as situações, os clientes só precisam de garantir que têm acesso ao servidor Web e não precisam de se preocupar em ter acesso a todos os sensores também (Figura 6.9).

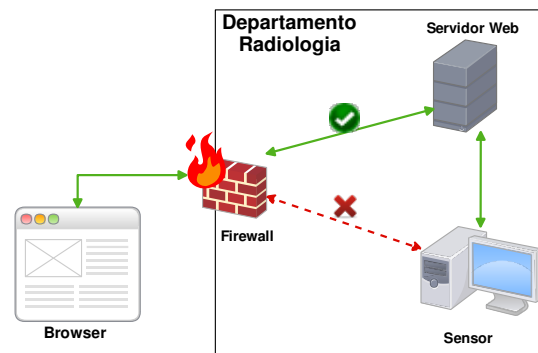


Figura 6.9: Formas de acesso aos sensores a partir do cliente Web.

A abordagem escolhida para acessar aos sensores foi a segunda, que estabelece ligação usando o servidor Web como intermediário. Para fazer esta intermediação foi necessário desenvolver um conjunto de classes PHP para comunicar com os sensores (Figura 6.10). Estas classes são utilizadas pelos controladores do CakePHP de forma a invocarem os métodos dos *web services* dos sensores e fazer os cálculos intermédios, libertando assim os clientes de realizar tarefas de processamento exigentes (em alguns casos) a partir do JavaScript.

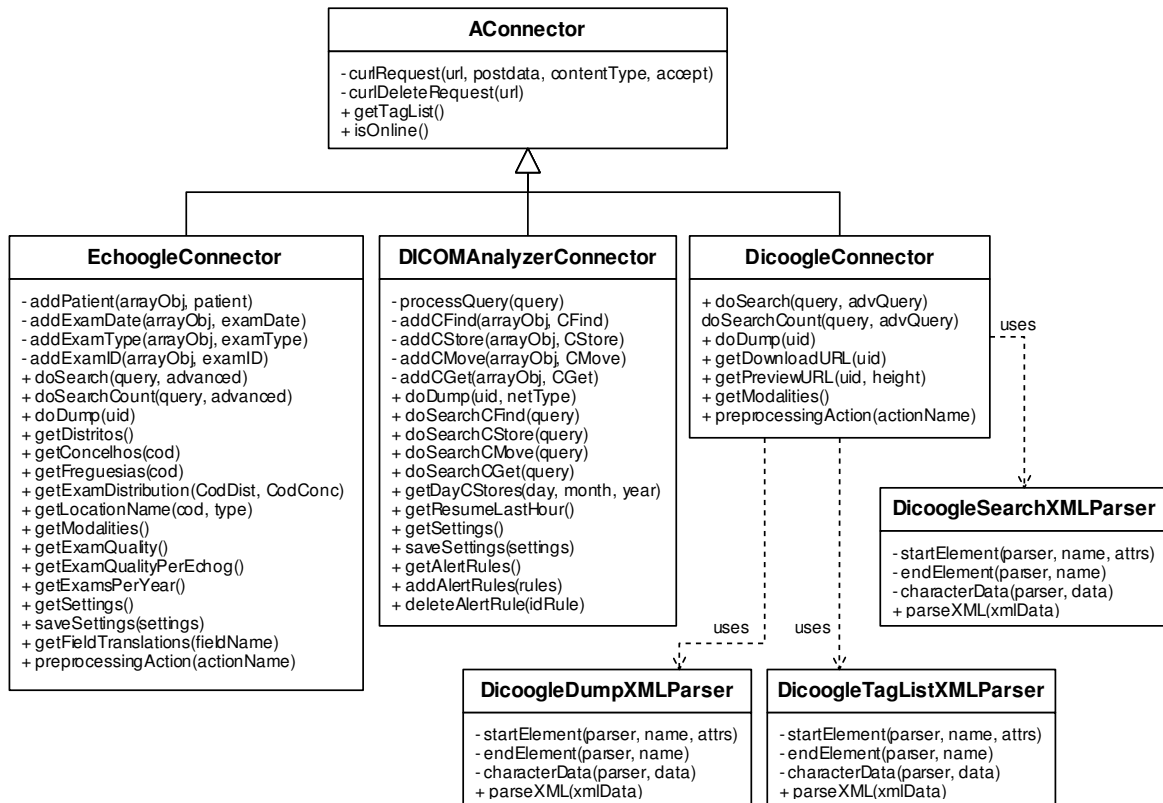
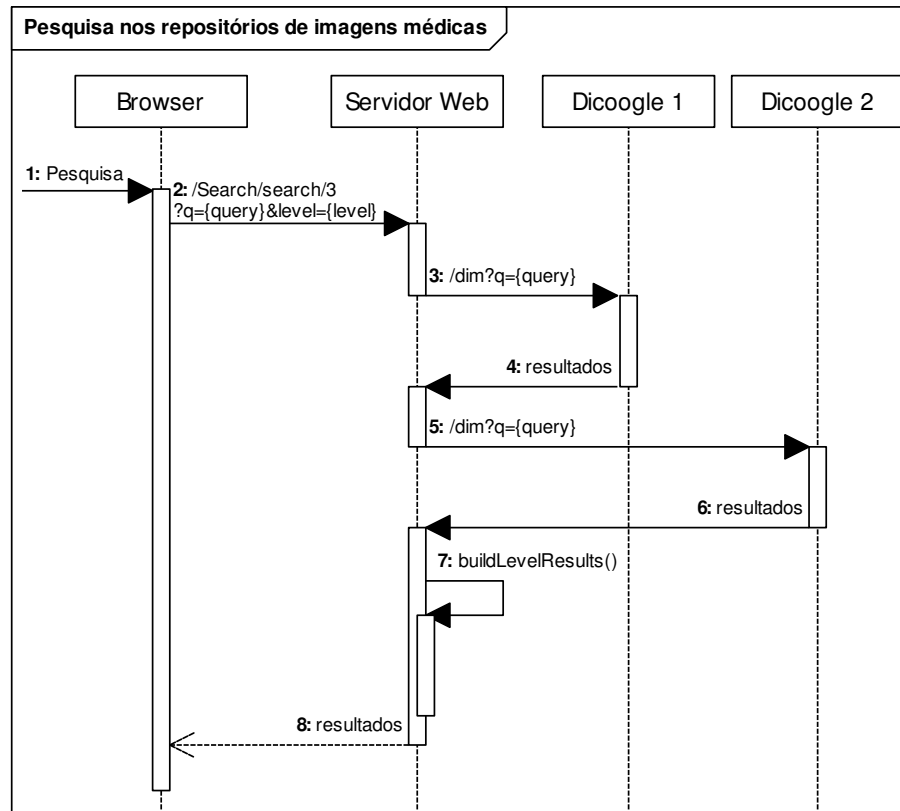


Figura 6.10: Classes de conexão aos sensores.

A comunicação com os sensores de rede e relatórios de ecocardiogramas é sempre feita usando JSON porque as suas estruturas têm tamanhos menores e são mais simples de processar, tanto em PHP como em JavaScript. A comunicação com o Dicoogle é feita usando XML entre o servidor Web e o sensor porque o Dicoogle não suporta JSON. No servidor Web, as estruturas XML são processadas e os resultados enviados para a *browser* são já codificados em JSON.

### Pesquisa

A Figura 6.11 apresenta o diagrama de sequência de uma pesquisa feita em todos os sensores de repositórios de imagens médicas a partir da aplicação Web desenvolvida. No caso da figura, existiam dois sensores de imagens médicas. A mensagem enviada do browser para o servidor Web é um HTTP POST cujo conteúdo é a *query*. A pesquisa nos diversos sensores do mesmo tipo é feita sequencialmente.



**Figura 6.11: Diagrama de sequência de uma pesquisa nos repositórios de imagens médicas com dois sensores de imagens médicas.**

A pesquisa feita sobre os sensores de rede e de relatórios de ecocardiogramas funciona de forma semelhante a esta. Também é possível pesquisar em vários tipos de sensores em simultâneo, conforme iremos ver no subcapítulo 6.3.3. Nesse caso, as pesquisas nos vários tipos de sensores em simultâneo irão decorrer em paralelo para cada um dos tipos. Dentro de cada tipo, a pesquisa decorre de forma simultânea, como vimos na Figura 6.11.

## 6.3. Interface de utilizador

Neste subcapítulo será apresentada a interface gráfica da aplicação Web desenvolvida. Para desenvolver a interface gráfica foram usados os dois componentes já apresentados, Twitter Bootstrap e jQuery UI. Além disso houve a preocupação de usar CSS para separar as estruturas HTML da definição dos estilos gráficos, facilitando futuras alterações ao estilo das páginas e evitando redundâncias na definição de estilos.

### 6.3.1. Área de trabalho

Ao conceber uma interface Web é fundamental ter em atenção a organização gráfica dos conteúdos. A área de trabalho foi então dividida em quatro blocos (Figura 6.12).

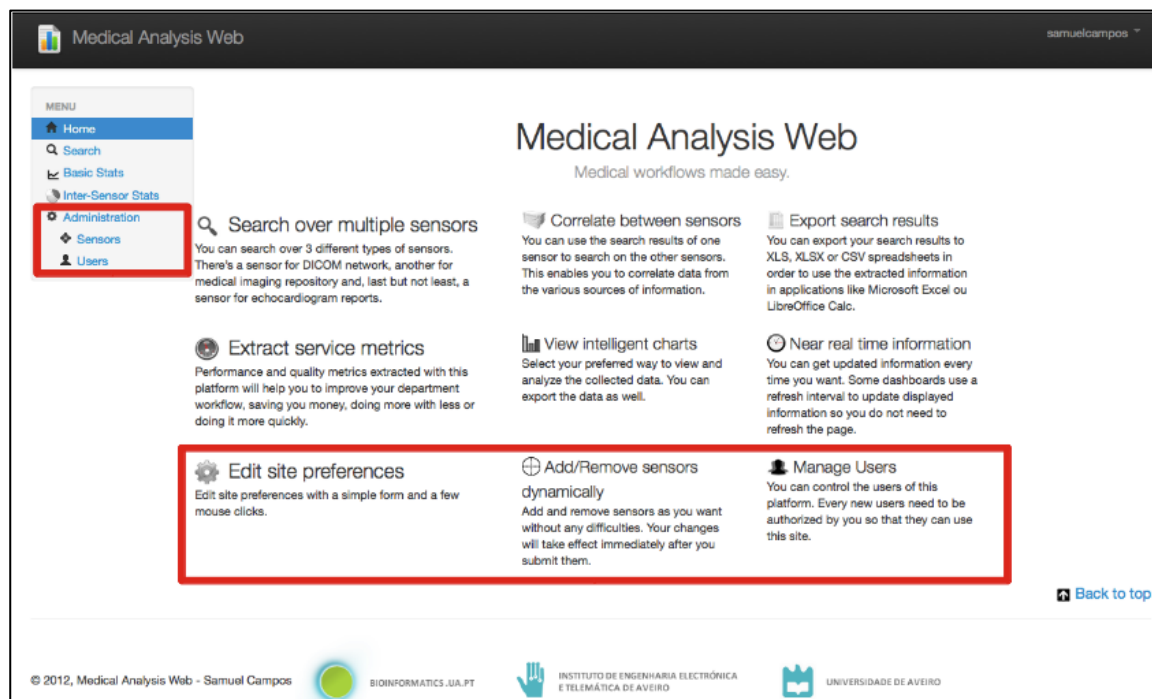


**Figura 6.12: Organização gráfica da aplicação Web desenvolvida.**

Os blocos do Cabeçalho, Menu e Rodapé são idênticos em todas as páginas do site. A única coisa que muda consoante a página é uma indicação no Menu de qual é a página atual. O bloco que contém a secção dinâmica varia consoante a página.

A Figura 6.13 apresenta a página inicial do site. Na figura podemos ver os vários blocos da área de trabalho e o aspeto geral do site. A secção dinâmica da página inicial contém uma lista com as principais funcionalidades da aplicação, acompanhada de uma pequena descrição de cada uma. Consoante o papel do utilizador no site, ele terá ou não acesso a várias zonas. Na Figura 6.13 podemos ver a vermelho as zonas que apenas os administradores conseguem ver.

Na maioria das restantes figuras de apresentação da interface gráfica, irá ser apenas apresentada a secção dinâmica de forma a poupar espaço, porque os restantes blocos são iguais aos da página inicial.



**Figura 6.13: Página inicial do Medical Analysis Web com indicação a vermelho das áreas que apenas são visíveis aos administradores.**

### 6.3.2. Conta de utilizador

Quando um utilizador acede a qualquer página do site (mesmo a inicial), caso ainda não tenha a sessão iniciada, é automaticamente redirecionado para uma página com o conteúdo da Figura 6.14. O utilizador só pode aceder a outras páginas após efetuar o *login*.

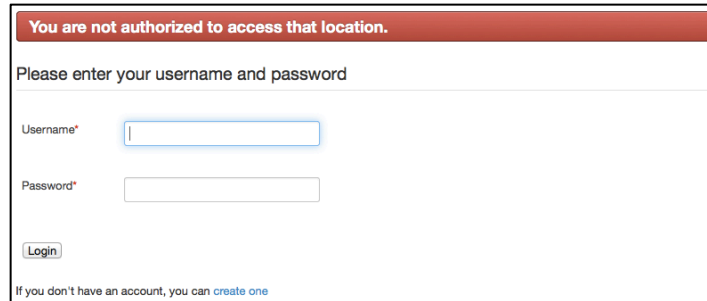


Figura 6.14: Página de início de sessão.

Caso ainda não tenha conta, pode criar uma (hiperligação no fundo da página da figura) e aguardar pela notificação de ativação da conta por email. A Figura 6.15 apresenta um diagrama com o processo de registo de novos utilizadores na aplicação Web desenvolvida.

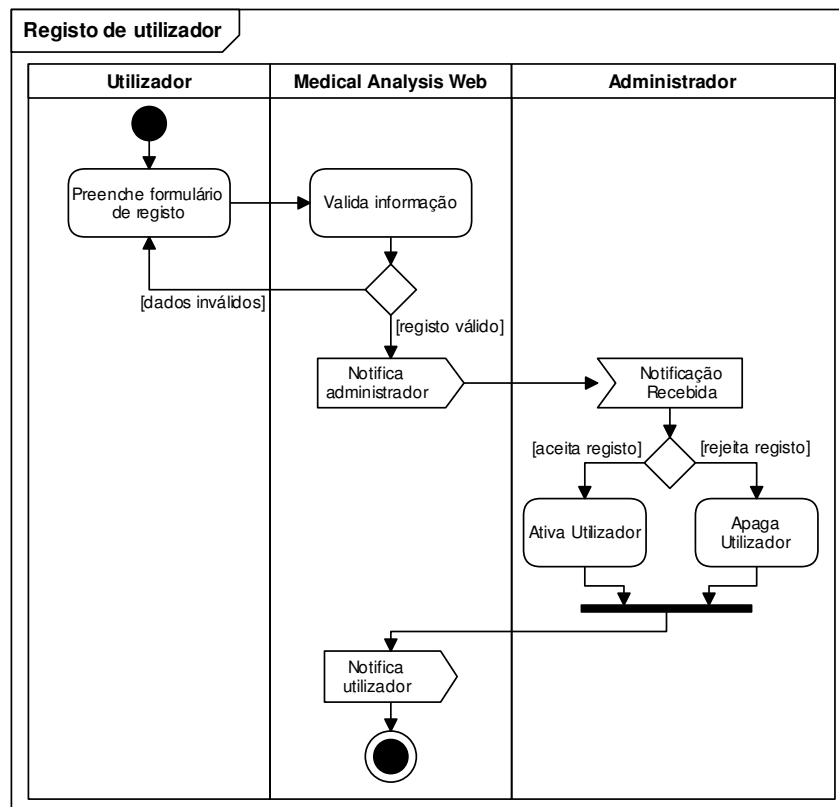


Figura 6.15: Diagrama de atividades do processo de registo de um utilizador.

A qualquer momento, o utilizador poderá atualizar o seu perfil, isto é, alterar o seu email ou a sua password (Figura 6.16). O nome de utilizador não pode ser alterado, uma vez criada a conta. Mais adiante iremos ver que os administradores também podem alterar alguns campos do perfil dos utilizadores. Conforme também podemos ver na Figura 6.16, o utilizador pode aceder ao seu perfil ou terminar a sessão clicando por cima do seu nome, no canto superior direito de qualquer página.



The screenshot shows the 'Medical Analysis Web' interface. At the top left is the logo and title. A navigation menu on the left includes 'Home', 'Search', 'Basic Stats', and 'Inter-Sensor Stats'. The main content area is titled 'Edit Profile' and contains four input fields: 'Username\*' with the value 'joao', 'Actual Password\*', 'Password\*', and 'Email\*' with the value 'joao@mail.com'. Below the fields is an 'Edit Profile' button. In the top right corner, a user profile dropdown menu is open, showing 'User: joao', a 'Profile' link with a person icon, and a 'Sign out' link with a power icon. A 'Back to top' link is located at the bottom right of the page.

Figura 6.16: Formulário de edição de perfil de um utilizador.

### 6.3.3. Pesquisa

A Figura 6.17 apresenta a página de pesquisa. No exemplo da figura foi feita uma pesquisa por relatórios de ecocardiogramas com a seguinte *query*: “PatientName:Paul”. Esta *query* retorna todos os relatórios cujo paciente tenha um dos seus nomes igual a “Paul”. No caso da figura, os relatórios encontravam-se pseudo anonimizados.

Conforme podemos ver no topo da Figura 6.17, a página de pesquisa tem três separadores. O primeiro para pesquisar nos sensores de relatórios, o segundo para pesquisar nos sensores do repositório de imagens médicas e o último para pesquisar nos sensores de rede. No topo da janela podemos ver um pequeno botão chamado “Show Query All Sensors”. Este botão faz aparecer uma nova caixa de texto e os respetivos botões para, de uma só vez, pesquisar em todos os tipos de sensores em simultâneo, ao invés de ter que pesquisar em todos os separadores

A *checkbox* chamada “Advanced” permite ao utilizado definir se quer enviar uma *query* avançada ou em texto livre. O anexo 10.3 apresenta exemplos dos dois tipos possíveis de *queries*.

Ainda no caso da Figura 6.17, foram retornados 66 resultados de dois sensores de relatórios de ecocardiogramas diferentes, identificados como “Echo” e “Echo 2”. Quando são retornados resultados de vários sensores para a mesma pesquisa, a aplicação usa cores diferentes na lista de resultados para que o utilizador possa reconhecer de onde são retornados. Na coluna do lado direito conseguimos ver uma legenda que indica o nome dos sensores que retornaram resultados com a respetiva cor de cada um.

Os resultados são apresentados em forma de árvore de forma a comprimir o espaço utilizado para apresentação dos resultados. A hierarquia usada para os relatórios de ecocardiogramas é a seguinte: Paciente, Data do Exame, Tipo do Exame e Identificação do Relatório. Para os resultados vindos do sensor de rede, a hierarquia usada é: Comando DICOM, Data da captura, Hora do dia da captura e identificação da captura. Finalmente, para os resultados do sensor de imagens médicas digitais, a hierarquia usada é: Paciente, Data do Exame, Identificação do Estudo e Identificação da Imagem médica.

The screenshot shows the Dicoogle search interface. At the top, there are tabs for 'Echocardiograms', 'PACS Archive', and 'Network Sensor'. Below the tabs, there is a search bar with the text 'PatientName:Paul' and a checked 'Advanced' option. A 'Search' button and a 'Search Tips' link are also visible. The search results are displayed as a tree structure under the heading 'Echocardiogram Repository Results: 66'. The tree shows a hierarchy of folders and files, with one file selected: '80458: b6389126-7429-4fd1-a0be-07377e92e653'. To the right of the tree, there is a 'SENSORS' section with 'Echo' and 'Echo 2' listed, and an 'ACTIONS' section with an 'Export Dump' button. A dropdown menu is open under 'Export Dump', showing options for 'XLS', 'XLSX', and 'CSV'. A 'Back to top' link is located at the bottom right of the interface.

**Figura 6.17:** Página de pesquisa com resultados anonimizados de uma pesquisa por relatórios de ecocardiogramas.

A árvore não é totalmente carregada quando os resultados são retornados porque isso sobrecarregaria o *browser* do cliente e demoraria mais a transferir quando o número de resultados de uma pesquisa é elevado. Ao invés disso, a árvore é carregada dinamicamente nível a nível, isto é, quando os resultados são retornados, apenas é transferido o primeiro nível da árvore do servidor Web para o browser. Ao expandir um nó (clitando no ícone da pasta ou através de duplo clique no texto à frente do ícone) é feita uma nova pesquisa, agora refinada, que irá apenas devolver o próximo nível de dados relativo a esse nó. Assim, o preenchimento da árvore é feita quase em tempo real, graças a pesquisas refinadas com os novos parâmetros, sem causar incômodos ao utilizador e poupando recursos desnecessários dado que raramente o utilizador irá expandir toda a árvore, pelo menos de uma vez só.

Por baixo da lista dos sensores que retornaram resultados, existe uma zona onde são apresentados diversos botões que permitem ao utilizador executar ações relativas à sua seleção na árvore de resultados. No exemplo da Figura 6.17, o utilizador clicou no botão “Export Dump” que permite exportar todos os campos indexados do resultado selecionado para uma folha de cálculo. O utilizador pode escolher o formato desejado, conforme está apresentado na figura.

Ao fazer duplo clique sobre um item do último nível da árvore, é apresentada uma nova janela, conforme a Figura 6.18. Neste caso, a janela apresenta todos os campos indexados de um relatório. Como os relatórios têm muitos campos e valores, poderia tornar-se difícil procurar aquele que se deseja. Para isso, na parte superior da janela, o utilizador pode definir o tamanho da paginação da tabela que apresenta os campos e pode também pesquisar/filtrar dados da tabela. Ao escrever na caixa de pesquisa, apenas serão apresentadas as linhas da tabela nas quais se encontra o valor escrito na caixa.

Tag	Value
AortGradMin	0.0
AortInsuf	3.0
AortValvType	1.0
CC	0.0
CodConc	17
CodDist	13
CodFreg	16
Content Type	ETT
CoroSun	1.58

**Figura 6.18:** Janela de listagem de todos os campos indexados para um relatório (Dump).

A pesquisa nos sensores de imagens médicas e de rede funciona de forma semelhante à que foi apresentada aqui, com as seguintes diferenças:

- Na pesquisa por imagens médicas, é possível pré-visualizar a imagem numa pequena janela e fazer download do ficheiro DICOM correspondente a um item do último nível da árvore. Contudo, o sensor de imagens médicas Dicoogle pode bloquear o acesso às pré-visualizações e ficheiros de imagem, por questões de segurança.
- A zona de pesquisa no sensor de rede é um pouco diferente daquela que foi apresentada na Figura 6.17. Não existe a *checkbox* de pesquisas avançadas mas existem 4 novas *checkboxes* que permitem definir o âmbito da pesquisa feita ao sensor de rede, conforme podemos ver na Figura 6.19.

**Figura 6.19:** Zona de pesquisa no sensor de rede.

#### 6.3.4. Métricas e estatísticas

Além de permitir pesquisar sobre todas as fontes de informação, um dos principais objetivos da aplicação Web desenvolvida é extrair indicadores de desempenho e qualidade que permitam avaliar os serviços prestados e, eventualmente ajudar a melhorar esses serviços.

A aplicação Web desenvolvida permite obter algumas métricas, na maioria sugeridas pelos profissionais de saúde contactados, através da análise dos dados recolhidos pelos sensores utilizados. Algumas utilizam inclusive dados obtidos por diferentes fontes. Essas métricas são apresentadas em forma de gráficos (Figura 6.20), utilizando principalmente a API “Google Chart Tools”, já apresentada anteriormente.

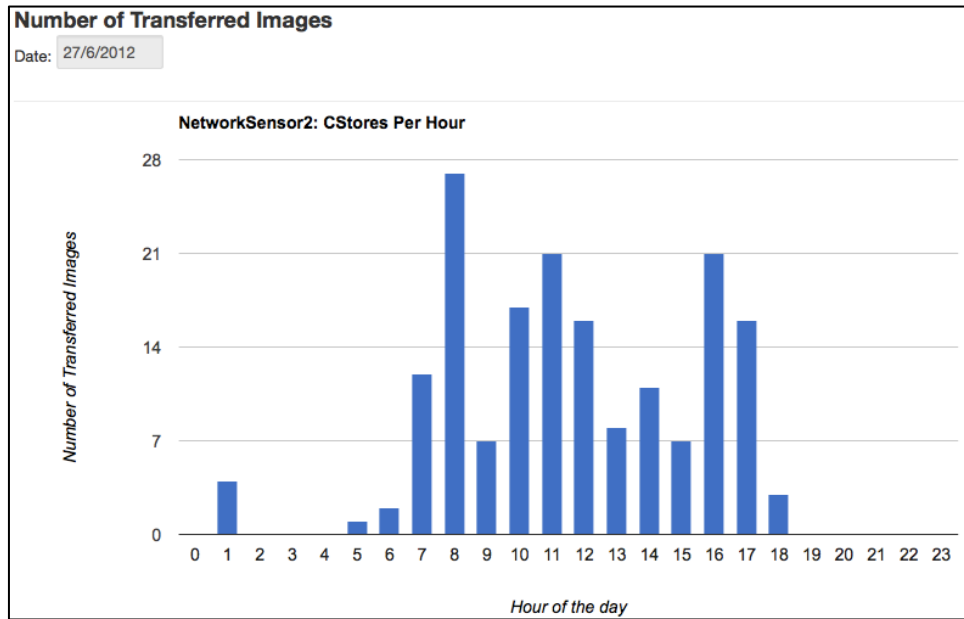


Figura 6.20: Gráfico gerado pela aplicação com o número de imagens médicas digitais transferidas ao longo do dia 27 de Junho de 2012, registadas pelo sensor “NetworkSensor2”.

Além da apresentação de gráficos, o utilizador pode descarregar várias folhas de cálculo, em formato XLS ou XLSX para tratar os dados no Microsoft Excel ou qualquer outro programa semelhante. As restantes métricas obtidas pelo sistema serão apresentadas no capítulo 7, quando forem apresentados os resultados do trabalho desenvolvido.

### 6.3.5. Administração

A aplicação Web desenvolvida tem uma secção que apenas está acessível aos utilizadores com papel de administrador. A secção de administração permite aos administradores gerir as preferências da aplicação, os sensores utilizados e os utilizadores do sistema. Na Figura 6.13 podemos ver quais os menus que estão disponíveis apenas aos administradores da aplicação (retângulo vermelho do lado esquerdo).

O menu “Administration” permite aos administradores aceder a uma zona onde podem ver e alterar as preferências da aplicação.


#### Sensores

Os administradores podem gerir os sensores que são utilizados pela aplicação desenvolvida, através do menu “Sensors”. A Figura 6.21 apresenta o exemplo de uma lista de sensores utilizados pelo Medical Analysis Web. Através da figura podemos ver que a aplicação tinha, no momento, quatro sensores configurados.

Sensors				
Name	URL	Sensor Type	State	Actions
Dicoogle	<a href="http://192.168.30.206:6060">http://192.168.30.206:6060</a>	PACS Archive	Offline	
Echo 2	<a href="https://localhost:9997/">https://localhost:9997/</a>	Echocardiogram	Online	
NetworkSensor	<a href="https://192.169.30.207:9999">https://192.169.30.207:9999</a>	DICOM Network	Offline	
NetworkSensor2	<a href="https://localhost:9999">https://localhost:9999</a>	DICOM Network	Online	

Figura 6.21: Lista de sensores utilizados pelo Medical Analysis Web.

Cada um dos sensores é identificado por um nome único, tem um URL de acesso aos seus *web services* e um tipo associado. No exemplo da Figura 6.21, existia um sensor de imagens médicas (*PACS Archive*) na máquina 192.168.30.206, um sensor de relatórios de ecocardiogramas (*Echocardiogram*) na mesma máquina do servidor Web e dois sensores de rede (*DICOM Network*), um na mesma máquina do servidor Web e outro na máquina 192.168.30.207. A lista de sensores que é apresentada em forma de tabela, tem ainda duas outras colunas que permitem identificar o estado do sensor (*Online* ou *Offline*) e executar ações associadas aos sensores. As ações podem ser de edição do registo, remoção o registo do sensor, alteração das definições, controlo do serviço de pré-processamento e gestão de alertas do sensor de rede.

No caso de sensores de imagens médicas digitais, não é possível alterar as suas definições a partir da aplicação Web desenvolvida, uma vez que esses sensores não disponibilizam tal funcionalidade através dos seus *web services*. Para os sensores de rede e de relatórios de ecocardiogramas, ao clicar no ícone  da coluna de ações, o administrador é redirecionado para uma página onde pode consultar e alterar as definições do sensor (Figura 6.22). O formulário de consulta e alteração das definições do sensor também é composto por vários separadores, com uma organização semelhante à da interface gráfica dos sensores.

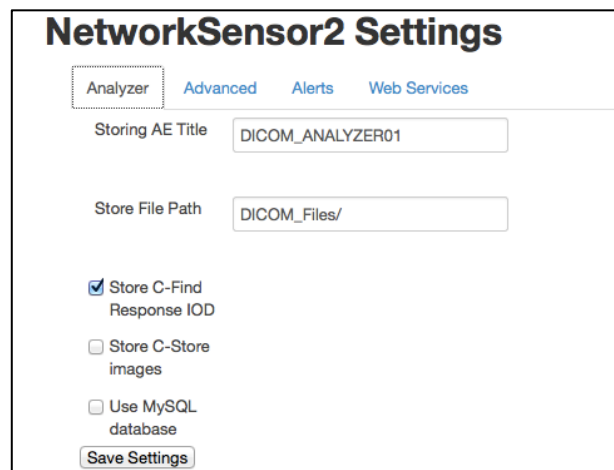



Figura 6.22: Parte do formulário de visualização e edição de definições do sensor de rede.

Para os sensores de relatórios de ecocardiogramas e de imagens médicas, o administrador pode controlar o seu serviço de pré-processamento. Isto é, pode executar todas as ações definidas em 3.5.1 e 5.4.4. Se o administrador clicar no ícone  da coluna de ações, será redirecionado para uma página onde pode interagir com o sensor de pré-processamento dos sensores. Na Figura 6.23 podemos ver a secção dinâmica da página de interação com o sensor. No caso da figura, o serviço de pré-processamento encontrava-se em execução. A barra de progresso, que é refrescada usando um mecanismo de *pooling* com um intervalo definido nas definições da aplicação, indica o estado atual do processamento, caso este esteja a decorrer. Os botões abaixo da barra de progresso permitem ao administrador interagir com o serviço.

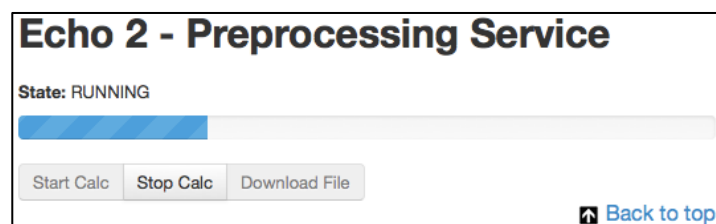

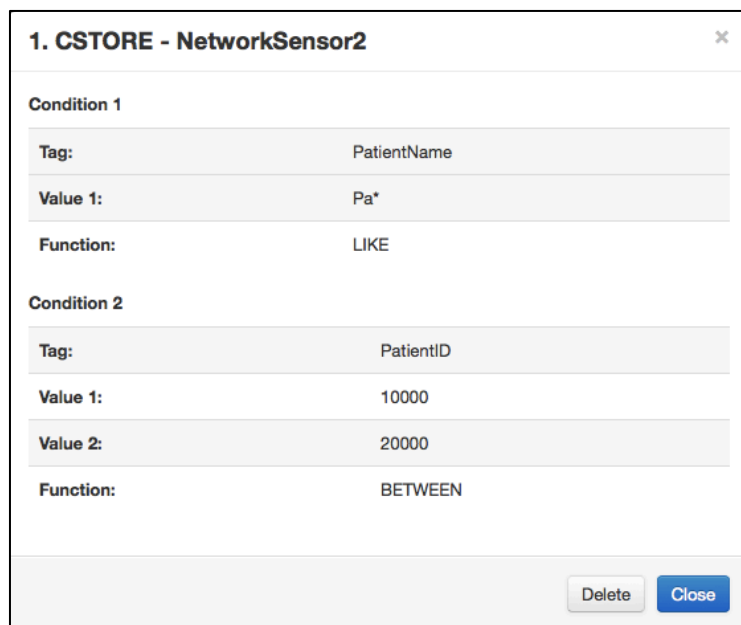


Figura 6.23: Página de interação com o serviço de pré-processamento durante a execução do mesmo.

Para os sensores de rede é possível controlar a sua lista de regras de alerta. Ao clicar no ícone , o administrador é redirecionado para uma página com uma lista de todas as regras de alerta configuradas nesse sensor. Ao clicar sobre um elemento dessa lista, são apresentados os detalhes da regra de alerta, conforme o exemplo da Figura 6.24. O administrador pode apagar ou adicionar regras de alerta ao sensor. O exemplo da Figura 6.24 é o mesmo que foi apresentado na Figura 4.7, no subcapítulo 4.5.

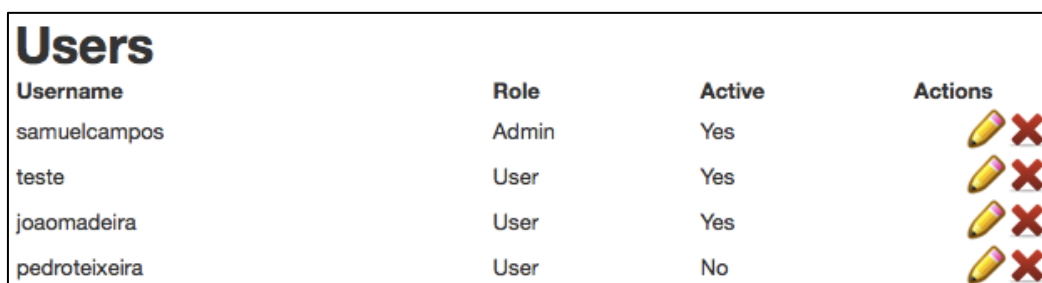


1. CSTORE - NetworkSensor2	
<b>Condition 1</b>	
Tag:	PatientName
Value 1:	Pa*
Function:	LIKE
<b>Condition 2</b>	
Tag:	PatientID
Value 1:	10000
Value 2:	20000
Function:	BETWEEN
<input type="button" value="Delete"/> <input type="button" value="Close"/>	

Figura 6.24: Condições de uma regra de alerta para o comando C-STORE configurada no sensor “NetworksSensor2”.

### Utilizadores

Ao aceder ao menu “Users”, o administrador tem acesso a uma lista com todos os utilizadores da plataforma (Figura 6.25). Essa lista apresenta também o papel do utilizador no site e indicação se esse utilizador se encontra ativo ou não. No exemplo da Figura 6.25, o utilizador “pedroteixeira” não se encontra ativo para usar a aplicação.











Users			
Username	Role	Active	Actions
samuelfcampos	Admin	Yes	 
teste	User	Yes	 
joaomadeira	User	Yes	 
pedroteixeira	User	No	 

Figura 6.25: Lista de utilizadores do Medical Analysis Web.

Utilizando os botões da coluna da direita, o administrador pode editar ou apagar os utilizadores. Ao editar o perfil de um utilizador, o administrador tem acesso a um formulário onde pode ativar ou desativar o utilizador e alterar o seu papel no sistema, isto é, definir se ele também será administrador ou não (Figura 6.26).

**Edit Profile**

Username\* pedroteixeira

Email\* pedroteixeira@mail.com

Active

User Role\* User

Edit Profile

Figura 6.26: Formulário de edição de um perfil fornecido ao administrador.





## 7 Resultados e discussão

Este capítulo apresenta os resultados do trabalho desenvolvido, os testes feitos para validar e avaliar o mesmo e qual é o impacto deste trabalho na área em que se pretende que atue.

### 7.1. Sensor de rede

O sensor de rede permite registar e analisar o tráfego DICOM que circula na rede informática de um departamento de imagiologia. O mecanismo de alertar criado permite aos administradores definirem um conjunto de regras baseadas nos valores dos campos DICOM, conforme foi apresentado. Este mecanismo de alerta envia um email para um endereço predefinido sempre que uma das regras se verifica. Desta forma é possível, por exemplo, saber quase em tempo real quando um aparelho de modalidade usa um nível de radiação superior a um limiar estabelecido, desde que o aparelho o registe nas imagens médicas recolhidas.

#### 7.1.1. Desempenho e fiabilidade

Para avaliar o desempenho e a fiabilidade do sensor de rede, foi criado um cenário de teste usando um utilitário da biblioteca DCM4CHE2, chamado “dcmqr”, com um *script* contendo 3 C-FIND *requests* diferentes, dentro de um ciclo. Foi usada uma rede de 100 Mbps entre o cliente DICOM (dcmqr) e o servidor de armazenamento que fornecia o serviço de Query/Retrieve.

O sensor de rede estava a correr no Windows XP, dentro de uma máquina virtual, usando o VirtualBox [67]. O VirtualBox estava a correr numa máquina com um processador Inter Core 2 Duo a 2.2 GHz. A máquina virtual foi configurada para usar apenas um núcleo do processador e 768 MB de RAM para limitar o desempenho da máquina de teste.

No total, foram enviados 60.000 pedidos C-FIND para o servidor de armazenamento, com 20 associações DICOM em simultâneo. Estes pedidos geraram um total de 80.000 respostas C-FIND. Tendo em conta a norma DICOM, foram transmitidos cerca de 580.000 pacotes na rede, originados pelo teste, ignorando os pacotes TCP para estabelecer e gerir as sessões.

Todos os 60.000 pedidos C-FIND foram capturados e analisados com sucesso pelo sensor de rede. Das 80.000 respostas, em média, apenas uma não foi capturada pelo sensor. Isto corresponde a uma taxa de sucesso de 99.99875% ao analisar as respostas C-FIND. Os testes levaram, em média, 13 horas e 48 minutos para concluir. O sensor de rede precisou de 5 horas e 58 minutos de tempo de processador, para analisar todos os pacotes transmitidos na rede e guardar a informação extraída dos pacotes numa base de dados MySQL.

### 7.2. Sensor de relatórios de ecocardiogramas

O sensor de relatórios de ecocardiogramas permite recolher informação vinda dos relatórios, indexando-os com o Lucene. A Tabela 7.1 apresenta o tempo total de recolha e indexação de 84063 relatórios em 3 casos distintos, usando sempre o mesmo *indexing step*. Este teste foi executado numa máquina virtual com um processador com 1 Core a 2.67 GHz e 8 GB de RAM e com a máquina virtual do Java configurada para utilizar no máximo 1GB de

memória. A localização física base de dados com os relatórios originais era uma unidade lógica cifrada com o utilitário TrueCrypt [68] usando o algoritmo AES. Usar unidades cifradas aumenta consideravelmente o tempo de acesso ao disco.

No primeiro caso de teste não foi feita qualquer tipo de anonimização. No segundo caso foi feita anonimização local e, por último, anonimização remota com autenticação noutro sensor a correr na mesma máquina do sensor de relatórios que estava a indexar.

**Tabela 7.1: Tempos de indexação do sensor de relatórios de ecocardiogramas com um *indexing step* de 50.**

<i>Caso</i>	<i>Tempo total de indexação</i>
Sem anonimização	4h 21m 04s
Anonimização Local	4h 21m 46s
Anonimização Remota	4h 44m 45s

Conforme podemos verificar pela tabela, a diferença entre os tempos de indexação sem anonimização e anonimização local têm apenas 42 segundos de diferença. Era esperado um acréscimo do tempo uma vez que é necessário realizar mais processamento para fazer a anonimização dos dados. Como este acréscimo é pequeno podemos dizer que a maior parte do tempo é gasto no acesso aos relatórios e escrita do índice em disco. O processo de anonimização é todo feito em memória, com estruturas do tipo *HashMap* que são muito eficientes. A diferença entre a anonimização local ou remota é mais significativa, uma vez que é preciso fazer um *request* a um *web service* e, neste caso, autenticar o sensor cliente por cada campo anonimizado. Em ambiente real é preciso escolher acertadamente qual dos sensores é que será o *master* de anonimização para que o impacto no desempenho do sistema não seja muito penalizador. Uma melhoria que pode vir a ser feita é usar um único *request* para enviar dados relativos a vários campos que necessitem de ser anonimizados.

Dado que a biblioteca de anonimização é comum aos 3 sensores utilizados neste trabalho, os resultados nos restantes sensores terão um comportamento semelhante aos do sensor de ecocardiogramas.

Alterar o parâmetro “*indexing step*” nas configurações do sensor de ecocardiogramas, altera o tempo total de indexação porque influencia diretamente o número de escritas do índice em disco. Através da Tabela 7.2 podemos ver que quanto maior for o *indexing step* menor é o tempo de indexação, conforme seria de esperar. A escolha do *indexing step* deve ter em conta a quantidade de memória RAM disponível na máquina que executa o sensor e também na máquina virtual do Java. Não é possível aumentar o *indexing step* indefinidamente pois isso irá originar problema de memória.

**Tabela 7.2: Tempos de indexação sem anonimização do sensor de relatórios de ecocardiogramas variando o *indexing step*.**

<i>Indexing step</i>	<i>Tempo total de indexação</i>
50	4h 21m 04s
100	3h 10m 05s
150	2h 54m 29s

### 7.3. Medical Analysis Web

O Medical Analysis Web é a aplicação desenvolvida para agregar a informação obtida pelos vários sensores apresentados neste documento. Esta aplicação permite aos utilizadores pesquisarem informação, relacionando os resultados provenientes das várias fontes. Os utilizadores podem construir *queries* em texto livre ou utilizando conjuntamente palavras chave e condições, para pesquisar sobre as várias fontes e assim estudar a informação recolhida. Vários indicadores de desempenho e qualidade dos serviços são automaticamente apresentados nas páginas da aplicação.

Os administradores podem gerir o conjunto de sensores de informação médica de forma centralizada numa única aplicação que está acessível a partir de qualquer computador dentro da rede da instituição e têm controlo sobre os utilizadores que tem acesso à informação obtida por todo o sistema.

Os indicadores e métricas fornecidos pelo Medical Analysis Web foram, na maioria, sugeridos por profissionais de saúde. As conclusões tiradas dos resultados apresentados de seguida não são tiradas usando conhecimento médico e profissional da área. Os mesmos resultados poderão permitir chegar a conclusões diferentes daquelas que serão apresentadas aqui, se forem analisadas por profissionais de saúde.

#### 7.3.1. Modalidades de imagens médicas

A Figura 7.1 apresenta um gráfico com as modalidades de imagens médicas digitais indexadas pelo sensor de imagens médicas Dicoogle. No momento existiam 4 modalidades de imagens médicas indexadas: CT, DR, DS e MR. A modalidade MR representa 42,9% do total de imagens médicas indexadas no sensor. É apresentado um gráfico deste género por cada sensor de imagens médicas que esteja a ser utilizado pela aplicação.

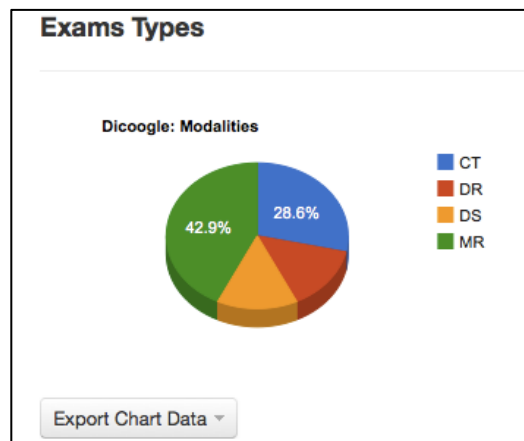


Figura 7.1: Gráfico que apresenta as modalidades de imagens médicas digitais indexadas e a sua respetiva percentagem.

No fundo da Figura 7.1 podemos ver um botão *dropdown* que permite aos utilizadores obterem folhas de cálculo com os dados utilizados para gerar este gráfico. As folhas de cálculo podem ser obtidas em formato XLS ou XLSX. Todos os gráficos gerados pela aplicação têm um botão semelhante para que os utilizadores possam manipular os dados, incluindo a eventual geração de novos gráficos. Por forma a reduzir a dimensão das figuras apresentadas de seguida, estas não irão apresentar este botão.

### 7.3.2. Tipos de relatórios de ecocardiografias

A aplicação também gera um gráfico por cada sensor de relatórios de ecocardiogramas semelhante ao da Figura 7.1 mas, ao invés de apresentar modalidades, apresenta a percentagem dos 3 tipos relatórios de ecocardiogramas.

Dos 84000 relatórios indexados, 93,4% são de ecocardiogramas transtorácicos (ETT), 5,4% são de ecocardiogramas transesofágicos (ETE) e apenas 1,2% são de ecocardiogramas de sobrecarga farmacológica.

### 7.3.3. Evolução do número de ecocardiogramas por ano

Um dos pedidos vindo dos profissionais de saúde era ver a evolução do número de ecocardiogramas realizados ao longo dos anos e saber qual o peso das várias faixas etárias para o total de exames realizados. A percepção que havia era de que o grupo de pacientes da ecocardiografia ia tendo uma média de idades superior. A Figura 7.2 apresenta a evolução do número de ecocardiografias entre os anos 1999 e 2012. As barras do gráfico encontram-se divididas pelas faixas etárias dos pacientes. A quebra do número de exames no ano corrente de 2012 é explicada pelo facto de os dados recolhidos dizem respeito apenas ao início do ano.

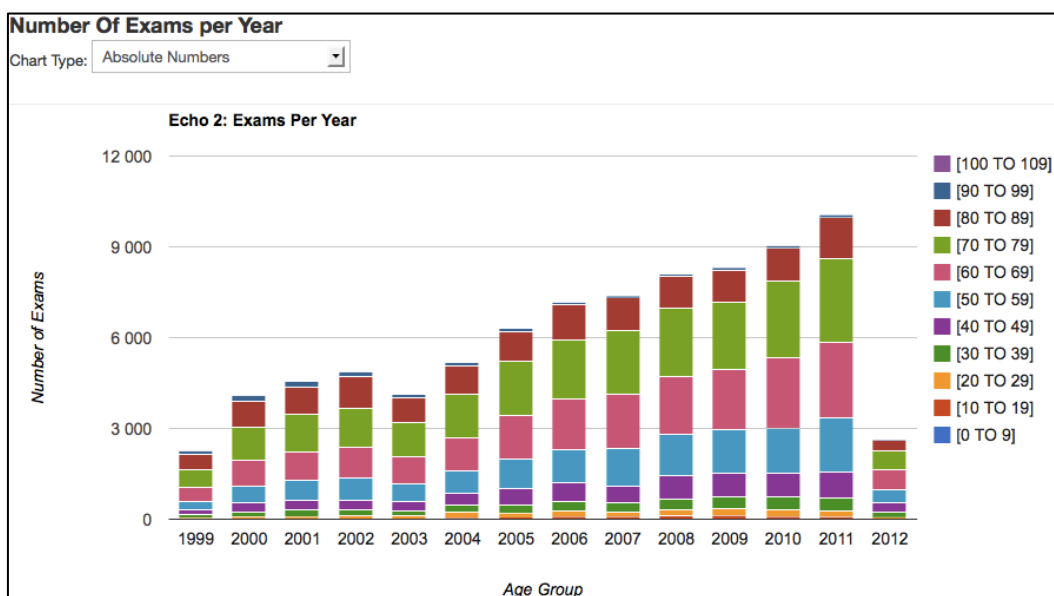


Figura 7.2: Evolução do número de exames entre os anos 1999 e 2012 salientando o peso de cada faixa etária.

Através do gráfico anterior podemos ver que a tendência é de realizar mais ecocardiografias a cada ano. O ano de 2003 representa uma exceção a esta tendência mas a explicação é dada por uma redução do número de profissionais. O gráfico anterior já nos permite ter uma ideia do impacto que cada faixa etária tem no valor final. Contudo, é interessante analisar estes dados não apenas usando números absolutos mas também em percentagens. No topo esquerdo de vários gráficos encontra-se uma *dropdown* que permite ao utilizador escolher visualizar os dados em números absolutos ou percentagens. O gráfico da Figura 7.3 apresenta os mesmos dados do gráfico anterior mas agora em percentagens.

Através destes dois gráficos podemos verificar que mais de 75% dos ecocardiogramas são realizados a pessoas com 50 anos ou mais. Podemos ainda ver que a contribuição para o número final dos pacientes com 80 anos ou mais, foi reduzindo ao longos anos. A contribuição dada pelos pacientes entre os 70 e os 79 anos manteve-se relativamente estável, sendo sempre a faixa etária que mais contribui para o número final. Verificou-se ainda um

aumento do número de pacientes entre os 30 e os 69 anos. Isto contraria em parte a preceção que havia junto dos profissionais de saúde. A média de idades tem baixado pouco a pouco.

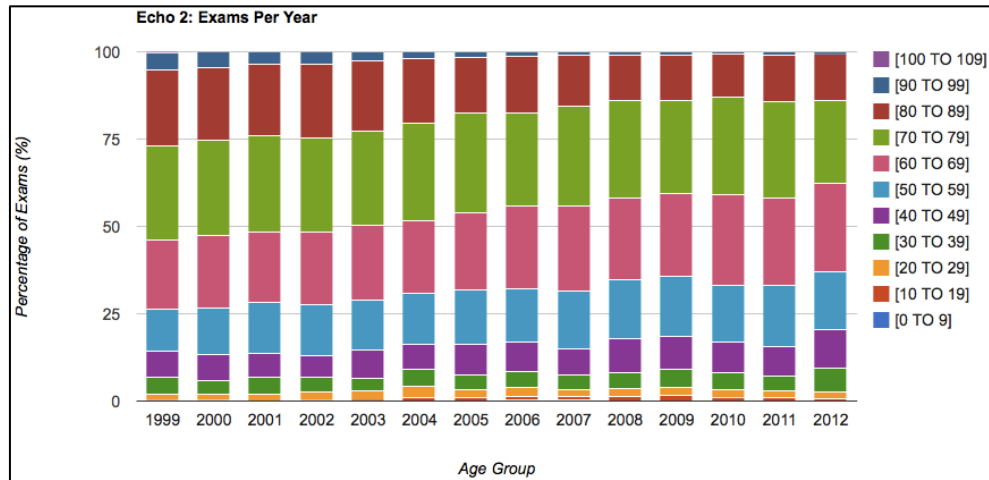


Figura 7.3: Evolução do peso de cada faixa etária entre os anos 1999 e 2012 para o número final de ecocardiogramas realizados.

#### 7.3.4. Qualidade dos exames

Um dos campos dos relatórios de ecocardiogramas diz respeito à qualidade das ecocardiografias que é percebida pelos profissionais que elaboram o relatório. É importante referir que esta apreciação de qualidade é subjetiva pois não existe um método determinístico para avaliar a qualidade de um exame. No entanto, como é feita por profissionais da área, existe um bom grau de confiança na apreciação feita. Existem 5 tipos de qualidade que podem ser atribuídos aos exames: “Boa”, “Regular”, “Má”, “Má Apical” e “Má Paraesternal”.

O gráfico da Figura 7.4 apresenta, para cada faixa etária, a percentagem de cada um dos tipos de qualidade. Tendo em conta os números absolutos por faixa etária, foram retiradas as idades cuja representação era muito baixa (o gráfico da Figura 7.3 também permite ver quais são as faixas etária com representação significativa).

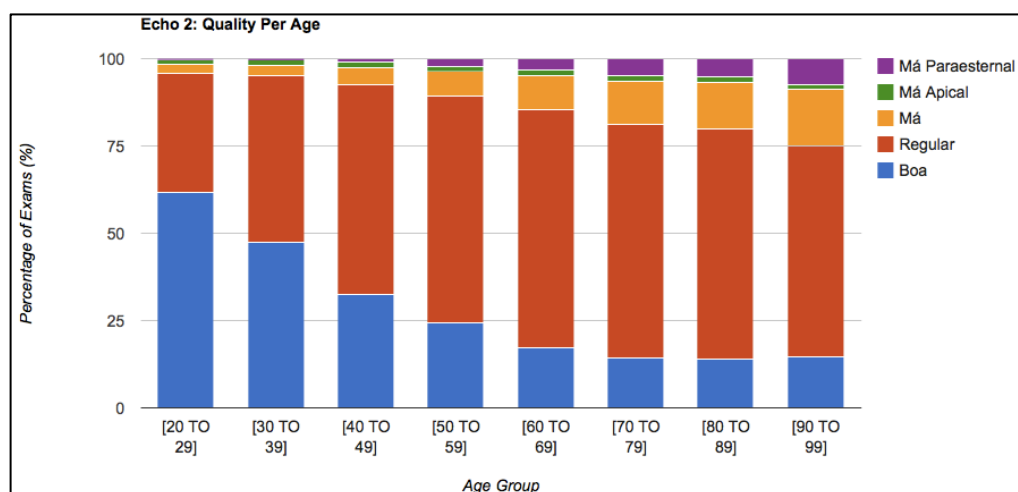


Figura 7.4: Qualidade dos exames por faixa etária com os tipos de qualidade representados em percentagem.

Através do gráfico da Figura 7.4 podemos concluir que a idade dos pacientes tem influência na qualidade do ecocardiograma. A probabilidade de um ecocardiograma ter uma qualidade “Boa” ou “Regular” é inferior para os pacientes com uma idade mais avançada. Para um paciente entre os 90 e os 99 anos, a probabilidade de ter um exame com má qualidade será cerca de 25%.

O gráfico da Figura 7.5 apresenta a percentagem de cada tipo de qualidade em função do peso dos pacientes. Novamente, foram excluídos os extremos porque o número de casos registados era muito baixo, comparativamente com os restantes casos. Através deste gráfico podemos concluir que o peso do paciente é outro dos fatores que tem influência na qualidade de um ecocardiograma. Também é possível que a mesma tendência seja verificada em qualquer exame baseado em técnicas de ultrassonografia. Pacientes com um índices de massa corporal elevados terão maior probabilidade de ter um exame com má qualidade.

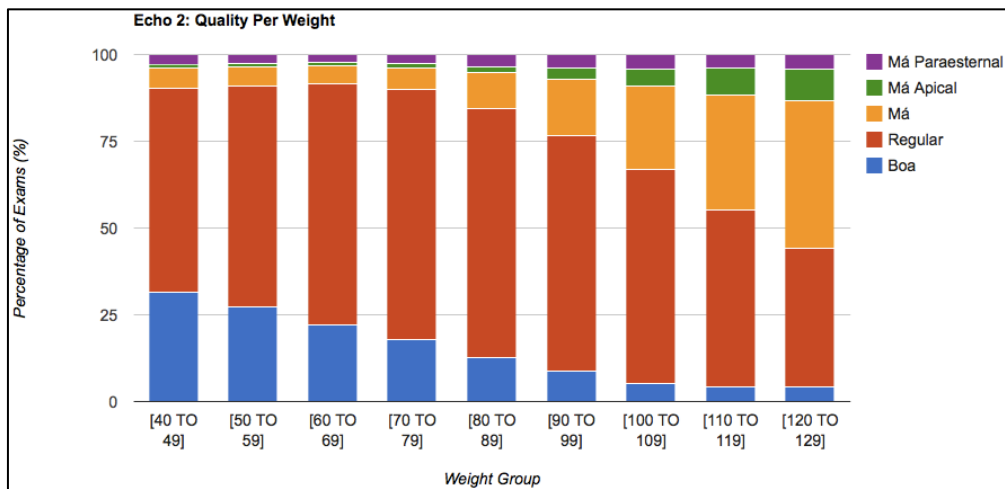


Figura 7.5: Qualidade dos exames por peso do paciente com os tipos de qualidade representados em percentagem.

A Figura 7.6 contém um gráfico que relaciona a qualidade dos exames com o ecógrafo que realizou o exame. O gráfico apresenta, mais uma vez, os valores em forma de percentagem mas foi feita uma filtragem para apresentar resultados apenas dos ecógrafos que realizaram mais de 1000 exames. Através da figura podemos ver que existem dois ecógrafos nos quais mais de 25% dos exames são percebidos como tendo má qualidade.

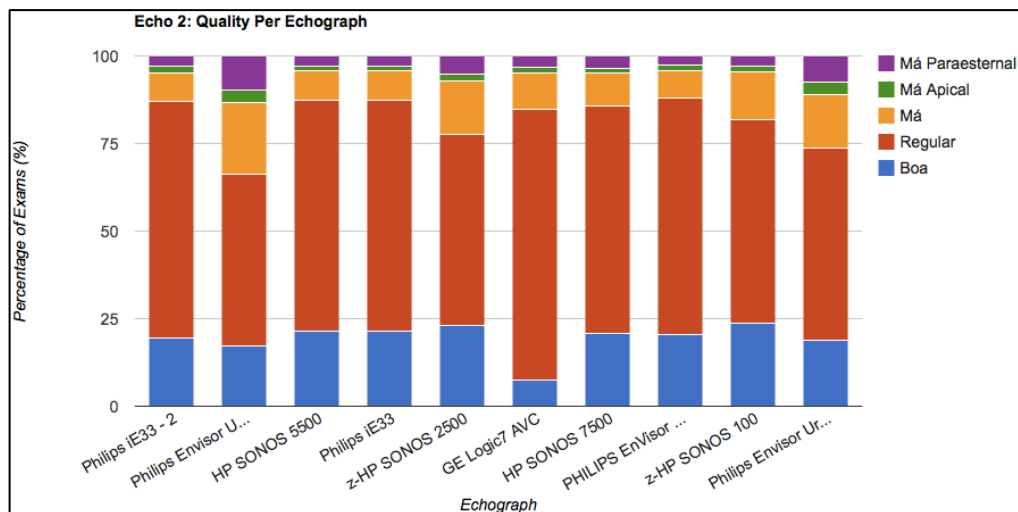


Figura 7.6: Qualidade do exame por ecógrafo com mais de 1000 exames realizados.

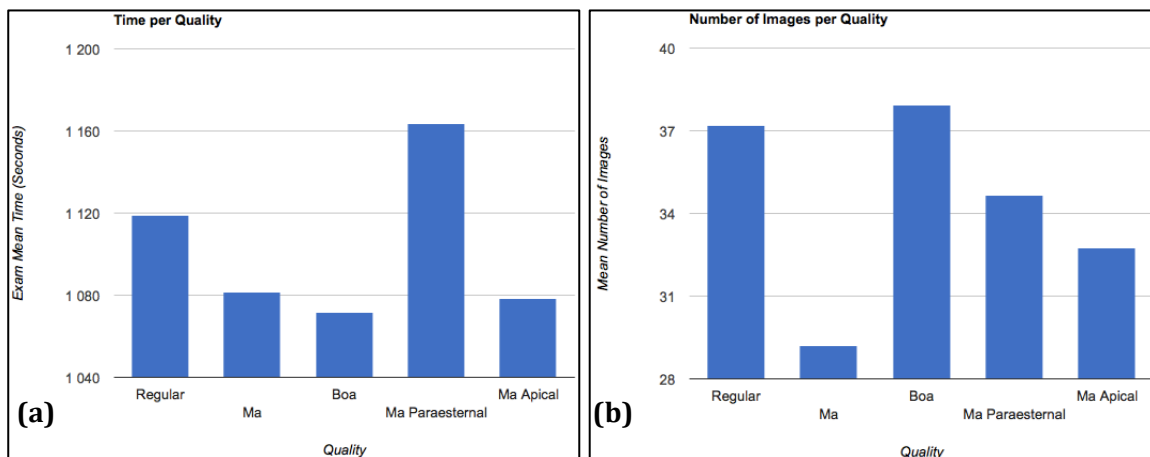
Isto pode ajudar os gestores do serviço no planeamento dos recursos no departamento durante a decisão de substituir ou não um determinado ecógrafo. Deve ter-se em conta que os dados do gráfico da Figura 7.6 dizem respeito a um período de mais de 13 anos e a noção de “qualidade” de um exame pode ter sofrido alterações ao longo do tempo. Contudo, usando o sistema desenvolvido, os utilizadores podem facilmente fazer deste tipo de estudos para cada um dos anos de atividade. É preciso analisar os vários fatores na tomada de decisões administrativas sobre os recursos.

### 7.3.5. Duração e número de imagens dos exames

As métricas que relacionam informação obtida pelo sensor de relatórios de ecocardiogramas com informação do sensor de imagens médicas são conseguidas usando os resultados do serviço de pré-processamento destes sensores.

Utilizando resultados dos dois sensores referidos, foi possível relacionar várias imagens médicas com o respetivo relatório de ecocardiogramas para 10463 exames.

A Figura 7.7 apresenta dois gráficos. O primeiro relaciona a qualidade percebida do exame com o tempo médio de duração do exame. O segundo gráfico apresenta o número médio de imagens médicas digitais adquiridas para cada tipo de qualidade percebida.



**Figura 7.7: a) Tempo médio de duração do exame em função da qualidade do mesmo. b) Número médio de imagens adquiridas também em função da qualidade.**

Através dos gráficos podemos ver que quando um exame tem boa qualidade, demora menos tempo a ser efetuado e é adquirido um maior número de imagens digitais. Quando a qualidade do exame é má (em qualquer das variantes), o número de imagens médicas adquiridas é menor.

O gráfico da Figura 7.8 agrega estas duas métricas para devolver o tempo médio de aquisição por imagem. Aqui podemos ver claramente quando os exames têm qualidade “Boa” ou “Regular”, o tempo de aquisição por imagem é um pouco menor relativamente aos exames com má qualidade.

Além de relacionar a qualidade do exame com o tempo e o número de imagens adquiridas, também foi feito trabalho no sentido de relacionar o ecógrafo utilizado no exame com o tempo e número de imagens adquiridas no mesmo. O Medical Analysis Web permite apresentar os mesmos três tipos de gráficos apresentados anteriormente, agora em função do ecógrafo utilizado.

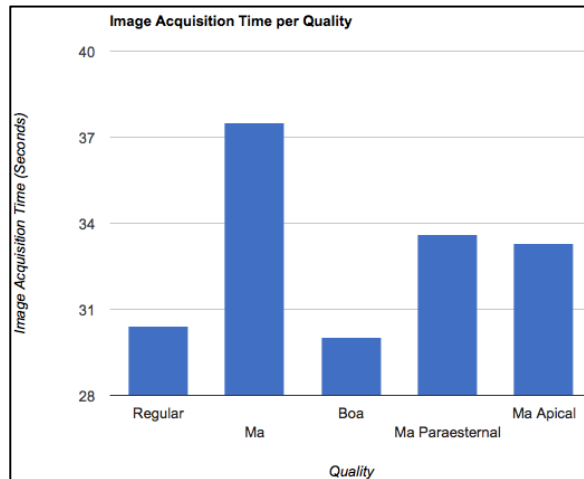


Figura 7.8: Tempo médio de aquisição por imagem para cada tipo de qualidade de exame.

A Figura 7.9 apresenta apenas o gráfico que relaciona o tempo médio de aquisição de uma imagem com o ecógrafo utilizado no exame. A coluna com o nome “Unknown” corresponde aos exames em que não foi registado o ecógrafo utilizado. No gráfico existem um ecógrafo que se salienta pela negativa. O ecógrafo “HP SONOS 7500” tem um tempo de aquisição por imagem significativamente maior que todos os restantes, demora sensivelmente o dobro dos dois ecógrafos que demoram menos tempo.

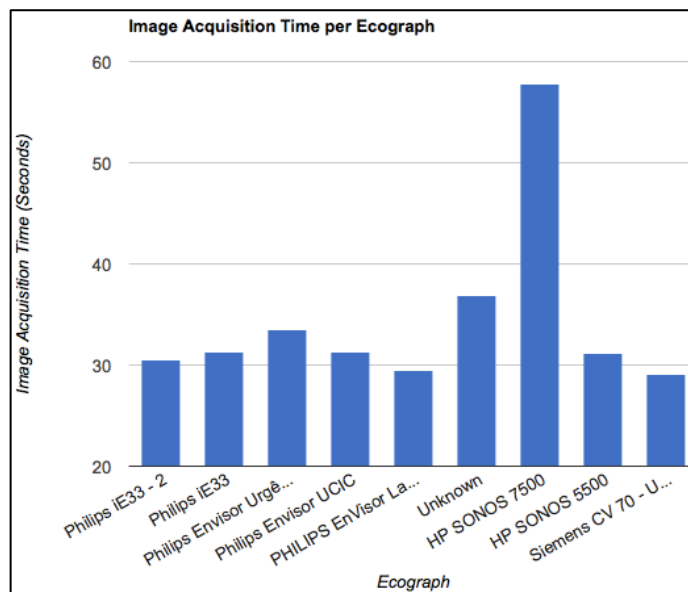


Figura 7.9: Tempo médio de aquisição de uma imagem médica em função do ecógrafo utilizado.

### 7.3.6. Origem dos pacientes

Numa perspetiva de negócio, a aplicação Web desenvolvida permite ter uma ideia da origem dos pacientes por distrito, concelho e freguesia nacional. A Figura 7.10 apresenta um mapa criado pela aplicação que representa graficamente um círculo cujo tamanho do raio é definido usando uma escala logarítmica que usa o número de exames naquela localização como parâmetro. Desta forma, o utilizador pode ter rapidamente uma noção da distribuição dos pacientes da instituição pelo território nacional.



O utilizador pode visualizar marcações no gráfico por distritos, concelhos ou freguesias. Para escolher pode usar um conjunto de *dropdowns* com as listas de distritos e concelhos nacionais, que está acima do mapa. Além disso, o utilizador pode ainda filtrar os resultados por ano para ter uma ideia da evolução. No futuro, outro tipo de filtragens podem vir a ser feitas. Ao clicar sobre um círculo, será apresentada o número de ecocardiogramas realizados a pacientes dessa localidade. A Figura 7.10 apresenta informação relativa aos concelhos do distrito de Aveiro com o número de exames do concelho de Águeda apresentado após um clique sobre o seu círculo representativo.

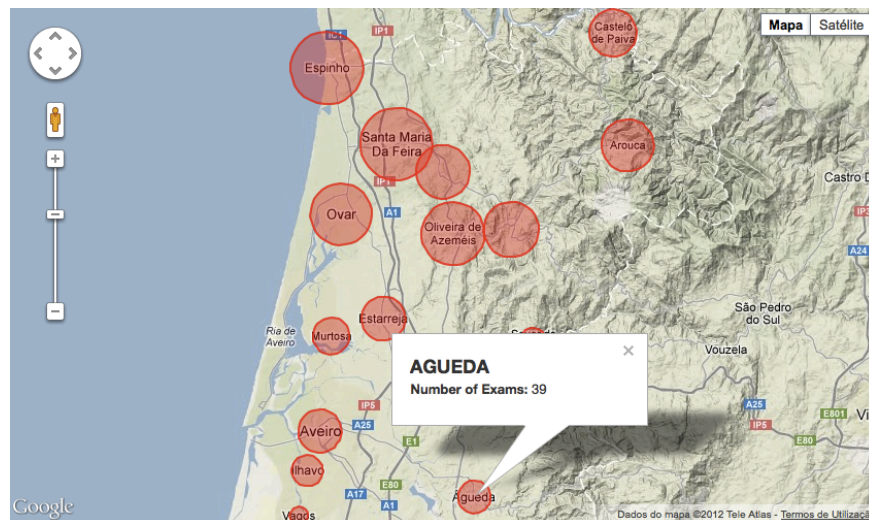


Figura 7.10: Origem dos pacientes da instituição.

### 7.3.7. Atividade na rede do PACS

Graças ao sensor de rede, o Medical Analysis Web também permite ter uma noção, quase em tempo real, da atividade na rede informática do PACS. A Figura 6.20 do subcapítulo 6.3.4 já apresentou um gráfico que a aplicação gera para que os utilizadores possam ver a atividade registada ao longo de qualquer dia em que o sistema foi utilizado.

A Figura 7.11 apresenta o conjunto de *dashboards* criados pela aplicação que indicam o número dos principais comandos DICOM executados durante a última hora. Enquanto a página respetiva estiver aberta, estes *dashboards* são atualizados com um mecanismo de *pooling* em intervalos de 10, 15, 30, 60 ou 120 segundos, conforme a preferências estabelecidas pelo administrador. A escala do vários *dashboards* é ajustada automaticamente se o número de comandos de cada tipo ultrapassar o limite.



Figura 7.11: Resumo da atividade da última hora, com os principais comandos DICOM, na rede do PACS.

No caso da Figura 7.11, na última hora tinham sido executadas apenas 3 pesquisas com o comando C-Find, 9 imagens tinham sido transferidas com comandos C-Find originados por 8 comandos C-Move. Nenhum comando C-Get tinha sido executado na última hora.

#### **7.4. Discussão**

Numa época em que a variedade de ambientes de execução é grande e os recursos utilizados na saúde têm que ser racionalizados, idealmente sem prejudicar a atividade, o sistema apresentado aqui foi desenvolvido inteiramente sobre tecnologias livres e independentes da plataforma. A instanciação de todos os componentes do sistema num ambiente real não estará condicionada pelos ambientes de execução utilizados na instituição.

Os resultados aqui apresentados são demonstrativos do valor do sistema desenvolvido. É possível extrair outros indicadores a partir da informação recolhida pelos vários sensores. Outros estudos estão a ser preparados atualmente em colaboração com profissionais da área para posterior publicação científica.

Fora o resultado apresentado em 7.3.1, não é apresentado nenhum outro resultado que utilize exclusivamente dados vindos do sensor de imagens médicas. O Dicoogle (o sensor de imagens médicas escolhido) já foi utilizado para extrair vários resultados, publicados na literatura [69].

Ao comparar os resultados obtidos com os requisitos definidos inicialmente, podemos concluir que os objetivos deste trabalho foram atingidos. A integração da informação vinda das três fontes previstas foi efetuada com sucesso. Os utilizadores podem analisar e extrair informação de múltiplas instâncias dos três tipos de sensores de informação, utilizando apenas uma aplicação.

Com o sistema desenvolvido, os utilizadores podem realizar outro tipo de estudos, além daqueles aqui apresentados, sem necessidade alterar o código fonte dos vários componentes, devido à sua flexibilidade, capacidade de pesquisa e exportação de dados. Foram também tomadas precauções de segurança adequadas com a importância da informação recolhida e produzida pelo sistema.

## 8 Conclusão e Trabalho Futuro

Neste capítulo serão apresentados aspectos importantes que foram surgindo ao longo de todo o processo, os principais problemas e contribuições do trabalho desenvolvido. Finalmente são sugeridos alguns caminhos para trabalho futuro, na sequência dos esforços aqui desenvolvidos.

### 8.1. Principais questões

Para desenvolver o sensor de rede baseado no protocolo de rede da norma DICOM, foi necessário passar por um longo período de estudo pois os detalhes são muitos e densos. A norma é muito extensa e a cada ano é lançada uma nova revisão. A biblioteca DCM4CHE2 ajudou especialmente na descodificação de objetos DICOM mas, ao nível da camada de rede do protocolo foi necessário desenvolver as classes para descodificar os PDUs, byte a byte. Trata-se de uma tarefa muito morosa e complexa.

O estudo dos relatórios de ecocardiogramas foi outro domínio do conhecimento novo com o qual não estava familiarizado. Apesar de a fonte de dados dos relatórios inicialmente parecer estruturada, a tarefa de adquirir os relatórios revelou-se muito mais complexa do que estava previsto. Existem várias aplicações que usam a base de dados que contém os relatórios de ecocardiogramas. Com o passar dos anos, foram feitas alterações a várias dessas aplicações, inclusivamente à própria aplicação que gere os relatórios. Estas alterações trouxeram necessidades de persistência de dados diferentes das originais e, por isso, a base de dados foi sendo estendida, sem muito planeamento. Hoje, a base de dados contém muitas redundâncias de informação, sem relações bem definidas. São poucas as chaves primárias e estrangeiras nas tabelas e, por isso, não é garantida a consistência dos dados, além do prejuízo para o desempenho do motor de base de dados.

Como foi falado neste documento, a codificação das várias estruturas XML mudou com uma reestruturação do sistema que teve lugar em 2009. Ao ver o resultado final, é possível constatar que esta reestruturação foi mal conduzida pois introduziu uma codificação pior daquela que existia (os campos deixaram de estar dentro de *tags* XML para passar a simples texto). Esta opinião não é apenas minha mas também dos profissionais de saúde, com quem pude contactar, pois perderam o acesso a várias funcionalidades que tinham anteriormente, como por exemplo: uma simples pesquisa por valores de um campo. Com este trabalho foi feito o esforço de uniformizar as duas codificações e assim voltar a recuperar essas funcionalidades pesquisando sobre o índice criado.

Um dos grandes desafios de todo o trabalho era integrar várias fontes de informação distintas numa única aplicação. Esta preocupação não foi tida em conta apenas durante a conceção da interface de utilizador na aplicação Web desenvolvida. Começou na definição das hierarquias e estruturas de dados, usadas para retornar os resultados das pesquisas nos *web services* dos sensores, tornando-se assim uma das principais preocupações durante o desenvolvimento de todos os componentes do sistema.

### 8.2. Principais contribuições

O trabalho desenvolvido no âmbito desta dissertação faz várias contribuições, tanto para os profissionais que poderão tirar partido dos seus resultados como na área da investigação científica.

Os vários sensores desenvolvidos no âmbito deste trabalho poderão ser reutilizados por outros sistemas, perante outros contextos, graças às suas interfaces com o exterior baseadas em *web services*.

O sensor de rede é uma aplicação que regista e analisa a informação transferida na rede informática, segundo a norma DICOM. Esta aplicação funcionará em qualquer instituição de saúde que use um sistema PACS tradicional. Com ela é possível analisar os fluxos de trabalho relacionados com imagiologia médica e assim ajudar os administradores de rede ou responsáveis pelo sistema na monitorização e resolução de problemas. Os seus registos podem também ser utilizados em mecanismos de *cache* de imagens médicas inteligentes, incluindo a utilização de *prefetching* e modelos preditivos.

O sensor de relatórios de ecocardiogramas foi desenvolvido para um caso de estudo específico e, portanto, não é genérico para funcionar em qualquer instituição sem serem necessárias alterações ao nível do seu código fonte. Contudo, a arquitetura desenvolvida permite que o esforço necessário para adicionar novos formatos de relatórios ou novas metodologias de acesso aos relatórios, não seja muito elevado. Este sensor trouxe novas possibilidades, no que diz respeito ao estudo da informação armazenada nos relatórios ao longo dos anos, que não eram possíveis até então.

A aplicação Web desenvolvida, Medical Analysis Web, utiliza os dois sensores desenvolvidos e ainda um sensor de informação vinda do repositório de imagens médicas digitais, o Dicoogle. O facto de ser uma aplicação Web traz muita flexibilidade a todos os utilizadores do sistema. Esta aplicação utiliza instâncias dos vários tipos de sensores e permite obter uma série de métricas e indicadores relacionados com o desempenho, qualidade e fluxo de trabalho da instituição, de forma automática. No entanto, as suas vantagens não ficam por aqui. Com ela os utilizadores podem facilmente fazer os seus próprios estudos, dada a possibilidade de pesquisa e exportação de dados.

Foi também publicado um artigo cujo foco é o sensor de rede desenvolvido no âmbito deste trabalho:

- Samuel Campos, Carlos Costa, Luís Bastião, “A Network Sensor for Medical Imaging Workflows” – 7ª Conferência Ibérica de Sistemas e Tecnologias de Informação (CISTI), Madrid, Espanha, Junho 2012

Outros artigos sobre o trabalho poderão vir ainda a ser escritos focando o restante trabalho desenvolvido, nomeadamente ao nível do sensor de relatórios e da aplicação Web que integra todas as fontes de informação. Oportunamente, os resultados deste trabalho serão também utilizados para fazer um artigo científico numa revista clínica da especialidade. Estes poderão servir para produzir algumas diretrizes para os profissionais de saúde e para quem é responsável pela tomada de decisões ao nível dos recursos existentes nos departamentos de imagiologia.

### **8.3. Trabalho Futuro**

Apesar das contribuições apontadas no subcapítulo anterior, muito trabalho pode vir a ser feito no futuro em seguimento do realizado até aqui. De seguida serão apresentadas algumas ideias para melhorar e estender a solução apresentada.

#### ***Melhorias de desempenho***

Foram identificados alguns pontos onde é possível melhorar o desempenho do sistema desenvolvido. Ao nível da pesquisa em vários sensores do mesmo tipo (Figura 6.11), é possível fazer essa operação em paralelo. O paralelismo pode ser implementado ao nível de

JavaScript no browser, enviando múltiplos HTTP *Requests*, ou em PHP no servidor Web. Este pode ser um fator importante caso existam várias instancias de cada um dos tipos de sensores e se queira acelerar o processo de pesquisa.

Atualmente o sensor de rede é o mais utilizado para ter uma noção daquilo que está a acontecer na rede da instituição. Neste momento é utilizado um mecanismo baseado em *pooling* para obter o seu estado. No futuro, poderão vir a ser extraídos mais indicadores do estado atual a partir deste ou de outros sensores. O mecanismo de *pooling* revelar-se-á então um pouco inadequado para este tipo de operações. Poderão ser implementados outros mecanismos para notificar a aplicação Web em tempo real, como por exemplo usar o protocolo XMPP (*Extensible Messaging and Presence Protocol*).

#### ***Integração de sensores do estado da rede “não DICOM”***

Para os administradores dos sistemas PACS, a utilidade do sensor de rede é grande. Contudo, existem outros parâmetros sobre o desempenho e conectividade da rede que podem ser importantes para a prevenção e manutenção dos sistemas a funcionar com o mínimo de quebras possível. Existem várias ferramentas que permitem monitorizar o estado das ligações de rede e das vários nós em tempo real. Seria interessante integrar uma dessas ferramentas no sistema desenvolvido de forma a ter, em tempo real, informação do estado da rede, podendo detetar-se rapidamente eventuais problemas na rede e atuar de imediato.

#### ***Analizador de Logs de aplicações médicas***

Os sistemas PACS que são utilizados atualmente têm vários componentes que não utilizam a norma DICOM. Hoje em dia, com a crescente popularidade dos ambientes Web, são várias as estações de trabalho nas instituições médicas que não utilizam aplicações DICOM tradicionais. Utilizam principalmente aplicações Web que correm nos browsers dessas estações de trabalho e que estão alojadas algures num servidor da instituição ou mesmo na *Cloud*. O tráfego gerado por essas aplicações é tráfego HTTP e muitas vezes vai cifrado e autenticado segundo o protocolo SSL/TLS. Assim sendo, é impossível analisar o tráfego transmitido e recebido por estas aplicações por métodos semelhantes àquele que foi apresentado. Felizmente a grande maioria dos componentes de sistemas PACS registam a sua atividade em *Logs* [23,70]. Poderá ser desenvolvido um novo tipo de sensor, com capacidades semelhantes às do sensor de rede, que possa analisar os *Logs* que registam a atividade das aplicações e extrair informação semelhante àquela que é extraída pelo sensor de rede.

#### ***Prospecção de dados e extração de conhecimento***

Os relatórios de ecocardiogramas contêm muita informação que foi indexada mas que ainda não foi alvo de nenhum estudo. Uma vez que possui vários campos de texto livre, principalmente o que contém a descrição do exame feita pelos profissionais de saúde, podem ser usadas técnicas de prospecção de dados (*data mining*) em conjunto com análise de vários outros campos para fazer extração de conhecimento. Os relatórios estão relacionados também com imagens médicas dos ecocardiogramas recolhidas durante o exame. Para estudos mais avançados, a informação dos relatórios pode servir para criar uma espécie de repositório de imagens anotadas e ajudar a fazer uma análise automática das patologias presentes em novas imagens.



## 9 Referências

- [1] I. N. de Estatística, “Proporção de hospitais que utilizam meios informáticos nas actividades desenvolvidas (%) por Tipo (actividade de gestão); Bienal,” 2009. [Online]. Disponível em: [http://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine\\_indicadores&indOcorrCod=0003216&contexto=bd&selTa](http://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine_indicadores&indOcorrCod=0003216&contexto=bd&selTa). [Data de acesso: 03-Maio-2012].
- [2] I. N. de Estatística, “Proporção de hospitais que utilizam meios informáticos nas actividades desenvolvidas (%) por Tipo (actividade de gestão - 2010); Bienal,” 2011. [Online]. Disponível em: [http://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine\\_indicadores&indOcorrCod=0005373&contexto=bd&selTa](http://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine_indicadores&indOcorrCod=0005373&contexto=bd&selTa). [Data de acesso: 03-Maio-2012].
- [3] H. K. Huang, *Pacs and Imaging Informatics - Basic Principles and Applications*, First edit. John Wiley & Sons, Inc., 2004.
- [4] G. A. Rose, “Understanding the Need for and Use of Digital Echocardiography,” *GE Medical Systems Ultrasound*, 2003.
- [5] O. S. Pianykh, *Digital imaging and communications in medicine (DICOM)*. Springer, 2008.
- [6] “Memorando de Entendimento Sobre as Condicionalidades de Política Económica,” 2011. [Online]. Disponível em: [www.portugal.gov.pt/media/371372/mou\\_pt\\_20110517.pdf](http://www.portugal.gov.pt/media/371372/mou_pt_20110517.pdf). [Data de acesso: 02-Maio-2012].
- [7] “DICOogle.” [Online]. Disponível em: <http://www.dicoogle.com/>. [Data de acesso: 08-Junho-2012].
- [8] R. Haux, A. Winter, E. Ammenwerth, and B. Brigl, *Strategic information management in hospitals: an introduction to hospital information systems*. Springer, 2004.
- [9] R. V. Velde and P. Degoulet, *Clinical information systems: a component-based approach*. Springer, 2010.
- [10] A. NEMA, “Digital Imaging and Communications in Medicine (DICOM) Part 2 : Conformance,” 2011.
- [11] A. NEMA, “Digital Imaging and Communications in Medicine (DICOM) Part 5 : Data Structures and Encoding,” 2011.
- [12] A. NEMA, “Digital imaging and communications in medicine (DICOM) Part 3: Information object definitions,” 2011.
- [13] H. L. S. International, “Health Level Seven International,” 2012. [Online]. Disponível em: <http://www.hl7.org/>. [Data de acesso: 25-Maio-2012].
- [14] R. Feneck, J. Kneeshaw, and M. Ranucci, *Core Topics in Transesophageal Echocardiography*. Cambridge University Press, 2010.
- [15] J. M. Pisco and L. A. de Sousa, *Noções Fundamentais de Imagiologia*, Primeira E. Lidel - Edições Técnicas, 1999.
- [16] “Hospital da Luz,” 2012. [Online]. Disponível em: <http://www.hospitaldaluz.pt/>. [Data de acesso: 09-Junho-2012].
- [17] Unicordis, “Unicordis - Ecocardiografia,” 2012. [Online]. Disponível em: <http://www.unicordis.com.br/ecocardiografia.php>. [Data de acesso: 09-Junho-2012].
- [18] M. McCandless, E. Hatcher, and O. Gospodnetic, *Lucene in action*. Manning Publications Co., 2010.
- [19] A. S. Foundation, “Apache Lucene,” 2012. [Online]. Disponível em: <http://lucene.apache.org/>. [Data de acesso: 22-Maio-2012].

- [20] C. Costa, L. Bastião, S. Campos, C. Ferreira, A. Silva, and J. L. Oliveira, "Dicoogle - a paradigm change in medical imaging networks," *Diagnostic Imaging Europe*, pp. 22-23, 2011.
- [21] C. Costa, C. Ferreira, L. Bastião, L. Ribeiro, A. Silva, and J. L. Oliveira, "Dicoogle - an Open Source Peer-to-Peer PACS," *Journal of Digital Imaging*, vol. 24, no. 5, pp. 848-856, 2011.
- [22] C. Costa, F. Freitas, M. Pereira, A. Silva, and J. L. Oliveira, "Indexing and retrieving DICOM data in disperse and unstructured archives.," *International journal of computer assisted radiology and surgery*, vol. 4, no. 1, pp. 71-7, Jan. 2009.
- [23] P. G. Nagy, M. Daly, M. Warnock, K. C. Ehlers, and J. Rehm, "PACSPulse: a web-based DICOM network traffic monitor and analysis tool.," *Radiographics : a review publication of the Radiological Society of North America, Inc*, vol. 23, no. 3, pp. 795-801, 2003.
- [24] P. G. Nagy, M. J. Warnock, M. Daly, C. Toland, C. D. Meenan, and R. S. Mezrich, "Informatics in radiology: Automated Web-based Graphical Dashboard for Radiology Operational Business Intelligence.," *Radiographics : a review publication of the Radiological Society of North America, Inc*, vol. 29, no. 7, pp. 1897-906, Nov. 2009.
- [25] M. Hu et al., "Informatics in Radiology: Efficiency Metrics for Imaging Device Productivity," *Radiographics*, vol. 31, pp. 603-616, 2011.
- [26] S. Wang et al., "An automated DICOM database capable of arbitrary data mining (including radiation dose indicators) for quality monitoring.," *Journal of digital imaging : the official journal of the Society for Computer Applications in Radiology*, vol. 24, no. 2, pp. 223-33, Apr. 2011.
- [27] L.-C. M. I. LLC, "DICOMetrix," 2010. [Online]. Disponível em: <http://dicometrix.com>. [Data de acesso: 08-Maio-2012].
- [28] Dvt. Project, "DVTk DICOM Network Analyzer," 2009. [Online]. Disponível em: <http://dicom.dvtk.org/modules/wiwimod/index.php?page=DICOM+Network+Analyzer&cmenu=products>. [Data de acesso: 09-Maio-2012].
- [29] D. Tool, "DICOM Detective," 2010. [Online]. Disponível em: <http://www.dicomtool.com/>. [Data de acesso: 09-Maio-2012].
- [30] Wireshark, "DICOM Protocol," 2010. [Online]. Disponível em: <http://wiki.wireshark.org/Protocols/dicom>. [Data de acesso: 09-Maio-2012].
- [31] J. C. Mazura, K. Juluru, J. J. Chen, T. a Morgan, M. John, and E. L. Siegel, "Facial Recognition Software Success Rates for the Identification of 3D Surface Reconstructed Facial Images: Implications for Patient Privacy and Security.," *Journal of digital imaging : the official journal of the Society for Computer Applications in Radiology*, pp. 11-15, Nov. 2011.
- [32] IETF, "Hypertext Transfer Protocol - HTTP/1.0," 2012. [Online]. Disponível em: <http://www.w3.org/Protocols/HTTP/1.0/draft-ietf-http-spec.html>. [Data de acesso: 28-Maio-2012].
- [33] A. Zúquete, *Segurança em redes informáticas*. FCA, 2006.
- [34] Keita Fujii, "Jpcap," 2007. [Online]. Disponível em: <http://netresearch.ics.uci.edu/kfujii/jpcap/doc/index.html>. [Data de acesso: 15-Maio-2012].
- [35] "dcm4che2 DICOM Toolkit." [Online]. Disponível em: <http://www.dcm4che.org/>. [Data de acesso: 15-Maio-2012].
- [36] "sqlite4java," 2011. [Online]. Disponível em: <http://code.google.com/p/sqlite4java/>. [Data de acesso: 15-Maio-2012].
- [37] O. Corporation, "MySQL::The world's most popular open source database," 2012. [Online]. Disponível em: <http://www.mysql.com/>. [Data de acesso: 20-Maio-2012].
- [38] "Appropriate Uses For SQLite," 2012. [Online]. Disponível em: <http://www.sqlite.org/whentouse.html>. [Data de acesso: 28-Maio-2012].
- [39] "MySQL vs SQLite3 performance," 2012. [Online]. Disponível em: <http://zenadmin.org/en/blog/post720.html>. [Data de acesso: 28-Maio-2012].

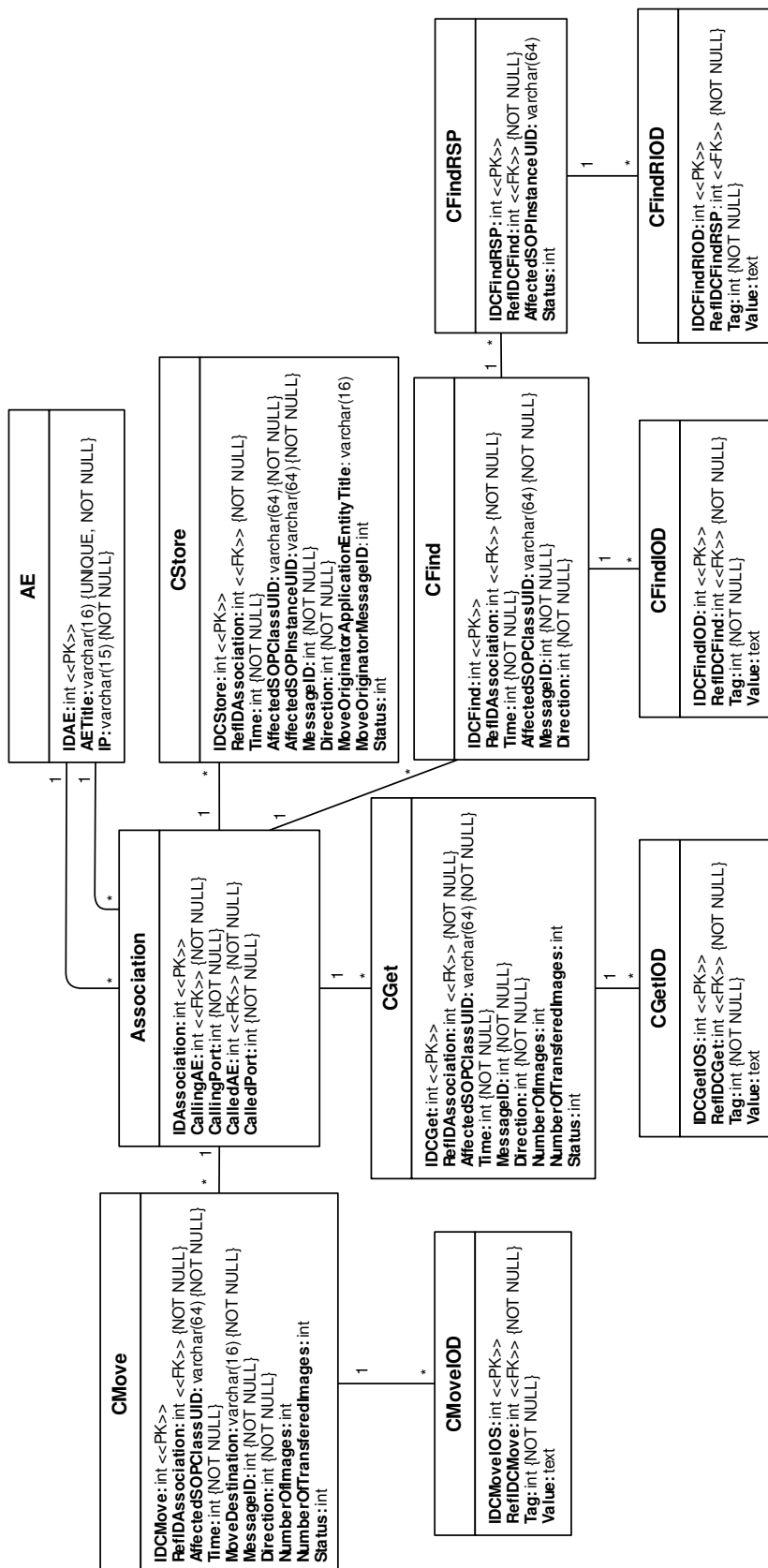


- [40] "Jersey," 2012. [Online]. Disponível em: <http://jersey.java.net/>. [Data de acesso: 15-Maio-2012].
- [41] "Grizzly," 2012. [Online]. Disponível em: <http://grizzly.java.net/>. [Data de acesso: 15-Maio-2012].
- [42] D. D. Clark, "IP DATAGRAM REASSEMBLY ALGORITHMS." IETF Request for Comments (RFC 815), 1982.
- [43] M. Allman, V. Paxson, and E. Blanton, "TCP Congestion Control." IETF Request for Comments (RFC 5681), pp. 1-18, 2009.
- [44] Oracle, "keytool - Key and Certificate Management Tool," 2012. [Online]. Disponível em: <http://docs.oracle.com/javase/6/docs/technotes/tools/solaris/keytool.html>. [Data de acesso: 28-Maio-2012].
- [45] Cisco, "Switched Port Analyzer (SPAN) Configuration Example." [Online]. Disponível em: [http://www.cisco.com/en/US/products/hw/switches/ps708/products\\_tech\\_note09186a008015c612.shtml](http://www.cisco.com/en/US/products/hw/switches/ps708/products_tech_note09186a008015c612.shtml). [Data de acesso: 20-Maio-2012].
- [46] E. Ort and B. Mehta, "Java Architecture for XML Binding (JAXB)," 2003. [Online]. Disponível em: <http://www.oracle.com/technetwork/articles/javase/index-140168.html>. [Data de acesso: 05-Junho-2012].
- [47] Microsoft, "CREATE XML INDEX (Transact-SQL)," 2012. [Online]. Disponível em: <http://msdn.microsoft.com/pt-br/library/bb934097.aspx>. [Data de acesso: 06-Junho-2012].
- [48] Oracle, "Indexing XMLType Data," 2008. [Online]. Disponível em: [http://docs.oracle.com/cd/B28359\\_01/appdev.111/b28369/xdb\\_indexing.htm](http://docs.oracle.com/cd/B28359_01/appdev.111/b28369/xdb_indexing.htm). [Data de acesso: 06-Junho-2012].
- [49] A. S. Foundation, "Apache Solr," 2011. [Online]. Disponível em: <http://lucene.apache.org/solr/>. [Data de acesso: 08-Junho-2012].
- [50] A. S. Foundation, "Apache Tomcat," 2012. [Online]. Disponível em: <http://tomcat.apache.org/>. [Data de acesso: 08-Junho-2012].
- [51] "Java Excel API." [Online]. Disponível em: <http://jexcelapi.sourceforge.net/>. [Data de acesso: 08-Junho-2012].
- [52] Autoridade Tributária e Aduaneira, "Portal das Finanças," 2012. [Online]. Disponível em: <http://www.portaldasfinancas.gov.pt/>. [Data de acesso: 08-Junho-2012].
- [53] "World Web Server Statistics," 2012. [Online]. Disponível em: <http://www.greatstatistics.com>. [Data de acesso: 13-Junho-2012].
- [54] CakePHP, "CakePHP: the rapid development php framework," 2012. [Online]. Disponível em: <http://cakephp.org/>. [Data de acesso: 13-Junho-2012].
- [55] PHPFrameworks.com, "PHP Frameworks," 2012. [Online]. Disponível em: <http://www.phpframeworks.com/>. [Data de acesso: 13-Junho-2012].
- [56] J. J. Garrett, "Ajax: A New Approach to Web Applications," 2005. [Online]. Disponível em: <http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>. [Data de acesso: 01-Jul-2012].
- [57] "PHPExcel," 2012. [Online]. Disponível em: <http://phpexcel.codeplex.com/>. [Data de acesso: 13-Junho-2012].
- [58] "PHPMailer," 2012. [Online]. Disponível em: <http://code.google.com/a/apache-extras.org/p/phpmailer/>. [Data de acesso: 13-Junho-2012].
- [59] T. jQuery Foundation, "jQuery JavaScript Library," 2012. [Online]. Disponível em: <http://jquery.com/>. [Data de acesso: 13-Junho-2012].
- [60] "jQuery File Tree," 2008. [Online]. Disponível em: <http://www.abeautifulsite.net/blog/2008/03/jquery-file-tree/>. [Data de acesso: 13-Junho-2012].
- [61] "DataTables," 2012. [Online]. Disponível em: <http://datatables.net/>. [Data de acesso: 13-Junho-2012].

- [62] Twitter, “Twitter Bootstrap,” 2012. [Online]. Disponível em: <http://twitter.github.com/bootstrap/>. [Data de acesso: 13-Junho-2012].
- [63] T. jQuery Foundation, “jQuery User Interface,” 2012. [Online]. Disponível em: <http://jqueryui.com/>. [Data de acesso: 13-Junho-2012].
- [64] Google, “Google Chart Tools,” 2012. [Online]. Disponível em: <https://developers.google.com/chart/>. [Data de acesso: 13-Junho-2012].
- [65] Google, “Google Maps API,” 2012. [Online]. Disponível em: <https://developers.google.com/maps/?hl=pt-BR>. [Data de acesso: 13-Junho-2012].
- [66] W3C, “Same Origin Policy,” 2010. [Online]. Disponível em: [http://www.w3.org/Security/wiki/Same\\_Origin\\_Policy](http://www.w3.org/Security/wiki/Same_Origin_Policy). [Data de acesso: 14-Junho-2012].
- [67] O. Corporation, “Oracle VM VirtualBox,” 2012. [Online]. Disponível em: <https://www.virtualbox.org/>. [Data de acesso: 28-Junho-2012].
- [68] “TrueCrypt - Free Open-Source On-The-Fly Disk Encryption Software for Windows 7/Vista/XP, Mac OS X and Linux,” 2012. [Online]. Disponível em: <http://www.truecrypt.org/>. [Data de acesso: 28-Junho-2012].
- [69] M. Santos, L. Bastião, C. Costa, A. Silva, and N. Rocha, “DICOM and Clinical Data Mining in a Small Hospital PACS: A Pilot Study,” in *ENTERprise Information Systems*, vol. 221, M. M. Cruz-Cunha, J. Varajão, P. Powell, and R. Martinho, Eds. Springer Berlin Heidelberg, 2011, pp. 254-263.
- [70] I. Boldt, L. Lapão, P. P. Rodrigues, A. Freitas, and R. Cruz-Correia, “Poor quality of Hospital Information Systems audit trails,” in *7ª Conferência Ibérica de Sistemas e Tecnologias de Informação*, 2012, pp. 584-589.

## 10 Anexos

### 10.1. Diagrama físico da base de dados do sensor de rede



## 10.2. Captura e decodificação de pacotes

A captura e decodificação de pacotes recebidos é das tarefas mais importantes que são executadas através do sensor de rede desenvolvido. Este tema foi abordado no subcapítulo 4.3 que apresentou a arquitetura do mesmo. Contudo, aqui serão apresentados mais alguns detalhes sobre a implementação desta tarefa.

A Figura 10.1 apresenta um diagrama com a sequência de invocações entre as classes diretamente envolvidas com a captura e decodificação de pacotes.

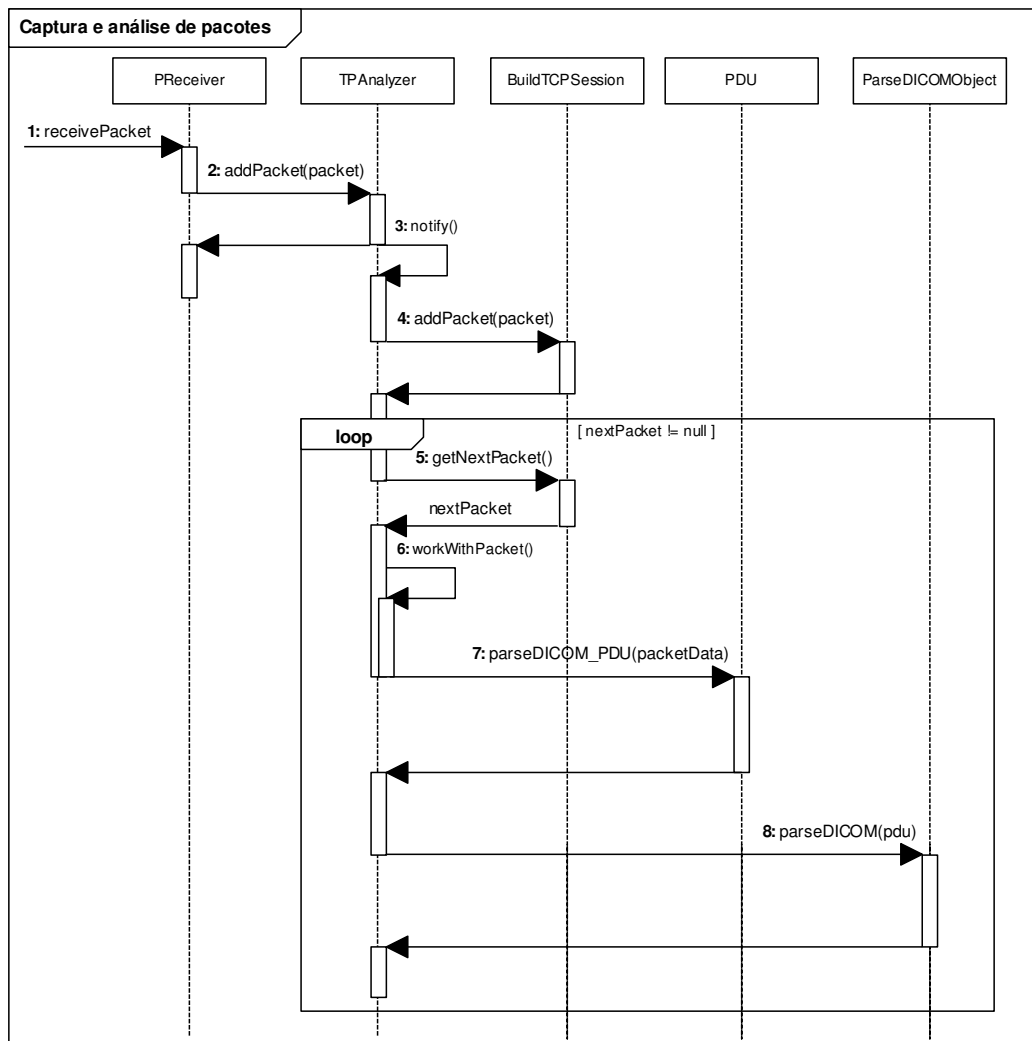


Figura 10.1: Sequência de invocações entre as classes responsáveis pela captura e decodificação de pacotes.

1. A classe *PReceiver* recebe os pacotes capturados pela biblioteca Jpcap, através do método *receivePacket*.
2. Os pacotes TCP recebidos pela classe *PReceiver* são enviados para a classe *TPAnalyzer* que implementa uma *thread* com uma fila de processamento de pacotes.
3. Sempre que é enviado um pacote para a fila de processamento, é invocado o método *notify* da classe *TPAnalyzer* que retira a *thread* de um eventual estado de inatividade.
4. A classe *TPAnalyzer* adiciona o pacote na lista de pacotes da classe *BuildTCPSession*, que é responsável por reconstruir a sessão TCP. Uma vez que os pacotes retornados

pela classe *BuildTCPSession* têm que ser ordenados e desfragmentados, podem ser retornados vários pacotes seguidos pela classe *BuildTCPSession*. Por este motivo, os passos 5 a 8 são invocados dentro de um ciclo.

5. O passo 5 é precisamente obter o próximo pacote da sessão TCP. Caso os pacotes da sessão TCP estejam fora de ordem, a classe *BuildTCPSession* não irá retornar nenhum pacote até receber o próximo pacote, tendo em conta o seu número de sequência.
6. Sempre que a classe *BuildTCPSession* retorna um novo pacote, é invocado o método *workWithPacket* que irá tratar o pacote da sessão TCP.
7. Os bytes contidos na zona de dados do pacote TCP são enviados para a classe *PDU*. Esta irá usar as classes da Figura 4.3, caso o pacote seja um PDU DICOM.
8. Por fim, se for um pacote DICOM, o PDU é enviado para a classe *ParseDICOMObject* que é responsável pela extração de informação relativa ao DIMSE. A Figura 10.2 apresenta a sequência dentro deste passo, quando o PDU é do tipo P-Data-TF.

A Figura 10.2 apresenta o diagrama de sequência relativo à análise de objetos DICOM. A classe *Association* é aquela que interage com a base de dados e com as classes de processamento dos objetos DICOM, conforme se pode ver na Figura 4.3.

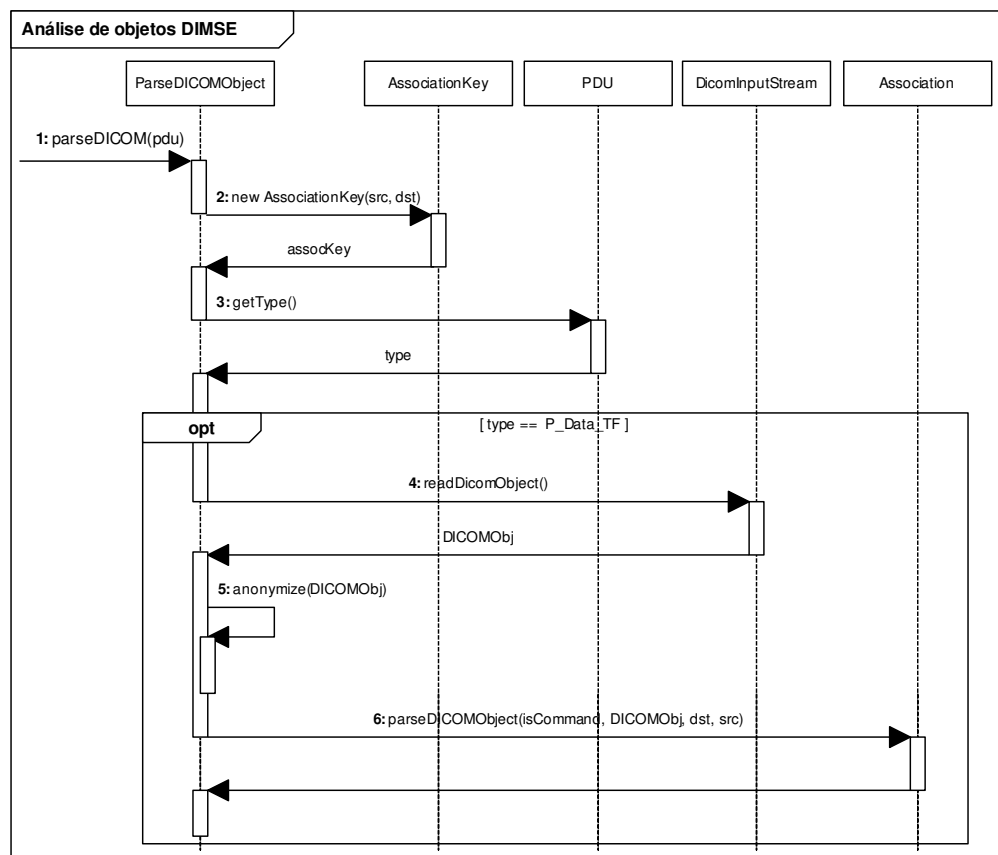


Figura 10.2: Análise de objetos DICOM DIMSE quando o PDU é do tipo P-Data-TF e ocorre anonimização da informação recolhida pelo sensor.

### 10.3. Construção de *Queries*

Com o sistema desenvolvido, os utilizadores podem estudar a informação armazenada nos vários sensores apresentados. Para tirar o máximo partido do sistema desenvolvido, é

importante que os utilizadores consigam construir as *queries* de pesquisa adequadamente. Este anexo apresenta alguns exemplos de possíveis *queries* que poderão ser adaptados consoante as necessidades dos utilizadores.

As figuras do subcapítulo 6.3.3, apresentam a interface utilizada para pesquisar nos vários sensores. O sensor de imagens médicas e o sensor de relatórios de ecocardiogramas suportam ambos dois tipos de *queries* diferentes. O utilizador pode construir *queries* simples de texto livre ou *queries* avançadas, que seguem algumas regras, usam palavras chave e os nomes dos campos que contêm a informação. O sensor de rede não suporta, por enquanto, *queries* em texto livre.

Para cada tipo de sensor, existe um botão chamado “Search Tips” que apresenta uma pequena janela para ajudar os utilizadores a construir as suas *queries*, conforme podemos ver na Figura 10.3. Se o utilizador começar a escrever o nome de um campo na caixa de texto da janela, automaticamente a aplicação irá apresentar sugestões de acordo com a lista de todos os campos de informação suportados pelo sensor.

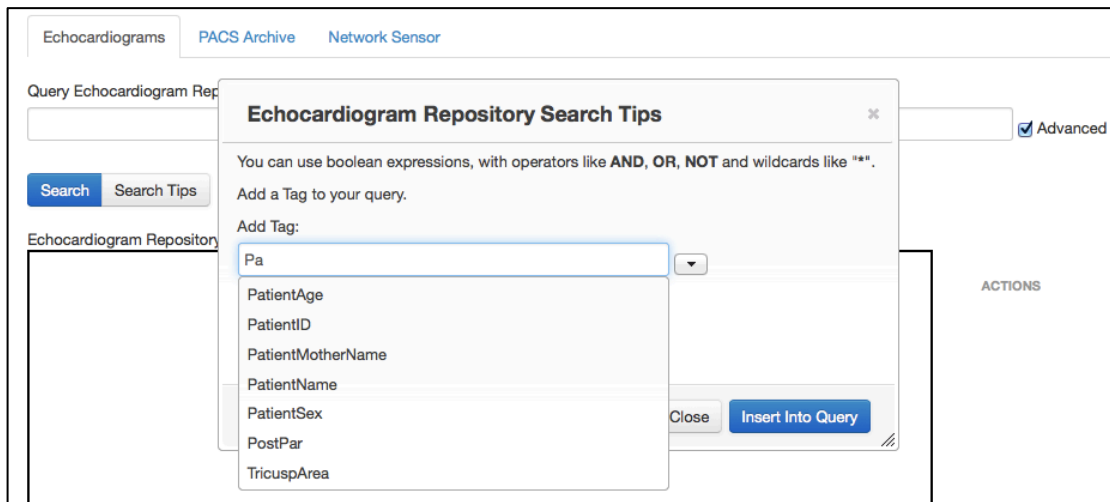


Figura 10.3: Janela de ajuda à construção de *queries* para os sensores de relatórios de ecocardiogramas.

Ambos os tipos de *queries* suportam *wildcards*, como por exemplo o carácter “\*”. Nas *queries* avançadas, o utilizado pode usar os operadores lógicos **AND**, **OR** e **NOT**, bem como os parêntesis para definir precedências. A lista seguinte apresenta alguns exemplos de *queries* enviadas ao sensor de relatórios de ecocardiogramas seguidas de uma breve descrição:

- “**Alex\***”: pesquisa em texto livre que retorna todos os documentos em que qualquer um dos campos indexados comece pelos caracteres “**Alex**” .
- “**PatientName:Matos**”: retorna todos os documentos que digam respeito a pacientes com o nome “**Matos**”. Note-se que o nome completo de um paciente é composto por vários nomes.
- “**ContentType:ETE AND Weight:Numeric:[70 TO 90]**”: Retorna todos os relatórios de ecocardiogramas transefágicos (ETE) cujo peso do paciente esteja entre os 70 e os 90 Kg, inclusive.
- “**ContentType:ETT AND ExamDate:199907\* AND NOT MotDescr:Arritmia**”: Retorna todos os relatórios de ecocardiogramas transtorácicos feitos em Julho de 1999 cujo motivo para realização do exame não seja “Arritmia”.