

ALEXANDRE SAVARIS

UMA PROPOSTA DE ARQUITETURA DE ALTO DESEMPENHO PARA
SISTEMAS PACS BASEADA EM EXTENSÕES DE BANCO DE DADOS

Tese apresentada ao Programa de Pós-Graduação
em Informática, Setor de Ciências Exatas,
Universidade Federal do Paraná, como requisito
parcial à obtenção do título de Doutor em
Informática.

Orientador: Prof. Dr. rer.nat. Aldo von Wangenheim

CURITIBA
2014

S265p

Savaris, Alexandre

Uma proposta de arquitetura de alto desempenho para sistemas PACS baseada em extensões de banco de dados / Alexandre Savaris. – Curitiba, 2014.

154f. : il. color. ; 30 cm.

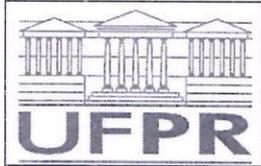
Tese - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-graduação em Informática, 2014.

Orientador: Aldo von Wangenheim .

Bibliografia: p. 116-128.

1. Sistemas de arquivamento e comunicação de imagens. 2. Diagnóstico por Imagem. 3. Comunicação na medicina. 4. Bancos de dados. I. Universidade Federal do Paraná. II. Wangenheim, Aldo von. III. Título.

CDD: 005.743



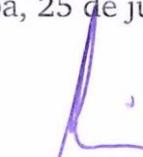
Ministério da Educação
Universidade Federal do Paraná
Programa de Pós-Graduação em Informática

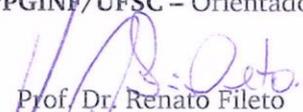
PARECER

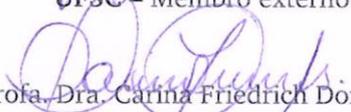
Nós, abaixo assinados, membros da Banca Examinadora da defesa do aluno de Doutorado em Ciência da Computação, Alexandre Savaris, avaliamos a tese de doutorado intitulada “Uma proposta de arquitetura de alto desempenho para sistemas pacs baseada em extensões de banco de dados”, cuja defesa pública foi realizada no dia 25 de julho de 2014, às 08:30 horas, no Departamento de Informática do Setor de Ciências Exatas da Universidade Federal do Paraná. Após avaliação, decidimos pela:

aprovação do candidato. **reprovação** do candidato.

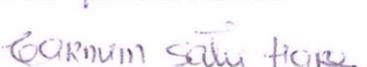
Curitiba, 25 de julho de 2014.

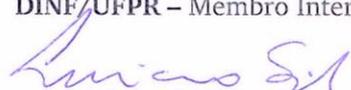

Prof. Dr. Aldo von Wangenheim
PPGINF/UFSC – Orientador


Prof. Dr. Renato Fileto
UFSC – Membro externo


Profa. Dra. Carina Friedrich Dorneles
UFSC – Membro Externo


Prof. Dr. Mario Antonio Ribeiro Dantas
UFSC – Membro Externo


Profa. Dra. Carmem Satie Hara
DINF/UFPR – Membro Interno


Prof. Dr. Luciano Silva
DINF/UFPR – Membro Interno



À minha família.

AGRADECIMENTOS

À Milena, por ter “segurado a barra” durante minha ausência nos 16 meses de doutorado sanduíche.

Ao programa conjunto de bolsas de doutorado CNPq/CAPES/DAAD, pela oportunidade de conhecer a face organizada de um processo de pesquisa.

A Thomas Fiedlschuster-Köhne e Petra Kreuter, em Münster, Alemanha, pela recepção e convivência nos quatro primeiros meses de viagem.

Ao professor Theo Härder e seu grupo de pesquisa na Universidade Técnica de Kaiserslautern, Alemanha, pela acolhida durante o período de doutorado sanduíche.

Aos professores componentes da banca examinadora, pelas correções e sugestões direcionadas à melhoria da tese.

Aos integrantes da equipe do laboratório de Telemedicina, localizado no HU/UFSC, pela perseverança na manutenção de um serviço que faz a diferença.

Finalmente, a todas as pessoas que mantêm-se próximas.

Time discovers truth.

Seneca

RESUMO

O uso de imagens digitais no processo de diagnóstico médico é observável em diferentes escalas e cenários de aplicação, tendo evoluído em termos de volume de dados adquiridos e número de modalidades de exame atendidas. A organização desse conteúdo digital, comumente representado por conjuntos de imagens no padrão DICOM (*Digital Imaging and Communications in Medicine*), costuma ser delegada a sistemas PACS (*Picture Archiving and Communication System*) baseados na agregação de componentes heterogêneos de *hardware* e *software*. Parte desses componentes interage de forma a compor a *camada de armazenamento* do PACS, responsável pela persistência de toda e qualquer imagem digital que, em algum momento, foi adquirida ou visualizada/manipulada via sistema. Apesar de empregarem recursos altamente especializados como SGBDs (Sistemas Gerenciadores de Banco de Dados), as camadas de armazenamento PACS atuais são visualizadas e utilizadas como simples repositórios de dados, assumindo um comportamento passivo (ou seja, sem a agregação de regras de negócio) quando comparadas a outros componentes do sistema.

Neste trabalho, propõe-se uma nova arquitetura PACS simplificada baseada em alterações na sua camada de armazenamento. As alterações previstas baseiam-se na troca do perfil passivo assumido atualmente por essa camada por um perfil ativo, utilizando-se de recursos de extensibilidade e de distribuição de dados (hoje não empregados) disponibilizados por seus componentes. A arquitetura proposta concentra-se na comunicação e no armazenamento de dados, utilizando-se de extensões de SGBDs e de estruturas heterogêneas para armazenamento de dados convencionais e não convencionais, provendo alto desempenho em termos de escalabilidade, suporte a grandes volumes de conteúdo e processamento descentralizado de consultas. Estruturalmente, a arquitetura proposta é formada por um conjunto de módulos projetados de forma a explorar as opções de extensibilidade presentes em SGBDs, incorporando características e funcionalidades originalmente distribuídas entre outros componentes do PACS (na forma de regras de negócio).

Em nível de protótipo, resultados obtidos a partir de experimentos indicam a viabilidade de uso da arquitetura proposta, explicitando ganhos de desempenho na pesquisa de metadados e na recuperação de imagens DICOM quando comparados a arquiteturas PACS convencionais. A flexibilidade da proposta quanto à adoção de

tecnologias de armazenamento heterogêneas também é avaliada positivamente, permitindo estender a camada de armazenamento PACS em termos de escalabilidade, poder de processamento, tolerância a falhas e representação de conteúdo.

Palavras-chave: PACS, DICOM, SGBD, extensibilidade, alto desempenho.

ABSTRACT

The use of digital images on medical diagnosis is observable in a number of application scenarios and in different scales, growing in terms of volume of data and contemplated medical specialties. To organize this digital content composed by image datasets in DICOM (Digital Imaging and Communications in Medicine), it is usual to adopt PACS (Picture Archiving and Communication System), an architecture built as an aggregation of hardware and software components. Some of these components compose the so-called PACS's storage layer, responsible for the persistence of every digital image acquired or visualized/manipulated through the system. Despite their high-specialized components (e.g., DBMS – Database Management System), PACS storage layers used today are visualized as simple data repositories, assuming a passive role (i.e., without the implementation of business rules) when compared to other components.

In this work, a simplified, new architecture is proposed for PACS, based in modifications on its storage layer. The modifications are based in the replacement of the current passive role by an active one, using extensibility and data distribution resources available on its components. The proposed architecture focuses on communication and data storage, using DBMS extensions and heterogeneous structures for the storage of conventional and non-conventional data, providing high-performance in terms of scalability, support to large volumes of data and decentralized query processing. Structurally, the proposed architecture is composed by a set of modules designed to explore extensibility options available in DBMSs, incorporating characteristics and functionalities originally distributed as business rules among other components of PACS.

At prototype level, results obtained through experiments indicate the viability of the proposal, making explicit the performance gains in the search for metadata and image retrieval when compared to conventional PACS architectures. The flexibility of the proposal regarding the adoption of heterogeneous storage technologies is also positively evaluated, allowing the extension of the PACS storage layer in terms of scalability, processing power, fault tolerance and content representation.

Keywords: PACS, DICOM, DBMS, extensibility, High-Performance Computing.

LISTA DE FIGURAS

FIGURA 1 – EXEMPLO DE ARQUITETURA PACS (DESTAQUE PARA A ORGANIZAÇÃO MODULAR DAS ENTIDADES DE APLICAÇÃO). FONTE: (HECKMAN; SCHULTZ, 2006) (modificado pelo autor).....	25
FIGURA 2 – EXEMPLO DE ORGANIZAÇÃO DE CONTEÚDO DICOM. EM SEU NÍVEL MAIS ALTO (ARQUIVO), UMA IMAGEM É COMPOSTA POR UM CONJUNTO DE TAGS. EM SEU NÍVEL MAIS BAIXO (TAG), UM ELEMENTO DE DADOS É COMPOSTO POR NÚMEROS DE GRUPO E ELEMENTO, ALÉM DE UM VALOR DE REPRESENTAÇÃO, MULTIPLICIDADE E CONTEÚDO. FONTE: (YU, 2008) (modificado pelo autor).....	32
FIGURA 3 – EXTENSÃO SQL/MED, DE ACORDO COM SEU PROJETO CONCEITUAL. À ESQUERDA, COMPONENTES DA WRAPPER INTERFACE. À DIREITA, O TIPO DE DADO DATALINK. FONTE: (MELTON, 2001)	37
FIGURA 4 – EXEMPLO DE DECOMPOSIÇÃO DE UMA RELAÇÃO HORIZONTAL EM UM CONJUNTO DE RELAÇÕES BINÁRIAS SIMPLES. FONTE: (COPELAND; KHOSHAFIAN, 1985) (modificado pelo autor) ...	38
FIGURA 5 – EXEMPLO DE ARMAZENAMENTO DE CONTEÚDO DICOM EM SISTEMAS DE ARQUIVOS CONVENCIONAIS. À ESQUERDA: ORGANIZAÇÃO DE UMA ÁRVORE DE DIRETÓRIOS EM UM NAS (NETWORK-ATTACHED STORAGE). À DIREITA: VISÕES DIFERENCIADAS CONSTRUÍDAS SOBRE UMA MESMA ESTRUTURA FÍSICA. FONTE: (YAKAMI, 2011)	45
FIGURA 6 – EXEMPLO DE ARMAZENAMENTO DE CONTEÚDO DICOM EM UMA ESTRUTURA COMBINADA DE MODELO DE DADOS HDF E SISTEMA DE ARQUIVOS PVFS. FONTE: (SOARES, 2012) .	46
FIGURA 7 – EXEMPLO DE UM CATÁLOGO DE METADADOS EM UM AMBIENTE DISTRIBUÍDO. NA PARTE SUPERIOR: COMPONENTES DO CATÁLOGO E SUA DISTRIBUIÇÃO. NA PARTE INFERIOR: ESQUEMA RELACIONAL USADO COMO BACK-END PARA PERSISTÊNCIA DOS METADADOS. FONTE: (KOBLOITZ; SANTOS; POSE, 2008)	47
FIGURA 8 – REPOSITÓRIO DICOM PARA O PROJETO eDiaMoND. NA PARTE SUPERIOR: EXEMPLO DE ORGANIZAÇÃO HIERÁRQUICA PARA O PADRÃO DICOM. NA PARTE INFERIOR: MODELO RELACIONAL PARA A ORGANIZAÇÃO HIERÁRQUICA. FONTE: (POWER, 2004)	62
FIGURA 9 – REPOSITÓRIO DICOM CONSTRUÍDO COMO DAL (DATA ACCESS LAYER) ACESSANDO UM BANCO DE DADOS RELACIONAL. FONTE: (CHANDRASHEKAR, 2006) (modificado pelo autor)	63
FIGURA 10 – REVISÃO SISTEMÁTICA DA LITERATURA EM NÚMEROS. À ESQUERDA: RESULTADO DA APLICAÇÃO DE CRITÉRIOS DE INCLUSÃO, EXCLUSÃO E ACEITAÇÃO. À DIREITA: PESO DOS CRITÉRIOS DE ACEITAÇÃO NA ESCOLHA DE REFERÊNCIAS RELEVANTES. FONTE: o autor (2012)	65
FIGURA 11 – PROPOSTA DE ARQUITETURA SIMPLIFICADA E DE ALTO DESEMPENHO PARA SISTEMAS PACS. DESTAQUE PARA A INCORPORAÇÃO DE SERVIÇOS NA CAMADA DE ARMAZENAMENTO (SGBD), PELA ADOÇÃO DE MÓDULOS E EXTENSÕES. FONTE: o autor (2014)	68
FIGURA 12 – DETALHAMENTO DOS ELEMENTOS QUE COMPÕEM A ARQUITETURA PROPOSTA. DESTAQUE PARA O FLUXO DE DADOS PREVISTO ENTRE OS ELEMENTOS EXTERNOS E INTERNOS AO SGBD. FONTE: o autor (2014).....	70

FIGURA 13 – EXTRATO DO ESQUEMA DE BD BASEADO NO MODELO DE ARMAZENAMENTO DECOMPOSTO. O ESQUEMA É CENTRALIZADO NA TABELA CHAVE_HIERARQUICA, COM A QUAL AS DEMAIS TABELAS SE RELACIONAM. FONTE: o autor (2014)	76
FIGURA 14 – ARQUITETURA MODULAR DO SGBD MYSQL, INCLUINDO <i>STORAGE ENGINES</i> CUSTOMIZADOS. FONTE: o autor (2012)	78
FIGURA 15 – TEMPO DE ARMAZENAMENTO DE <i>DATASETS</i> COMPLETOS POR CONFIGURAÇÃO E MODALIDADE DE EXAME. FONTE: o autor (2014)	82
FIGURA 16 – TEMPO DE PESQUISA DE METADADOS POR ESTRATÉGIA DE ARMAZENAMENTO E MODALIDADE DE EXAME. FONTE: o autor (2014)	84
FIGURA 17 – TEMPO DE PESQUISA DE METADADOS POR ESTRATÉGIA DE ARMAZENAMENTO E GRUPOS DE <i>TAGS</i> (TOMOGRÁFIA COMPUTADORIZADA). FONTE: o autor (2014)	86
FIGURA 18 – TEMPO DE RECUPERAÇÃO DE IMAGENS COMPLETAS POR ESTRATÉGIA DE ARMAZENAMENTO E MODALIDADE DE EXAME. FONTE: o autor (2014)	87
FIGURA 19 – <i>CLUSTER</i> HOMOGÊNEO UTILIZADO NOS EXPERIMENTOS DE REPLICAÇÃO E DISTRIBUIÇÃO DE CONTEÚDO. FONTE: o autor (2014)	89
FIGURA 20 – TEMPO DE ARMAZENAMENTO PARA IMAGENS DE CT POR CONFIGURAÇÃO, NÚMERO DE PROCESSOS CONCORRENTES E ETAPA DE GRAVAÇÃO (<i>PARSING</i> , <i>TAGS</i> INDIVIDUAIS E CONTEÚDO COMPLETO). FONTE: o autor (2014)	92
FIGURA 21 – TEMPO DE PESQUISA DE METADADOS POR CONFIGURAÇÃO, NÚMERO DE PROCESSOS CONCORRENTES E ALTERNATIVA DE BUSCA (NÍVEL HIERÁRQUICO DE PACIENTE). FONTE: o autor (2014)	94
FIGURA 22 – TEMPO DE PESQUISA DE METADADOS POR CONFIGURAÇÃO, NÚMERO DE PROCESSOS CONCORRENTES E ALTERNATIVA DE BUSCA (NÍVEL HIERÁRQUICO DE ESTUDO). FONTE: o autor (2014)	95
FIGURA 23 – TEMPO DE PESQUISA DE METADADOS POR CONFIGURAÇÃO, NÚMERO DE PROCESSOS CONCORRENTES E ALTERNATIVA DE BUSCA (NÍVEL HIERÁRQUICO DE SÉRIE). FONTE: o autor (2014)	95
FIGURA 24 – TEMPO DE PESQUISA DE METADADOS POR CONFIGURAÇÃO, NÚMERO DE PROCESSOS CONCORRENTES E ALTERNATIVA DE BUSCA (NÍVEL HIERÁRQUICO DE IMAGEM). FONTE: o autor (2014)	96
FIGURA 25 – TEMPO DE RECUPERAÇÃO DE CONTEÚDO POR CONFIGURAÇÃO, NÚMERO DE PROCESSOS CONCORRENTES E NÍVEL HIERÁRQUICO. FONTE: o autor (2014)	97
FIGURA 26 – ESQUEMA DE BD IMPLEMENTADO SOBRE APACHE CASSANDRA E ACESSADO VIA INSTÂNCIA RELACIONAL, POR MEIO DE SUA IMPLEMENTAÇÃO SQL/MED. FONTE: o autor (2014) ...	99
FIGURA 27 – CENÁRIO COMPLETO PARA O AMBIENTE EXPERIMENTAL #3, CONTEMPLANDO A INTEGRAÇÃO ENTRE AS INSTÂNCIAS RELACIONAL E NoSQL. FONTE: o autor (2014)	101
FIGURA 28 – TEMPO DE ARMAZENAMENTO DE <i>DATASETS</i> COMPLETOS POR MODALIDADE DE EXAME, EM CENÁRIOS COM VARIÂNCIA NO NÚMERO DE PROCESSOS DE ESCRITA E DISTRIBUIÇÃO DE DADOS. FONTE: o autor (2014)	103
FIGURA 29 – TEMPO DE PESQUISA DE METADADOS POR MODALIDADE DE EXAME, RELACIONADA À DISTRIBUIÇÃO DE DADOS E NÍVEL HIERÁRQUICO. FONTE: o autor (2014)	104

FIGURA 30 – TEMPO DE RECUPERAÇÃO DE CONTEÚDO POR MODALIDADE DE EXAME, RELACIONADA À DISTRIBUIÇÃO DE DADOS E NÍVEL HIERÁRQUICO. FONTE: o autor (2014) 106

FIGURA 31 – TRABALHOS FUTUROS INTEGRADOS À PROPOSTA DE ARQUITETURA PACS. DESTAQUE PARA A INCORPORAÇÃO DE SERVIÇOS DE PROCESSAMENTO DE ALTO DESEMPENHO E PARA A DISTRIBUIÇÃO TEMPORAL DE DADOS *ONLINE*, *NEAR-LINE* E *OFFLINE*. FONTE: o autor (2014)..... 114

LISTA DE QUADROS

QUADRO 1 - TRABALHOS EXCLUÍDOS DA REVISÃO SISTEMÁTICA PELO NÃO ATENDIMENTO AOS CRITÉRIOS DEFINIDOS. FONTE: o autor (2012)	60
QUADRO 2 - SÍNTESE DAS CARACTERÍSTICAS RELEVANTES PARA OS TRABALHOS AVALIADOS. FONTE: o autor (2012).....	61
QUADRO 3 – DIFERENÇAS ENTRE O DSM ORIGINAL E O ESQUEMA DE BD PROPOSTO NO PRESENTE TRABALHO. FONTE: o autor (2014)	76
QUADRO 4 – CONFIGURAÇÕES DE <i>HARDWARE</i> E <i>SOFTWARE</i> PARA OS EXPERIMENTOS DE REPLICAÇÃO E DISTRIBUIÇÃO DE CONTEÚDO. FONTE: o autor (2014)	90
QUADRO 5 – CONFIGURAÇÕES DE <i>HARDWARE</i> E <i>SOFTWARE</i> PARA OS EXPERIMENTOS ENVOLVENDO TECNOLOGIAS HETEROGÊNEAS. FONTE: o autor (2014)	101

LISTA DE TABELAS

TABELA 1 – <i>DATASETS</i> UTILIZADOS NA PRIMEIRA RODADA DE TESTES DO ESQUEMA + EXTENSÃO DE BD	80
TABELA 2 – <i>DATASETS</i> UTILIZADOS NA SEGUNDA RODADA DE TESTES DO ESQUEMA DE BD BASEADO NO MODELO DE ARMAZENAMENTO DECOMPOSTO	91

LISTA DE ABREVIATURAS E SIGLAS

3D	-	Tridimensional
ACR	-	<i>American College of Radiology</i>
ADT	-	<i>Abstract Data Type</i>
AE	-	<i>Application Entity</i>
AIM	-	<i>Advanced Image Management</i>
API	-	<i>Application Programming Interface</i>
BD	-	Banco de Dados
BIMM	-	<i>Biomedical Image Metadata Manager</i>
BIWS	-	<i>Bio-Image Warehouse System</i>
BLOB	-	<i>Binary Large Object</i>
CA	-	<i>Continuous Available</i>
CBIR	-	<i>Content-Based Image Retrieval</i>
CE	-	Configuração Estendida
CL	-	<i>Cluster</i>
CP	-	Configuração Padrão
CR	-	<i>Computed Radiology</i>
CRUD	-	<i>Create, Read, Update, Delete</i>
CT	-	<i>Computed Tomography</i>
DAL	-	<i>Data Access Layer</i>
DICOM	-	<i>Digital Imaging and Communications in Medicine</i>
DIMSE	-	<i>DICOM Message Service Element</i>
DML	-	<i>Data Manipulation Language</i>
DSM	-	<i>Decomposed Storage Model</i>
DVD	-	<i>Digital Video Disk</i>
ER	-	Entidade-Relacionamento
FDW	-	<i>Foreign Data Wrapper</i>
FS	-	<i>Foreign Server</i>
FT	-	<i>Foreign Table</i>
GIN	-	<i>Generalized Inverted Index</i>
GiST	-	<i>Generalized Search Tree</i>

GPGPU	-	<i>General-Purpose computation on Graphics Processing Units</i>
GTM	-	<i>Global Transaction Manager</i>
HDF	-	<i>Hierarchical Data Format</i>
HIS	-	<i>Hospital Information System</i>
HL7	-	<i>Health Level Seven</i>
HPC	-	<i>High-Performance Computing</i>
IaaS	-	<i>Infrastructure as a Service</i>
IBM	-	<i>International Business Machines</i>
INU	-	<i>Instância de Nódo Único</i>
IOD	-	<i>Information Object Definition</i>
ISO	-	<i>International Organization for Standardization</i>
JDBC	-	<i>Java Database Connectivity</i>
MED	-	<i>Management of External Data</i>
MIFAS	-	<i>Medical Image File Accessing System for Co-allocation Data Grids</i>
MPE	-	<i>Múltiplos Processos de Escrita</i>
MPP	-	<i>Massively Parallel Processor</i>
MR	-	<i>Magnetic Resonance</i>
MRI	-	<i>Magnetic Resonance Imaging</i>
MVCC	-	<i>Multi-Version Concurrency Control</i>
MWL	-	<i>DICOM Modality Worklist</i>
NAS	-	<i>Network-Attached Storage</i>
NEMA	-	<i>National Electrical Manufacturers Association</i>
NetCDF	-	<i>Network Common Data Format</i>
ODBC	-	<i>Open Database Connectivity</i>
PaaS	-	<i>Platform as a Service</i>
PACS	-	<i>Picture Archiving and Communication System</i>
PET	-	<i>Positron Emission Tomography</i>
PEU	-	<i>Processo de Escrita Único</i>
PVFS	-	<i>Parallel Virtual File System</i>
RIS	-	<i>Radiology Information System</i>
SAAP	-	<i>Service Architecture for Archiving and Presentation</i>
SaaS	-	<i>Software as a Service</i>

SC	-	<i>Secondary Capture</i>
SCP	-	<i>Service Class Provider</i>
SCU	-	<i>Service Class User</i>
SGBD	-	<i>Sistema Gerenciador de Banco de Dados</i>
SMP	-	<i>Symmetric Multiprocessor</i>
SOP	-	<i>Service-Object Pair</i>
SPI	-	<i>Server Programming Interface</i>
SPOF	-	<i>Single Point Of Failure</i>
SQL	-	<i>Structured Query Language</i>
UDF	-	<i>User-Defined Function</i>
VM	-	<i>Value Multiplicity</i>
VR	-	<i>Value Representation</i>
XA	-	<i>X-Ray Angiography</i>
XML	-	<i>Extensible Markup Language</i>

SUMÁRIO

1 INTRODUÇÃO	21
1.1 CONTEXTUALIZAÇÃO.....	23
1.2 DESCRIÇÃO DO PROBLEMA.....	26
1.3 HIPÓTESES E OBJETIVOS.....	27
1.3.1 Hipótese #1	28
1.3.2 Hipótese #2	28
1.3.3 Hipótese #3	28
1.4 ORGANIZAÇÃO DO TRABALHO.....	29
2 FUNDAMENTAÇÃO TEÓRICA.....	30
2.1 ORGANIZAÇÃO DE CONTEÚDO NO PADRÃO DICOM.....	30
2.2 PESQUISA E RECUPERAÇÃO DE CONTEÚDO NO PADRÃO DICOM.....	32
2.3 EXTENSIBILIDADE EM SGBDs.....	34
2.4 EXTENSÃO SQL/MED	35
2.5 <i>DECOMPOSED STORAGE MODEL</i>	37
2.6 COMPUTAÇÃO DE ALTO DESEMPENHO.....	39
2.6.1 Escalabilidade.....	40
2.6.2 Tolerância a falhas.....	41
3 REVISÃO DE LITERATURA	44
3.1 ARMAZENAMENTO DE IMAGENS MÉDICAS NO PADRÃO DICOM EM ARQUITETURAS PACS – ESTRATÉGIAS DIVERSAS	44
3.2 ARMAZENAMENTO DE IMAGENS MÉDICAS NO PADRÃO DICOM EM ARQUITETURAS PACS – USO DE SGBDs	47
3.2.1 Contextualização	48
3.2.2 Questões de pesquisa	48
3.2.3 Definição e organização dos termos de busca.....	49
3.2.4 Critérios de aceitação das referências	49
3.2.5 Procedimentos para avaliação das referências.....	49
3.2.6 Estratégias para a extração de dados	50
3.2.7 Estratégias para a síntese de dados	50
3.2.8 Seleção de referências	51
3.2.9 Identificação de referências relevantes	51
3.2.10 Exclusão de referências pelo não atendimento aos critérios de aceitação	51

3.2.11 Síntese do conteúdo das referências selecionadas.....	60
3.2.11.1 <i>A relational approach to the capture of DICOM files for Grid-enabled medical imaging databases</i> (POWER, 2004).....	61
3.2.11.2 <i>COTS-Like Generic Medical Image Repository</i> (CHANDRASHEKAR, 2006)	63
3.2.12 Discussão	64
4 MATERIAIS E MÉTODOS	67
4.1 ARQUITETURA PROPOSTA.....	67
4.2 EXTENSÃO DE SGBD PARA RECEPÇÃO E ENVIO DE MENSAGENS NO PADRÃO DICOM.....	69
4.2.1 <i>MySQL Daemon Plugins</i>	71
4.2.2 <i>PostgreSQL Custom Background Workers</i>	71
4.2.3 Esquema de BD.....	72
4.2.3.1 Esquema de BD baseado no modelo de armazenamento decomposto	73
4.3 DISTRIBUIÇÃO DE DADOS ENTRE TECNOLOGIAS HETEROGÊNEAS DE ARMAZENAMENTO	77
4.3.1 <i>PostgreSQL FDWs</i>	77
4.3.2 <i>MySQL Storage Engines</i>	78
4.4 CONSIDERAÇÕES SOBRE O CAPÍTULO	79
5 EXPERIMENTOS, RESULTADOS E DISCUSSÕES	80
5.1 AMBIENTE EXPERIMENTAL #1.....	80
5.1.1 Resultados e discussão – armazenamento (C-STORE)	81
5.1.2 Resultados e discussão – pesquisa (C-FIND)	83
5.1.3 Resultados e discussão – recuperação (C-GET).....	86
5.2 AMBIENTE EXPERIMENTAL #2.....	88
5.2.1 Resultados e discussão – armazenamento (C-STORE)	91
5.2.2 Resultados e discussão – pesquisa (C-FIND)	93
5.2.3 Resultados e discussão – recuperação (C-GET).....	96
5.3 AMBIENTE EXPERIMENTAL #3.....	97
5.3.1 Resultados e discussão – armazenamento (C-STORE)	102
5.3.2 Resultados e discussão – pesquisa (C-FIND)	103
5.3.3 Resultados e discussão – recuperação (C-GET).....	105
5.4 CONSIDERAÇÕES SOBRE O CAPÍTULO	106
6 CONCLUSÕES E TRABALHOS FUTUROS	108
6.1 HIPÓTESE #1 – ESQUEMA DE BD.....	108
6.2 HIPÓTESE #2 – EXTENSÃO DE SGBD.....	109
6.3 HIPÓTESE #3 – ARMAZENAMENTO HETEROGÊNEO DE ALTO DESEMPENHO	110

6.4 CONTRIBUIÇÕES	111
6.5 TRABALHOS FUTUROS.....	113
REFERÊNCIAS.....	116
GLOSSÁRIO DE TERMOS E EXPRESSÕES	129
APÊNDICE 1 – EXPRESSÕES DE BUSCA GERADAS PELAS INTERFACES DE ACESSO ÀS BIBLIOTECAS DIGITAIS.....	132
<i>IEEE Xplore</i>	132
<i>SpringerLink</i>	132
<i>ACM Digital Library</i>	133
<i>ScienceDirect</i>	133
<i>Wiley</i>	133
<i>Web of Knowledge</i>	134
APÊNDICE 2 – DICIONÁRIO DE DADOS DO ESQUEMA DE BD PARA ARMAZENAMENTO DICOM	135
APÊNDICE 3 – EXEMPLOS DE INSTRUÇÕES SQL USADAS NOS EXPERIMENTOS DE PESQUISA DE METADADOS	150
APÊNDICE 4 – EXEMPLOS DE INSTRUÇÕES SQL USADAS NOS EXPERIMENTOS DE RECUPERAÇÃO DE CONTEÚDO.....	152
ANEXO 1 – TAGS USADAS NA EXECUÇÃO DE EXPERIMENTOS DE PESQUISA	153

1 INTRODUÇÃO

Diferentes especialidades médicas têm se beneficiado da digitalização de imagens adquiridas a partir de modalidades de exame baseadas em filme (como por exemplo radiografia convencional), bem como pelo uso de imagens geradas originalmente em formato digital (provenientes de exames de Tomografia Computadorizada (*Computed Tomography* – CT) e Ressonância Magnética (*Magnetic Resonance Imaging* – MRI), por exemplo). Independentemente de sua origem, imagens digitais têm contribuído para melhorias no processo de avaliação estrutural e funcional necessário ao diagnóstico médico. Sua adoção é refletida no aumento do número de possíveis cenários de aplicação e do volume de exames realizados, gerando demandas relacionadas ao manuseio e ao gerenciamento dos dados digitais adquiridos pelas instituições de saúde (BOURNE, 2010).

Do ponto de vista organizacional, a adoção e o uso diário de dados médicos em formato digital (incluindo imagens) demandam uma logística para aquisição, distribuição e armazenamento de conteúdo. A infraestrutura necessária ao suporte desse cenário é composta por uma agregação de componentes de *hardware* e *software*, comumente organizados como HIS (*Hospital Information System*), RIS (*Radiology Information System*) e PACS (*Picture Archiving and Communication System*) (ver GLOSSÁRIO DE TERMOS E EXPRESSÕES) (SMITH, 2006; HECKMAN; SCHULTZ, 2006). É comum que instituições de saúde adotem esses sistemas de forma integrada, distribuindo os dados disponíveis de acordo com as características de cada um. HIS e RIS são usados para a organização de dados convencionais (alfanuméricos, como por exemplo dados demográficos de pacientes, prescrições de medicamentos, laudos de exames), enquanto PACS é usado para a organização de dados não convencionais (binários, como por exemplo imagens digitais, áudio de laudos, vídeos de determinadas modalidades de exame), garantindo que os mesmos sejam adequadamente adquiridos, distribuídos de forma segura e armazenados de forma confiável.

Quando usados de forma integrada em uma ou mais instituições de saúde, HIS, RIS e PACS baseiam-se em padrões de comunicação e representação de dados bem definidos (LI, 2011; EICHELBERG; ADEN; RIESMEIER, 2005), tais como HL7 (*Health Level Seven*) e DICOM (*Digital Imaging and Communications in*

Medicine) (ver GLOSSÁRIO DE TERMOS E EXPRESSÕES). Historicamente, o conjunto de padrões HL7 é utilizado na organização, troca de informações, integração, compartilhamento e recuperação de dados convencionais (HEALTH LEVEL SEVEN INTERNATIONAL, 2012; MEO; QUATTRONE; URSINO, 2011; DOLIN, 2001; DOLIN, 2006). O padrão DICOM, por sua vez, é comumente utilizado na aquisição, armazenamento, exibição, processamento, troca de informações, busca e impressão de dados não convencionais (DICOM, 2012; BIDGOOD JR, 1997; MILDENBERGER; EICHELBERG; MARTIN, 2002). Juntos, DICOM e HL7 contribuem para a definição e criação de fluxos de trabalho consonantes com as políticas das instituições que os utilizam.

Em um escopo limitado ao armazenamento de dados, o padrão DICOM define um conjunto de formatos de arquivo, perfis de aplicação e relacionamentos entre formatos de arquivo e mídias físicas (PS 3.10-2011, 2012; PS 3.11-2011, 2012; PS 3.12-2011, 2012). Essas definições não contemplam diretrizes ou *guidelines* para o armazenamento de dados convencionais e não convencionais, resultando em inúmeras implementações *ad hoc* com base em Sistemas Gerenciadores de Banco de Dados (SGBDs), sistemas de arquivos convencionais e sistemas de arquivos distribuídos. As camadas de armazenamento resultantes dessas implementações são utilizadas pelas arquiteturas PACS atuais como simples repositórios de dados, restritos a funções de gravação e leitura de conteúdo. Essa restrição desconsidera características incorporadas a SGBDs e *frameworks* para gerenciamento de dados (como por exemplo garantia de consistência, indexação e busca), essenciais no atendimento a objetivos como confiabilidade e flexibilidade de uso. Sistemas baseados nessas arquiteturas costumam disponibilizar essas características na forma de *features* derivadas de regras de negócio, distribuídas entre diferentes componentes. Essa distribuição contribui para o aumento da complexidade dos sistemas, negligenciando fatores relacionados a desempenho (importantes em cenários onde a interação humana é relevante, como por exemplo a visualização e interpretação de conjuntos de imagens).

Objetivando melhorar a organização e a execução de tarefas relacionadas ao armazenamento de dados médicos convencionais e não convencionais, o presente trabalho propõe uma nova arquitetura de alto desempenho para sistemas PACS centrada em extensões de SGBDs. A proposta estende a camada de armazenamento adotada nesses sistemas, transformando-a em um componente ativo da

arquitetura PACS conhecido como AE (*Application Entity*) (ver GLOSSÁRIO DE TERMOS E EXPRESSÕES) (PS 3.2-2011, 2012). Essa alteração é feita pela incorporação de módulos especializados a um SGBD, construídos em consonância com a extensão SQL/MED (*Structured Query Language / Management of External Data*) (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO), 2003) e seguindo APIs (*Application Programming Interfaces*) que possibilitam a integração entre extensões de *software* e o *core* de um SGBD. Os módulos derivados da proposta são projetados para atender a demandas de comunicação, armazenamento e processamento de forma integrada a um SGBD, utilizando-se de fatores como a redução do número de componentes da arquitetura, a escalabilidade visando persistência de dados com tecnologias heterogêneas e o processamento distribuído de consultas para ganho de desempenho.

1.1 CONTEXTUALIZAÇÃO

As arquiteturas PACS usadas atualmente são compostas por um conjunto de componentes de *hardware* e *software*, capazes de trocar entre si dados médicos convencionais e não convencionais seguindo as políticas de representação e comunicação estabelecidas pelo padrão DICOM. Esses componentes são organizados como entidades independentes, responsáveis pela implementação de regras de negócio definidas pelo fluxo de trabalho da instituição de saúde atendida; sua combinação permite construir sistemas de complexidade variada, mas com elementos básicos facilmente identificáveis (como pode ser visualizado na FIGURA 1) (HECKMAN; SCHULTZ, 2006):

1. o servidor HL7 atua como ponto de entrada para dados convencionais provenientes de fontes externas (tais como sistemas HIS e RIS), além de receber dados informados manualmente via interfaces customizadas. Os dados recebidos, correspondentes a informações coletadas previamente à execução dos exames, podem ser encaminhados diretamente aos equipamentos de aquisição de imagem ou ao servidor DICOM, na forma de DICOM *Modality Worklists* (MWLs) (ver GLOSSÁRIO DE TERMOS E EXPRESSÕES) (GALE; GALE, 2000);

2. os equipamentos de aquisição de imagem são responsáveis pela captura e gravação de imagens individuais ou conjuntos de imagens, de acordo com a especialidade médica atendida; as imagens adquiridas são complementadas com os dados provenientes das *worklists* enviadas pelo servidor HL7, com dados provenientes das *worklists* enviadas pelo servidor DICOM ou por dados informados manualmente via console;
3. visando a integração de diferentes AEs, o servidor DICOM assume o papel de elemento concentrador, enviando MWLs para os equipamentos de aquisição de imagem e recuperando as imagens geradas após a execução dos exames. Essas imagens são organizadas de acordo com a estrutura de atributos e arquivos definida pelo padrão DICOM;
4. conjuntos de dados compostos pelos dados prévios dos exames e pelas imagens digitais resultantes são encaminhados pelo servidor DICOM para arquivamento em estruturas de *hardware* e *software* responsáveis pela organização, distribuição e armazenamento *online*, *near-line* e *offline* (ver GLOSSÁRIO DE TERMOS E EXPRESSÕES) (NAGY; SCHULTZ, 2006);
5. uma vez arquivados, os dados dos exames ficam disponíveis para visualização, avaliação e interpretação por diferentes profissionais de saúde. O trabalho executado por esses profissionais varia da simples exibição dos dados em *workstations* clínicas até a manipulação das imagens em *workstations* radiológicas (ver GLOSSÁRIO DE TERMOS E EXPRESSÕES), o que demanda recursos de *hardware* e *software* específicos para a execução de cada tarefa. A estrutura de armazenamento adotada pelas instituições de saúde determina a disponibilidade dos dados: quando armazenados de forma *online*, sua recuperação é imediata; quando armazenados de forma *near-line* ou *offline*, sua recuperação demanda um intervalo de tempo para busca, recuperação e disponibilização;
6. conjuntos de dados de exames são compostos por dados convencionais e por dados não convencionais, encapsulados individualmente em atributos e agrupados em imagens e arquivos de acordo com o padrão DICOM. É uma prática comum a arquiteturas PACS organizar dados alfanuméricos em bancos de dados relacionais ou objeto-relacionais,

usando diferentes níveis de normalização com o objetivo de reduzir sua redundância e, conseqüentemente, melhorar sua consistência.

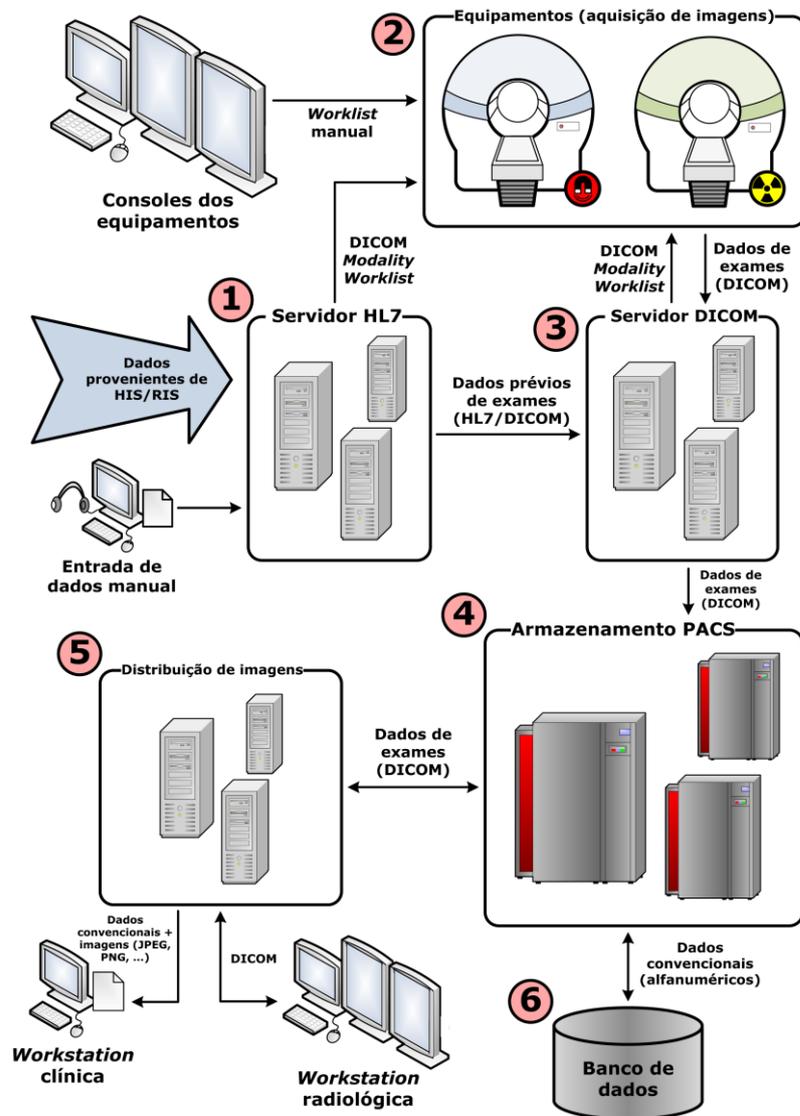


FIGURA 1 – EXEMPLO DE ARQUITETURA PACS (DESTAQUE PARA A ORGANIZAÇÃO MODULAR DAS ENTIDADES DE APLICAÇÃO). FONTE: (HECKMAN; SCHULTZ, 2006) (modificado pelo autor)

O cenário apresentado na FIGURA 1 exemplifica a complexidade do processo de gerenciamento de dados em instituições de saúde, e exhibe uma das inúmeras possíveis configurações que podem ser adotadas. Sistemas PACS permitem a utilização de AEs redundantes, ou mesmo a omissão de componentes específicos (como por exemplo o servidor HL7); considerando o armazenamento e o gerenciamento de dados, as escolhas feitas de maneira *ad hoc* durante o estágio de configuração dos serviços têm impacto direto nos resultados obtidos a médio e longo pra-

zos, no que se refere ao volume de dados armazenados e às estratégias de recuperação construídas sobre esses dados.

1.2 DESCRIÇÃO DO PROBLEMA

A literatura relacionada descreve PACS como um sistema de informações capaz de prover serviços de gerenciamento, armazenamento e distribuição de imagens digitais objetivando o diagnóstico médico. Apesar de correta, essa descrição limita o entendimento da complexidade inerente à adoção, instalação e uso de sistemas que se enquadram nessa categoria. É comum que arquiteturas PACS provoquem transformações radicais no escopo organizacional por meio de alterações em fatores econômicos, humanos e, não menos importantes, técnicos (PARÉ; TRUDEL, 2007).

Dentre os fatores técnicos impactantes à adoção de sistemas PACS, podem ser destacados a integração com sistemas HIS e RIS (para a recepção de dados adquiridos previamente e o retorno da situação dos exames executados), a indisponibilidade de atributos (*tags*) DICOM (ver GLOSSÁRIO DE TERMOS E EXPRESSÕES), como por exemplo a data e hora da execução dos exames ou da gravação das imagens adquiridas (derivada de implementações parciais do padrão) e a adoção de AEs com declarações de conformidade (*DICOM Conformance Statements*) (ver GLOSSÁRIO DE TERMOS E EXPRESSÕES) limitadas ou incorretas, o que reduz a possibilidade de implementação de serviços (PARR, 2001; HONEYMANBUCK, 2003; MANSOORI; ERHARD; SUNSHINE, 2012). Há também fatores relacionados à integração entre diferentes AEs, derivados da heterogeneidade de equipamentos instalados e da necessidade de distribuir serviços entre entidades de aplicação. A complexidade resultante do número e da distribuição das AEs em diferentes cenários de uso permite concluir que, quanto maior o número dessas entidades de aplicação em uma arquitetura PACS, maior é a propensão ao surgimento de problemas de interconectividade (COSTA; SILVA; OLIVEIRA, 2007; BELLON, 2011).

A redução no número de AEs em uma arquitetura PACS pode contribuir com a diminuição dos problemas de heterogeneidade e interconectividade entre seus componentes. Para tanto, é necessário que as entidades de aplicação remanescentes incorporem um maior número de serviços, sem comprometer seu desempe-

nho e sua disponibilidade. Uma possibilidade de incorporação de serviços diretamente relacionados a dados (como comunicação, armazenamento e processamento) está na utilização dos SGBDs relacionais ou objeto-relacionais, por meio de suas características de extensibilidade (BATORY, 1987). Atualmente, esses SGBDs (que compõem a camada de armazenamento do PACS) são utilizados como simples repositórios de dados convencionais (e, dependendo do cenário de uso, de dados não convencionais), assumindo um perfil passivo quando comparados a componentes presentes em outras camadas da arquitetura. O aprimoramento dessas entidades de aplicação por meio da criação de funções definidas pelo usuário (*User-Defined Functions* – UDFs) (GODFREY, 1998), bem como pela definição e implementação de extensões e estruturas diferenciadas para armazenamento e acesso a dados (CAREY; HAAS, 1990) permitem visualizar cenários que atendam às propostas de simplificação.

Nesse contexto, considerando a complexidade das arquiteturas PACS atuais e as características de extensibilidade disponíveis em SGBDs, o problema relacionado à presente tese pode ser expresso pela seguinte questão: *é possível definir uma arquitetura PACS simplificada e de alto desempenho com base em alterações na sua camada de armazenamento, centradas em extensões implementadas em nível de SGBD?*

1.3 HIPÓTESES E OBJETIVOS

A questão apresentada ao final da Seção 1.2 leva a um conjunto amplo de possibilidades, principalmente no que compete às possíveis extensões que permitiriam a incorporação de diferentes serviços pela camada de armazenamento de uma arquitetura PACS simplificada. De forma a restringir esse conjunto, três hipóteses relacionadas à comunicação, armazenamento e processamento de dados foram formuladas, com o objetivo de guiar a obtenção de uma resposta que valide a proposta do presente trabalho.

1.3.1 Hipótese #1

A construção de um esquema de BD (banco de dados) relacional aderente às características do padrão DICOM contribui para ganhos de desempenho em cenários onde a razão entre armazenamento e pesquisa/recuperação de conteúdo é de 1 para n ; essa razão é característica de sistemas PACS, onde uma imagem é adquirida e gravada uma vez, e consultada/recuperada sempre que necessário.

Objetivo específico relacionado: prover um esquema de BD relacional simplificado e de alto desempenho que seja indiferente às diversas combinações de *tags* encontradas em imagens no padrão DICOM, e que atenda a diferentes modalidades de exame e equipamentos sem modificações estruturais.

1.3.2 Hipótese #2

A modelagem e implementação de uma extensão de SGBD para a recepção e o envio de mensagens no padrão DICOM, apoiada sobre o esquema de BD relacional proposto na hipótese #1, permite agregar serviços de comunicação à camada de armazenamento de dados da arquitetura PACS.

Objetivo específico relacionado: estender a camada de armazenamento da arquitetura PACS com a incorporação de serviços de *parsing*, interpretação e composição de mensagens no padrão DICOM.

1.3.3 Hipótese #3

A adoção de tecnologias heterogêneas de armazenamento para dados convencionais e não convencionais permite explorar características de organização e persistência diversas, resultando em uma arquitetura PACS mais escalável, tolerante a falhas, com um maior desempenho e mais flexível quanto à forma como o conteúdo DICOM é armazenado (tanto em nível de *tag* quanto em nível de imagem).

Objetivo específico relacionado: promover a integração entre diferentes tecnologias de armazenamento, de forma a flexibilizar a distribuição de dados para as modalidades de armazenamento *online*, *near-line* e *offline*.

1.4 ORGANIZAÇÃO DO TRABALHO

Neste capítulo foram apresentadas informações relacionadas à aquisição e utilização de dados médicos convencionais e não convencionais, bem como sua distribuição em sistemas HIS, RIS e PACS. O contexto do trabalho (restrito a arquiteturas PACS) foi descrito, bem como o problema a ser abordado, as hipóteses a serem verificadas e os objetivos a serem atingidos.

O capítulo dois descreve o padrão DICOM (base para a comunicação e o armazenamento de dados em cenários envolvendo PACS) em termos estruturais e funcionais, além de fundamentar extensibilidade em SGBDs, a extensão SQL/MED e o modelo de armazenamento decomposto. O capítulo também apresenta fundamentos de computação de alto desempenho voltados à escalabilidade para armazenamento e processamento de dados distribuídos e tolerância a falhas.

O capítulo três organiza uma série de trabalhos correlatos na forma de uma revisão de literatura, com destaque para a utilização de bancos de dados em sistemas PACS – contexto a ser explorado no desenvolvimento do presente trabalho.

O capítulo quatro apresenta a proposta de arquitetura PACS simplificada e de alto desempenho, seguida dos materiais e métodos a serem utilizados em sua construção (incluindo os *softwares* empregados na prototipação das extensões de SGBD).

O capítulo cinco descreve os ambientes experimentais configurados para a execução dos testes necessários à verificação das hipóteses formuladas na Seção 1.3. Para cada ambiente, um conjunto de resultados envolvendo as operações de persistência, pesquisa de metadados e recuperação de imagens completas é apresentado e discutido, permitindo a comparação da arquitetura proposta com instâncias de uma arquitetura PACS reconhecida e bem estabelecida.

O capítulo seis conclui o trabalho, abordando cada hipótese e seu objetivo específico relacionado individualmente. As contribuições derivadas da tese são relacionadas na sequência, explicitando o que o trabalho trouxe de mais relevante à área de gerenciamento de imagens médicas. Como complemento, são apresentadas indicações de trabalhos futuros passíveis de serem desenvolvidos como adendos e melhorias à arquitetura PACS proposta.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta fundamentos do padrão DICOM referentes à sua organização estrutural e funcional, com destaque à representação de conteúdo e serviços de busca e recuperação. São apresentados também conceitos utilizados no desenvolvimento da proposta, incluindo fundamentos de extensibilidade em SGBDs, especificações sobre a extensão SQL/MED e regras de construção do modelo de dados decomposto – modelo esse que serve de base para a implementação do esquema de BD utilizado no presente trabalho. Fundamentos de computação de alto desempenho voltados à escalabilidade para armazenamento e processamento de dados e tolerância a falhas também são apresentados, servindo como base para, juntamente com a extensão SQL/MED, prover recursos teóricos para a distribuição de dados entre tecnologias heterogêneas.

2.1 ORGANIZAÇÃO DE CONTEÚDO NO PADRÃO DICOM

O padrão DICOM, desenvolvido originalmente pelo *American College of Radiology* (ACR) e pela *National Electrical Manufacturers Association* (NEMA), compreende um conjunto de especificações não-proprietárias para a estruturação e a formatação de imagens médicas digitais, bem como a definição de protocolos de comunicação utilizados na transmissão dessas imagens em cenários envolvendo uma ou várias instituições de saúde. Com sua primeira versão (ACR/NEMA 300) liberada em 1985, o padrão tem evoluído de acordo com as deliberações do DICOM *Standards Committee*, formado por fabricantes de equipamentos médicos, universidades e institutos de pesquisa (BIDGOOD JR, 1997; MILDENBERGER; EICHELBERG; MARTIN, 2002).

Dados armazenados em imagens DICOM são organizados por meio de *Information Object Definitions* (IODs) (ver GLOSSÁRIO DE TERMOS E EXPRESSÕES), que agrupam semanticamente conjuntos de *elementos de dados*. Cada IOD é associado a um ou mais DICOM *Message Service Elements* (DIMSEs) através de *Service-Object Pairs* (SOPs) (ver GLOSSÁRIO DE TERMOS E EXPRESSÕES), definindo assim quais operações poderão ser executadas com os dados em questão

(por exemplo, armazenamento, pesquisa e recuperação de imagens de tomografia computadorizada). A comunicação de dados e execução de serviços é negociada com base em um modelo ponto-a-ponto, envolvendo duas AEs que assumem, cada, um perfil que perdura durante toda a interação: *Service Class User* (SCU) ou *Service Class Provider* (SCP) (ver GLOSSÁRIO DE TERMOS E EXPRESSÕES) (PIANYKH, 2008).

Em seu nível organizacional mais baixo, *tags* DICOM identificadas por valores de *grupo* e *elemento* compõem os elementos de dados. Valores de grupo correlacionam elementos semanticamente, e valores de elemento identificam unicamente cada elemento dentro de um mesmo grupo. *Tags* são caracterizadas por *Value Representations* (VRs) e *Value Multiplicities* (VMs) (ver GLOSSÁRIO DE TERMOS E EXPRESSÕES), responsáveis por especificar tipos de dados, regras de formatação e número de ocorrências possíveis para o conteúdo de cada *tag* (PS 3.5-2011, 2012). O padrão DICOM define um dicionário de dados composto por um conjunto de *tags* padrão com valores reservados para grupos e elementos, permitindo a sua expansão pela inclusão de *tags* proprietárias (PS 3.6-2011, 2012).

Em seu nível organizacional mais alto, imagens DICOM são estruturadas como conjuntos de *tags*. O número de *tags* componentes de uma imagem varia de acordo com a disponibilidade de informações no momento do agendamento e da execução do exame, bem como da modalidade de exame a ser executada (como tomografia computadorizada ou ressonância magnética, por exemplo) e o fabricante do equipamento utilizado; assim, o conteúdo real de uma imagem só é conhecido em tempo de avaliação (*parsing*). Por fim, imagens são organizadas em uma hierarquia de quatro níveis, com cada nível sendo identificado unicamente pelo valor de uma *tag* específica. Essa hierarquia estabelece que um paciente (primeiro nível, *tag patientid*) pode realizar vários estudos, um estudo (segundo nível, *tag studyinstanceuid*) pode ser composto por várias séries, e uma série (terceiro nível, *tag seriesinstanceuid*) pode relacionar várias imagens. É na imagem (quarto nível, identificada unicamente pela *tag sopinstanceuid*) que os dados são efetivamente armazenados. A FIGURA 2 exemplifica, estruturalmente, a organização do conteúdo DICOM em nível de arquivo e *tag*.

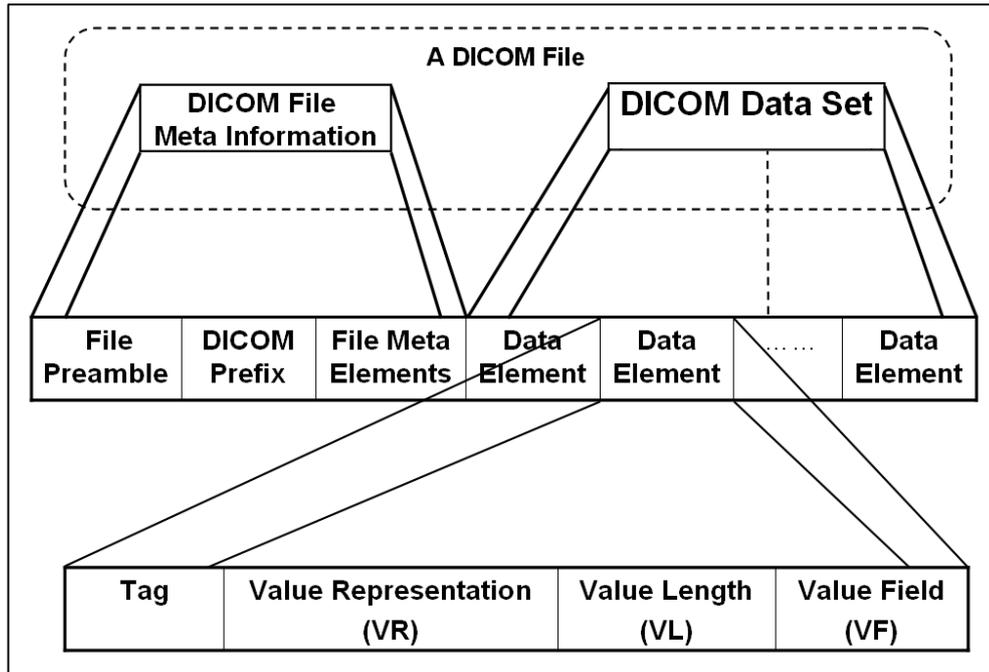


FIGURA 2 – EXEMPLO DE ORGANIZAÇÃO DE CONTEÚDO DICOM. EM SEU NÍVEL MAIS ALTO (ARQUIVO), UMA IMAGEM É COMPOSTA POR UM CONJUNTO DE TAGS. EM SEU NÍVEL MAIS BAIXO (TAG), UM ELEMENTO DE DADOS É COMPOSTO POR NÚMEROS DE GRUPO E ELEMENTO, ALÉM DE UM VALOR DE REPRESENTAÇÃO, MULTIPLICIDADE E CONTEÚDO.
 FONTE: (YU, 2008) (modificado pelo autor)

2.2 PESQUISA E RECUPERAÇÃO DE CONTEÚDO NO PADRÃO DICOM

O padrão DICOM especifica como as operações de pesquisa e recuperação de conteúdo devem ser executadas, de forma a garantir a interoperabilidade entre diferentes interpretações e implementações. Um nível mínimo de conformidade é esperado pela adoção de um *comportamento básico*, que pode ser complementado por um *comportamento estendido* a partir da disponibilização de um número maior de recursos (PS 3.4-2011, 2012).

Basicamente, as operações de pesquisa e/ou recuperação de conteúdo em DICOM seguem uma definição conhecida como *Entity-Relationship Model Definition*, construída sobre a hierarquia de quatro níveis que inclui paciente, estudo, série e imagem. Para cada um desses níveis, o padrão define um conjunto de atributos que podem ser usados como chaves de busca *únicas*, *obrigatórias* ou *opcionais*, de forma individual ou combinada. Cada atributo pode ser usado em comparações de igualdade simples, como parte de uma lista de valores, como parte de um intervalo de valores, como parte de uma sequência de valores e em comparações envolvendo

padrões (complementados ou não por *wildcards*), o que provê uma grande flexibilidade por parte das aplicações cliente na construção de predicados de pesquisa e recuperação.

Em sua versão atual, o padrão DICOM dá suporte a dois modelos de informação para pesquisa e recuperação de conteúdo (*Query/Retrieve Information Models*): *Patient Root* e *Study Root*. Cada modelo identifica o nível da hierarquia a partir do qual a pesquisa/recuperação é iniciada. Quatro valores de nível (*Query/Retrieve Level Values*) limitam a busca: *Patient*, *Study*, *Series* e *Image*. Juntos, um *Information Model* e um *Level Value* definem um intervalo entre níveis hierárquicos a ser percorrido, do nível superior ao nível inferior.

Para atender a esses critérios de busca, o comportamento básico estabelece que AEs responsáveis por responder a pesquisas devem ser capazes de processar métodos de busca hierárquica (*Hierarchical Search Methods*). Esses métodos percorrem recursivamente a hierarquia (limitados pelos níveis superior e inferior), executando as seguintes etapas:

1. a busca inicia pelo nível hierárquico definido pelo *Query/Retrieve Information Model*, descendo pela hierarquia até o nível indicado pelo *Query/Retrieve Level Value* (também conhecido como nível-alvo);
2. quando o nível-alvo é atingido, atributos identificados como chaves de busca são usados para comparações com os valores fornecidos pela aplicação cliente;
3. entidades (atributos ou imagens) cujos dados satisfaçam os critérios de busca são retornadas à aplicação cliente, uma de cada vez, juntamente com os atributos identificadores do nível-alvo e de seus níveis hierárquicos superiores.

O comportamento estendido, por sua vez, define consultas relacionais (*Relational-Queries*), construções que permitem comparações de valores de atributos e recuperação de entidades em múltiplos níveis hierárquicos, como segue:

1. a busca inicia pelo nível hierárquico definido pelo *Query/Retrieve Information Model*, descendo pela hierarquia até o nível indicado pelo *Query/Retrieve Level Value* (também conhecido como nível-alvo);
2. para cada nível hierárquico, atributos identificados como chave de busca são usados para comparações com os valores fornecidos pela aplicação cliente. Entidades cujos dados satisfaçam os critérios de busca são levadas para o próximo nível da hierarquia;

3. quando o nível-alvo é atingido, entidades cujos dados foram coletados em todos os níveis percorridos são retornadas à aplicação cliente, uma de cada vez.

Apesar da descrição algorítmica presente na especificação do padrão DICOM, cada implementação é responsável por decisões particulares relacionadas aos detalhes da tecnologia de persistência empregada. Em SGBDs relacionais, por exemplo, *Hierarchical Search Methods* e *Relational-Queries* são traduzidas para uma ou mais consultas em SQL, cuja complexidade é derivada direta do esquema de BD adotado.

2.3 EXTENSIBILIDADE EM SGBDs

Os fundamentos de extensibilidade em SGBDs são datados da década de 80, derivados da avaliação de que um armazenamento puramente baseado em tuplas e relações não seria suficiente para atender a demandas envolvendo elementos de dados complexos (HASKIN; LORIE, 1982). Na época, pesquisas e experimentos identificaram a possibilidade de incorporar aos SGBDs técnicas já ratificadas pela engenharia de *software* no desenvolvimento de aplicações (como por exemplo o modelo em camadas). Sistemas Gerenciadores de Banco de Dados passaram a ser organizados como um conjunto de camadas interconectadas por interfaces simplificadas; essa estratégia permitia que camadas específicas (mapeadas fisicamente para diferentes tipos de dados, algoritmos de indexação, *engines* de armazenamento de dados, entre outras) pudessem ser substituídas, em tempo de compilação, por camadas mais adequadas ao contexto de uso (BATORY, 1987; BATORY, 1988).

O aprimoramento do processo de incorporação de diferentes camadas à estrutura base dos SGBDs extensíveis permitiu o mapeamento de outras características desses softwares ao mesmo modelo. Como resultado, surgiu o conceito de tipo de dado customizado à aplicação (mais conhecido como tipo de dado abstrato – *Abstract Data Type* – ADT), permitindo estender o catálogo de tipos de dados padrão (números inteiros, caracteres, entre outros) pela definição e implementação de tipos contextualizados, orientados a aplicações-alvo cujos dados possuíam características específicas (como por exemplo imagens e dados geométricos e espaciais)

(GARDARIN, 1989). Inicialmente compilados como camadas de acoplamento variado (CAREY, 1986; SCHWARZ, 1986), e posteriormente definidos como composições envolvendo códigos compilados para execução de operações e especificações armazenadas no dicionário de dados dos SGBDs (STONEBRAKER, 1986), esses novos tipos de dados promoveram o processo de modelagem e construção de instâncias de bancos de dados a um nível mais alto, simplificando a representação e o mapeamento de entidades presentes no domínio de aplicação para entidades no domínio de persistência.

Como opção/complemento à extensibilidade via ADTs, as operações definidas no contexto dos tipos customizados passaram a ser implementadas de forma isolada. Surgiram assim as UDFs, procedimentos escritos e compilados em diferentes linguagens de programação e “ligados” à estrutura base do SGBD por meio de uma interface bem definida. Essa interface provê meios para *marshalling* e *unmarshalling* de argumentos e valores de retorno, suportada por descrições definidas em dicionário de dados que permitem, em tempo de execução, verificar a correteza sintática de chamadas aos procedimentos a partir de instruções DML (*Data Manipulation Language*) (LINNEMANN, 1988; ORDONEZ, 2010).

2.4 EXTENSÃO SQL/MED

Definida oficialmente pela ISO (*International Organization for Standardization*) em meados de 2003, a extensão SQL/MED passou a integrar o padrão SQL com o objetivo de estabelecer uma metodologia de acesso a fontes de dados externas às instâncias relacionais. Através da sua utilização, é possível complementar os dados normalizados disponíveis em instâncias de bancos de dados relacionais com dados provenientes de origens diversas, sem que o SGBD assumira a responsabilidade pela gerência e manutenção desses dados. Essa integração entre fontes de dados heterogêneas visa simplificar o acesso a diferentes conjuntos estruturalmente distintos, provendo uma interface unificada (baseada na linguagem SQL) que permita a seleção, a manipulação e o estabelecimento de relações diretas entre os conjuntos disponíveis (MELTON, 2001; MELTON, 2002).

O uso da extensão SQL/MED está condicionado à escolha, feita em tempo de modelagem e implementação de um esquema relacional físico, de como acessar os dados externos à instância de BD. Conceitualmente, duas opções são disponibilizadas: *wrapper interfaces* e *datalinks*. Ambas são definidas como segue, e podem ser visualizadas na FIGURA 3:

- *wrapper interfaces* baseiam-se em três componentes, conectados entre si por APIs bem definidas. O acesso físico aos dados externos à instância relacional é feito por *Foreign Servers* (FSs), responsáveis pela manutenção dos dados acessados em seu formato original; esses componentes são utilizados por um ou mais *Foreign Data Wrappers* (FDWs), que estabelecem uma conexão lógica entre representações abstratas de tabelas (*Foreign Tables – FTs*) e as representações físicas correspondentes. Instruções SQL envolvendo FTs são encaminhadas aos respectivos FDWs, que por sua vez as encaminham para execução pelos FSs;
- *datalinks* são instâncias de um tipo de dado específico, criado com o intuito de referenciar arquivos armazenados em sistemas de arquivos externos ao SGBD. Diferentemente de um *path* convencional, instâncias do tipo *datalink* permitem controle sobre uma possível integridade referencial aplicada ao campo que as armazenam, bem como o acesso ao conteúdo referenciado sem a necessidade de o mesmo ser importado fisicamente na instância de BD.

Mesmo antes de sua normatização pela ISO e posterior inclusão ao padrão SQL, a extensão SQL/MED foi utilizada como suporte à implementação de uma série de modelos de armazenamento, além de servir como base teórica para a concepção de soluções envolvendo a consolidação de dados heterogêneos. Destacaram-se na época trabalhos voltados ao gerenciamento de grandes volumes de dados de simulação (PAPIANI, 1999; PAPIANI; WASON; NICOLE, 2000) e a modularização de repositórios de dados heterogêneos baseados em SGBDs e sistemas de arquivos (SURJANTO; RITTER; LOESER, 2000; MITTAL; HSIAO, 2001; BHATTACHARYA, 2002). Atualmente, a extensão é utilizada como base teórica para a construção de modelos de armazenamento que apostam na simbiose entre SGBDs e repositórios de dados não-relacionais (IVANOVA; KERSTEN; MANEGOLD, 2012), bem como na integração entre SQL e NoSQL (LAWRENCE, 2014).

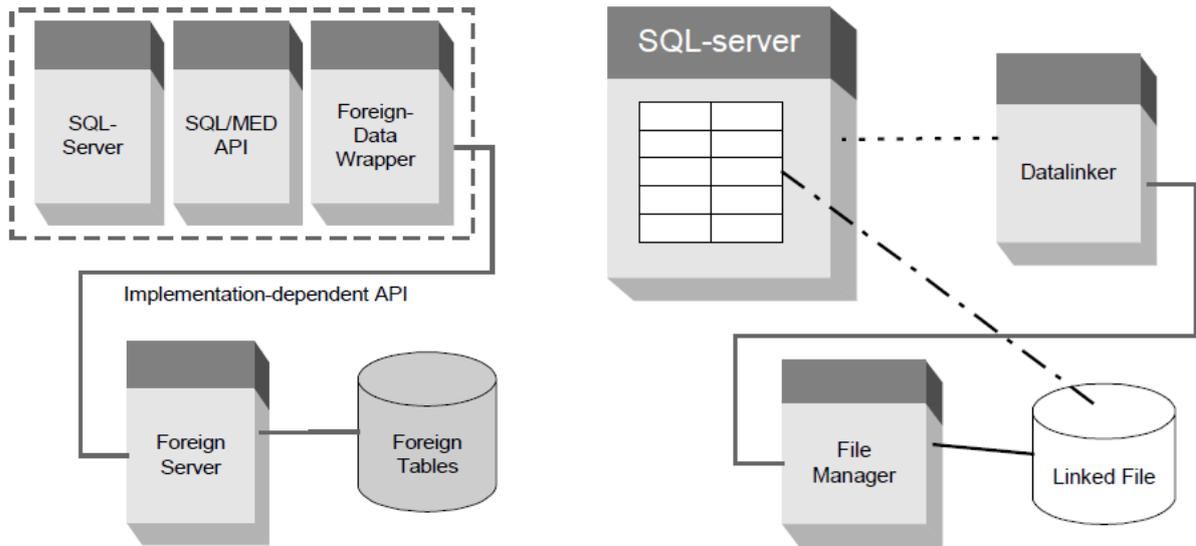


FIGURA 3 – EXTENSÃO SQL/MED, DE ACORDO COM SEU PROJETO CONCEITUAL. À ESQUERDA, COMPONENTES DA *WRAPPER INTERFACE*. À DIREITA, O TIPO DE DADO *DATALINK*. FONTE: (MELTON, 2001)

As definições da extensão SQL/MED, tal qual apresentado na FIGURA 3, são interpretadas de diferentes maneiras no momento de sua implementação e integração a um SGBD. Há implementações bastante aderentes à modularização definida conceitualmente, e também implementações que a simplificam na forma de uma extensão, integrada ao ciclo de vida dos processos do SGBD. Independentemente do tipo de implementação adotada, o comportamento primário se mantém: acessar dados externos à instância de BD, traduzindo-os para uma representação relacional utilizável pelo SGBD.

2.5 DECOMPOSED STORAGE MODEL

O modelo de armazenamento decomposto (*Decomposed Storage Model – DSM*) (COPELAND; KHOSHAFIAN, 1985) é um modelo de armazenamento baseado na decomposição das relações de um esquema conceitual em um conjunto de relações binárias simples. Para cada atributo originalmente definido no esquema conceitual, o modelo propõe a criação de uma relação binária composta por uma chave primária substituta (*surrogate key*) e o atributo, ordenado pelo valor da chave substituta (além de uma outra relação binária com a mesma estrutura, mas orde-

nada pelo valor do atributo). Um exemplo de decomposição de uma relação (originalmente em sua forma horizontal) para DSM pode ser visualizada na FIGURA 4. Nele, uma relação (chamada originalmente de R) é composta por uma chave substituta (*surrogate key* – *sur*) e por três atributos de tipos de dados e domínios quaisquer ($a1$, $a2$ e $a3$). A aplicação das regras de decomposição resulta na criação de três relações ($RD1$, $RD2$ e $RD3$), cada qual herdando a chave substituta da relação original e um dos atributos.

R	sur	a1	a2	a3
	s1	v11	v21	v31
	s2	v12	v22	v32
	s3	v13	v23	v33

RD1	sur	a1
	s1	v11
	s2	v12
	s3	v13

RD2	sur	a2
	s1	v21
	s2	v22
	s3	v23

RD3	sur	a3
	s1	v31
	s2	v32
	s3	v33

FIGURA 4 – EXEMPLO DE DECOMPOSIÇÃO DE UMA RELAÇÃO HORIZONTAL EM UM CONJUNTO DE RELAÇÕES BINÁRIAS SIMPLES. FONTE: (COPELAND; KHOSHAFIAN, 1985) (modificado pelo autor)

Esse modelo pode ser considerado como o predecessor das tecnologias orientadas a colunas, usadas como base para a implementação de diversas variantes NoSQL (ABADI, 2007). Sua organização permite otimizar o acesso pontual a atributos, além de reduzir/eliminar a esparsidade no armazenamento de dados. Como cada atributo é armazenado em uma relação em separado, com uma estrutura mais simples, o tempo de recuperação é consideravelmente menor quando comparado ao tempo gasto no acesso a dados de relações mais complexas (com um maior número de atributos por tupla). Seu desempenho, porém, é comprometido quando muitos atributos são manipulados simultaneamente em operações de inserção, atualização, exclusão e junção. Aqui, todo atributo é tratado em separado, demandando uma operação de escrita ou leitura individual; em esquemas de BD ditos horizontais, o número dessas operações é reduzido pelo fato de os atributos estarem em uma mesma relação.

2.6 COMPUTAÇÃO DE ALTO DESEMPENHO

A computação de alto desempenho (*High-Performance Computing* – HPC) é caracterizada pelo uso de recursos de *hardware* em conjunto a técnicas de processamento paralelo e distribuído, com o objetivo de atender a demandas que envolvam o uso de complexas infraestruturas computacionais para melhoria de desempenho (DANTAS, 2005). Historicamente, essas demandas foram atendidas pela adoção de supercomputadores (tais como o *Cray 1*), sistemas baseados em processadores vetoriais (extensivamente usados como plataforma de execução para aplicações científicas), multiprocessadores simétricos (*Symmetric Multiprocessors* – SMPs, caracterizados pelo compartilhamento total dos recursos computacionais disponíveis) e processadores massivamente paralelos (*Massively Parallel Processors* – MPPs, com seu fraco acoplamento e conexão internodos de alta velocidade), juntamente com modelos de programação para uso de memória compartilhada e baseados em passagem de mensagens (BELL; GRAY, 2002; DONGARRA, 2006). Atualmente, essas arquiteturas têm sido abstraídas pelo uso dos termos *cluster*, *grid* e *cloud*, cuja definição (por vezes controversa) (DONGARRA, 2005) pode ser resumida como segue (SADASHIV; KUMAR, 2011, PRATHIBHA; LATHA; SUMATHI, 2014):

- *Cluster*: uma coleção de computadores interconectados por redes de alta velocidade, organizados em um ambiente homogêneo administrado de forma centralizada, e vistos externamente como uma unidade capaz de suportar cargas de trabalho voltadas a processamento ou gerenciamento de dados;
- *Grid*: uma organização que possibilita o compartilhamento, a seleção e a agregação de recursos computacionais geograficamente dispersos, em tempo de execução e sob demanda, para atender a uma carga de trabalho de duração variada. Diferentemente de *clusters*, ambientes de *grid* são administrados de forma descentralizada, caracterizam-se pela heterogeneidade e configuram-se de forma dinâmica, incluindo e liberando recursos de acordo com a tarefa a ser executada;
- *Cloud*: um tipo de sistema paralelo/distribuído composto por recursos computacionais virtualizados interconectados, oferecendo infraestrutura (IaaS – *Infrastructure as a Service*), plataforma para instalação e configuração (PaaS – *Platform as a Service*) e software (SaaS – *Software as a Service*) sob demanda, com provisionamento feito de acordo com a carga de trabalho a ser

atendida. Essa organização demanda um mínimo de esforço de gerenciamento por parte do usuário, sendo de responsabilidade do fornecedor do serviço o provimento dos recursos previamente contratados.

Apesar do entendimento comum de que soluções baseadas em HPC têm foco em desempenho de processamento, outros benefícios decorrentes do uso de *clusters*, *grids* e *clouds* têm sido explorados extensivamente; dentre esses benefícios, destacam-se a escalabilidade dessas organizações de recursos computacionais para o gerenciamento distribuído de dados e a disponibilização de estratégias para a implementação de tolerância a falhas.

2.6.1 Escalabilidade

O conceito de escalabilidade define a capacidade ou habilidade de um sistema, rede ou processo de incorporar um número crescente de componentes ou objetos, com o objetivo de prover recursos para atender a um incremento na demanda da sua carga de trabalho (BONDI, 2000). Esse conceito generaliza um atributo que, de acordo com a literatura, pode ser tipificado e relacionado com características estruturais (envolvendo estruturas de dados e algoritmos) e espaciais (envolvendo consumo de memória), pode ser caracterizado, analisado e entendido de acordo com o contexto em que é avaliado (DUBOC; ROSENBLUM; WICKS, 2006), e que já teve sua validade questionada pelo uso indiscriminado e pela falta de uma definição mais específica (HILL, 1990). Controvérsias à parte, a escalabilidade como característica é relevante em diferentes áreas, sendo um diferencial desejável (se não obrigatório) em sistemas cuja carga de trabalho é variável a curto, médio e longo prazos.

Em se tratando de gerenciamento distribuído de dados, escalabilidade pode ser aplicada em diferentes níveis. O mais baixo (sobre o qual apoiam-se os demais) envolve a organização estrutural dos componentes físicos utilizados na distribuição, bem exemplificada por cenários englobando redes de alto desempenho (GROSSMAN, 2009), pela adoção de componentes diversificados na construção das chamadas hierarquias de armazenamento (WATSON, 2005), pelo uso de componentes reconfiguráveis nos nodos de armazenamento, permitindo a execução local de rotinas de *software* para otimizar o uso de recursos de processamento (ZHANG; FENG, 2008) e pela criação de camadas de armazenamento intermediárias entre aplicação e *storage*, compondo uma analogia a *caches* de alto desempenho e capa-

cidade (ABBASI, 2009). Em um nível superior, são configurados e disponibilizados serviços para a otimização do armazenamento propriamente dito, tais como versões melhoradas de sistemas de arquivos incluindo recursos de paralelismo (HILDEBRAND; HONEYMAN, 2005) e protocolos para a transferência e replicação de conteúdo, construídos especificamente para execução em ambientes distribuídos de alto desempenho (ALLCOCK, 2002). Para o nível de aplicação, esses serviços são abstraídos por *frameworks* ou APIs permitindo o acesso e o processamento paralelo de grandes volumes de dados via modelos de programação como MapReduce (FADIKA, 2014, HUMBETOV, 2012), o particionamento e a distribuição de dados para SGBDs distribuídos (HABABEH; RAMACHANDRAN; BOWRING, 2007) e a adoção efetiva de ferramentas capazes de suportar o armazenamento e a indexação distribuída de dados estruturados e não-estruturados (GAO; NACHANKAR; QIU, 2011).

2.6.2 Tolerância a falhas

Um sistema é dito tolerante a falhas quando a falha de um de seus componentes (ou de um subsistema) não compromete o funcionamento do sistema como um todo. Esse conceito (e seus correlatos, como por exemplo confiabilidade, disponibilidade e manutenibilidade) é extensivamente estudado na literatura que trata de sistemas eletroeletrônicos, sendo adaptado ao contexto de *hardware* e *software* para indicar a robustez de um sistema quando da ocorrência de erros físicos e/ou lógicos (LAPRIE, 1985). Independentemente da área de aplicação, a tolerância a falhas é tida como um fator diferencial que atesta a qualidade de sistemas computacionais, e que é conquistada (total ou parcialmente) pelo atendimento a quatro princípios básicos (WHITE; MILES, 1996):

- *redundância*: implica na existência de componentes passíveis de serem utilizados como substitutos a componentes que sofreram falhas, de forma a manter a disponibilidade do sistema;
- *isolamento de falhas*: falhas ocorridas em um componente ou módulo não podem se propagar para outros componentes ou módulos, gerando um efeito cascata que pode levar o sistema como um todo ao colapso. Esse princípio demanda decisões de projeto especificamente voltadas à contenção;

- *detecção e aviso de falhas*: falhas devem ser detectadas no momento de sua ocorrência, e avisos devem ser emitidos de forma a permitir a sua correção imediata. Esse princípio é válido também para as chamadas *falhas latentes* (falhas que, não necessariamente, comprometem o funcionamento do sistema);
- *correção de falhas*: implica no conjunto de ações necessário à correção da falha detectada. Comumente, é executado a partir dos avisos emitidos pelos procedimentos de detecção, utilizando os componentes redundantes disponíveis.

Em HPC, dadas as dimensões atuais de *clusters*, *grids* e *clouds*, a ocorrência de falhas tem sido tratada como um evento “normal” (deixando de ser uma exceção e passando a ser uma possibilidade sempre presente) (CAPPELLO, 2009). Nessas condições, abordagens baseadas apenas na redundância de componentes (que podem ser classificadas como reativas) têm sido complementadas por estratégias baseadas em protocolos e algoritmos proativos, importantes na detecção das falhas e na tomada de medidas paliativas imediatas. Exemplos de estratégias que extrapolam a redundância de componentes empregam a replicação de conjuntos de dados usados em processamentos distribuídos, permitindo que nodos íntegros possam assumir o processamento de nodos defeituosos sem que seja necessária a migração de *datasets* inteiros (BOSILCA, 2009). Abordagens mistas também são possíveis, utilizando algoritmos para detecção de falhas em conjunto com a migração proativa de tarefas em execução e o uso de *checkpoints* (resultados parciais persistidos periodicamente) para a recuperação de falhas não previstas ou não identificadas antes de sua ocorrência (LAN; LI, 2008). Para os casos em que houver redundância de recursos físicos, com os mesmos em *standby*, é possível reagendar tarefas sob demanda contando com a incorporação desses recursos ao *cluster/grid* (LI, 2009), bem como a utilização dos mesmos para hospedar máquinas virtuais instanciadas sob demanda (FU; XU, 2009). Há, ainda, a possibilidade da adoção de bibliotecas de comunicação que incorporem prevenção e tratamento de falhas, abstraíndo-as das aplicações cliente (FAGG, 2005).

Os fundamentos apresentados neste capítulo servem de base teórica para a modelagem e implementação da proposta de arquitetura PACS simplificada e de alto desempenho, e serão empregados na construção dos protótipos utilizados para ex-

perimentação. No próximo capítulo, é feita uma revisão de literatura com destaque ao uso de SGBDs em sistemas PACS – cenário explorado e estendido no presente trabalho.

3 REVISÃO DE LITERATURA

Neste capítulo é feita uma revisão de literatura que visa identificar e relacionar trabalhos representativos no contexto de armazenamento de dados em arquiteturas PACS. Os trabalhos relacionados são agrupados em duas categorias: armazenamento baseado em estratégias diversas (que não SGBDs), e armazenamento baseado em SGBDs; é dada ênfase à utilização de sistemas gerenciadores de banco de dados, pelo fato de essa tecnologia suportar extensibilidade de forma nativa – recurso sobre o qual a proposta do presente trabalho é baseada.

3.1 ARMAZENAMENTO DE IMAGENS MÉDICAS NO PADRÃO DICOM EM ARQUITETURAS PACS – ESTRATÉGIAS DIVERSAS

Uma das estratégias mais simples adotada para armazenamento de conteúdo DICOM em sistemas PACS baseia-se na utilização de sistemas de arquivos convencionais, disponibilizados em *hardware* também convencional. Independentemente do tipo de sistema utilizado, imagens DICOM armazenadas como arquivos são distribuídas em árvores de diretório que representam, fisicamente, a organização hierárquica do padrão em seus quatro níveis: *paciente*, *estudo*, *série* e *imagem* (YAKAMI, 2011) (vide FIGURA 5). É uma abordagem de baixo custo e de rápida instalação, facilitando a expansão do espaço de armazenamento quando necessário e permitindo a composição de diferentes visões sobre a estrutura física através do uso de atalhos para diretórios e arquivos; porém, é extremamente limitante no que diz respeito à recuperação de imagens baseada em conteúdo (mesmo que alfanumérico). Como sistemas de arquivos convencionais não fornecem de forma nativa estruturas de indexação para o conteúdo armazenado, a localização e posterior recuperação de imagens a partir de um de seus elementos componentes (como por exemplo o nome do paciente) só é possível a partir do *parsing* feito sobre os arquivos – operação inviável mesmo quando executada em volumes de dados de tamanho reduzido.

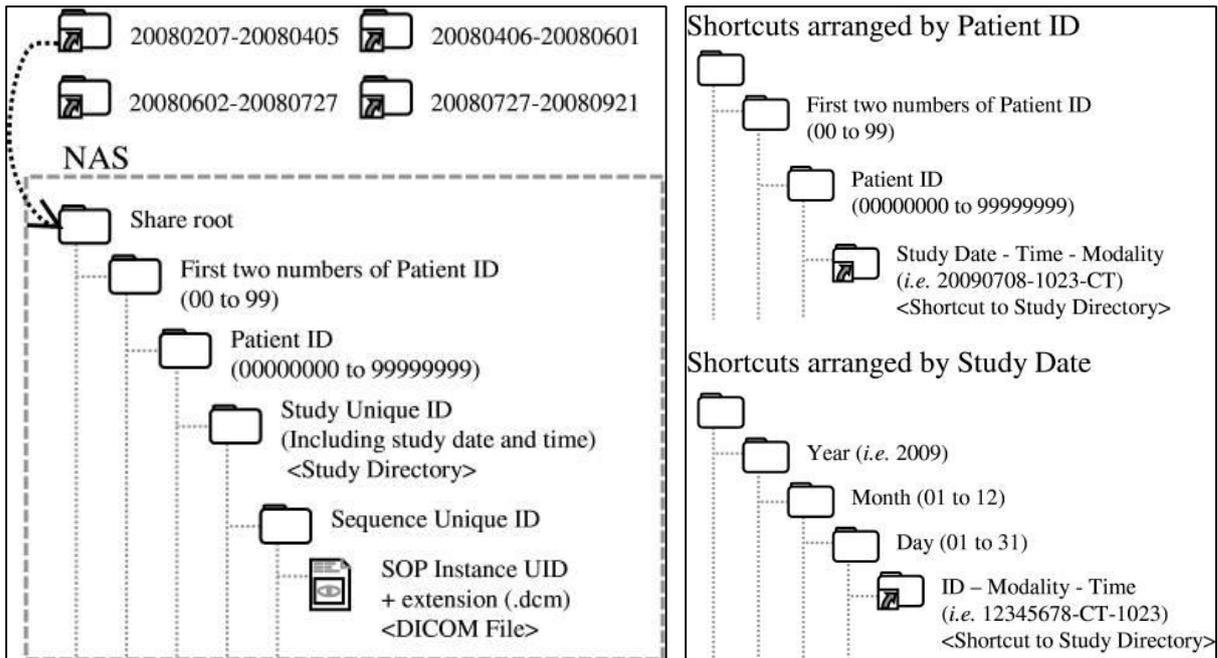


FIGURA 5 – EXEMPLO DE ARMAZENAMENTO DE CONTEÚDO DICOM EM SISTEMAS DE ARQUIVOS CONVENCIONAIS. À ESQUERDA: ORGANIZAÇÃO DE UMA ÁRVORE DE DIRETÓRIOS EM UM NAS (*NETWORK-ATTACHED STORAGE*). À DIREITA: VISÕES DIFERENCIADAS CONSTRUÍDAS SOBRE UMA MESMA ESTRUTURA FÍSICA. FONTE: (YAKAMI, 2011)

O armazenamento e busca de conteúdo DICOM em sistemas de arquivos convencionais pode ser melhorado através da adoção de técnicas combinadas, baseadas em modelos de dados aprimorados (como por exemplo HDF (*Hierarchical Data Format*) e NetCDF (*Network Common Data Format*)), usadas em conjunto com sistemas de arquivos distribuídos/paralelos (como por exemplo PVFS – *Parallel Virtual File System*). Essas técnicas são naturalmente voltadas a ambientes de *cluster* e/ou *Grid*, o que as direciona a cenários que demandam escalabilidade e onde a distribuição e o particionamento de dados são fatores relevantes (MACEDO, 2009; MAGNUS, 2012; SOARES, 2012). *Dividir e conquistar* passa a ser uma estratégia aplicável tanto para armazenamento quanto para recuperação de conteúdo, com a carga de trabalho sendo distribuída e, em teoria, minimizando o tempo de resposta total. A falta de ferramenta para indexação, no entanto, continua sendo um fator limitador. Um exemplo de combinação entre modelo de dados e sistema de arquivos pode ser visualizado na FIGURA 6.

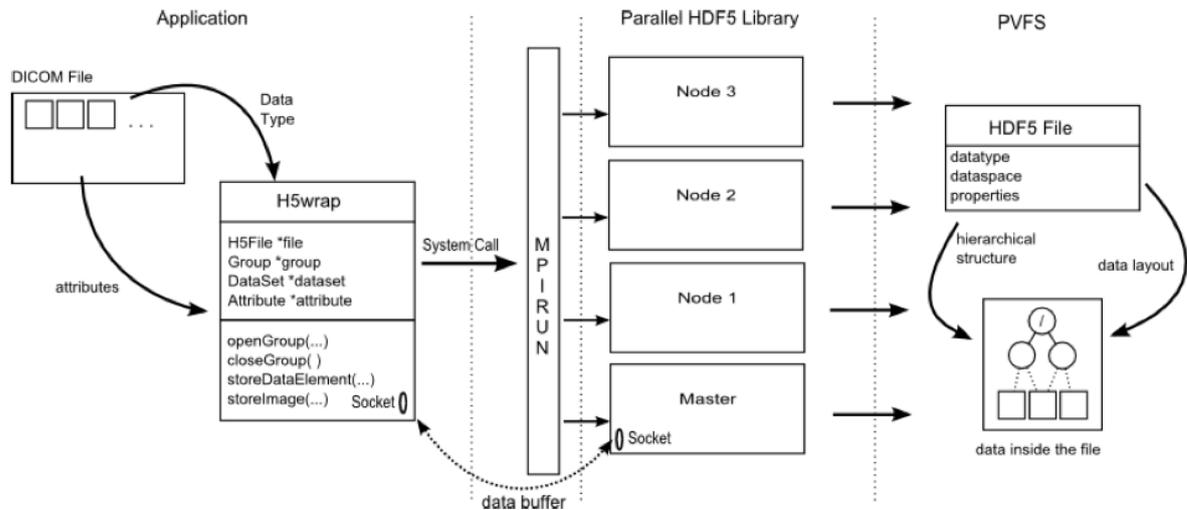


FIGURA 6 – EXEMPLO DE ARMAZENAMENTO DE CONTEÚDO DICOM EM UMA ESTRUTURA COMBINADA DE MODELO DE DADOS HDF E SISTEMA DE ARQUIVOS PVFS. FONTE: (SOARES, 2012)

A ausência de índices para metadados em sistemas de arquivos pode ser suprida pela adoção de estratégias de baixo nível, como o uso de atributos estendidos. Quando disponíveis, esses atributos são definidos e armazenados em nível de arquivo na forma de pares chave/valor, complementando os atributos padrão (como por exemplo as permissões de escrita e leitura) e sendo acessados via chamadas de sistema (CORRIERO, 2011). Estratégias de mais alto nível, por sua vez, utilizam catálogos construídos a partir de metadados extraídos dos arquivos armazenados; esses catálogos são disponibilizados para acesso descentralizado, permitindo buscas sobre o conteúdo dos metadados indexados e um posterior acesso aos arquivos para recuperação do seu conteúdo completo (KOBBLITZ; SANTOS; POSE, 2008) (vide FIGURA 7). Ambas as alternativas podem ser utilizadas como complemento aos sistemas de arquivos convencionais e distribuídos; porém, sua construção é altamente dependente do conteúdo disponível no momento da definição dos atributos estendidos ou da criação dos catálogos. É comum que apenas uma parte dos metadados disponíveis nos arquivos e imagens originais seja utilizada na indexação, o que pode não atender a possíveis futuras demandas de pesquisa por conteúdo.

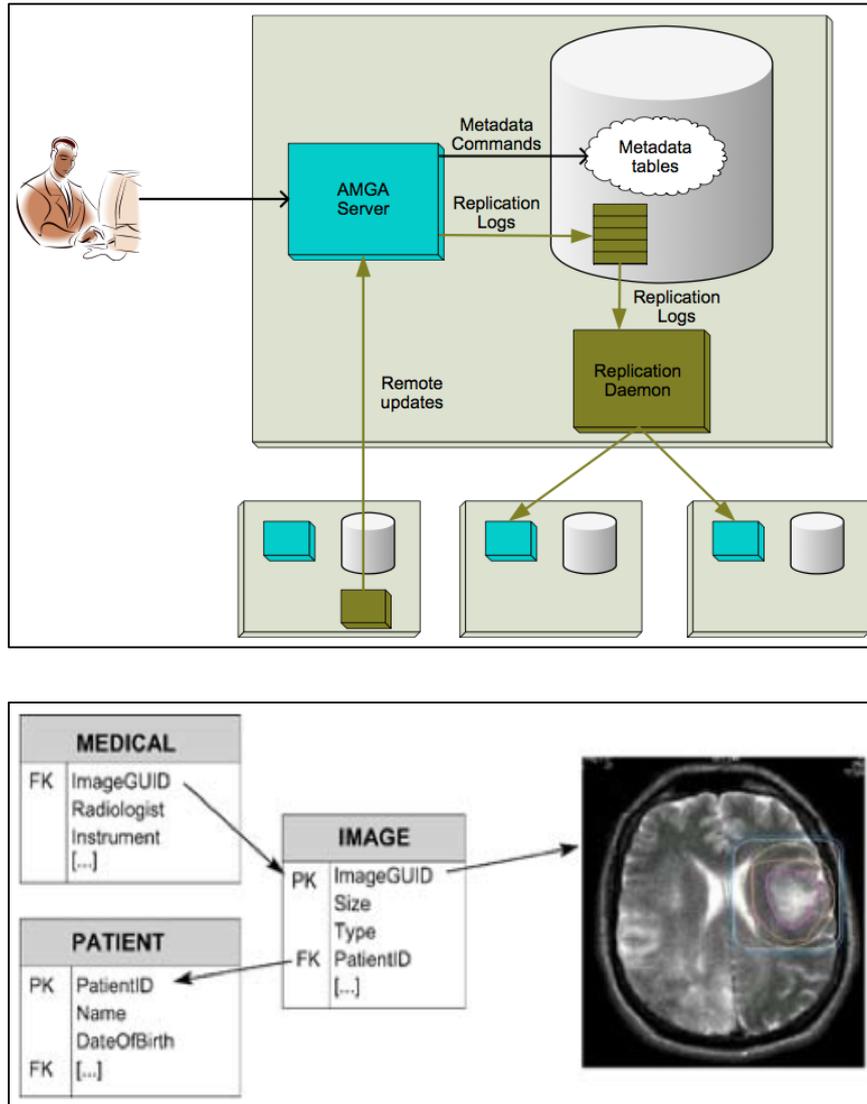


FIGURA 7 – EXEMPLO DE UM CATÁLOGO DE METADADOS EM UM AMBIENTE DISTRIBUÍDO. NA PARTE SUPERIOR: COMPONENTES DO CATÁLOGO E SUA DISTRIBUIÇÃO. NA PARTE INFERIOR: ESQUEMA RELACIONAL USADO COMO *BACK-END* PARA PERSISTÊNCIA DOS METADADOS. FONTE: (KOBLOITZ; SANTOS; POSE, 2008)

3.2 ARMAZENAMENTO DE IMAGENS MÉDICAS NO PADRÃO DICOM EM ARQUITETURAS PACS – USO DE SGBDs

Dada a relevância do uso de SGBDs em arquiteturas PACS para o presente trabalho, a revisão da literatura relacionada segue o modelo de revisão sistemática proposto por Barbara Kitchenham (KITCHENHAM, 2004), no qual são especificados os critérios para a contextualização e a definição das questões de pesquisa que justifiquem a execução da revisão, bem como a síntese das referências obtidas. Como

fontes de dados, foram escolhidas as seguintes bibliotecas digitais: *IEEE Xplore*¹, *SpringerLink*², *ACM Digital Library*³, *ScienceDirect*⁴, *Wiley*⁵ e *Web of Knowledge*⁶. Os termos de busca utilizados foram selecionados com base em acessos prévios às bibliotecas digitais, sendo agrupados de acordo com as questões de pesquisa. Esta revisão objetiva apresentar o estado da arte do processo de utilização de SGBDs no contexto de sistemas PACS, com foco no armazenamento de dados convencionais e não convencionais.

3.2.1 Contextualização

A inexistência de uma especificação para o armazenamento de imagens médicas leva à adoção de diferentes técnicas, escolhidas e utilizadas de maneira *ad hoc* por diferentes instituições de saúde ou mesmo grupos de pesquisa. Como tentativa de identificar uma tendência quanto ao procedimento utilizado no armazenamento dessas imagens, esta revisão sistemática objetiva relacionar estratégias e tecnologias para o armazenamento de imagens médicas no padrão DICOM em SGBDs, considerando a sua adoção em ambientes organizados por um sistema PACS.

3.2.2 Questões de pesquisa

A execução desta revisão sistemática visa organizar e relacionar um conjunto de referências, cuja análise permita responder às seguintes questões de pesquisa:

- quais são as estratégias que têm sido adotadas para o armazenamento de imagens no padrão DICOM em SGBDs no contexto de sistemas PACS?
- Que tecnologias têm sido utilizadas para viabilizar a adoção dessas estratégias?

¹ <http://ieeexplore.ieee.org/Xplore/guesthome.jsp>

² <http://www.springerlink.com/>

³ <http://dl.acm.org/>

⁴ <http://www.sciencedirect.com/>

⁵ <http://www.wiley.com/WileyCDA/>

⁶ <http://wokinfo.com/>

3.2.3 Definição e organização dos termos de busca

A definição dos termos de busca utilizados nesta revisão baseou-se em acessos prévios às bibliotecas digitais citadas no início da Seção 3.2. Uma vez escolhidos, os termos foram agrupados como segue:

- quanto aos objetos de interesse: "DICOM", "*Digital Imaging and Communications in Medicine*", "*Medical Imaging*", "*Medical Image*", "*Medical Images*", "*Clinical Imaging*", "*Clinical Image*", "*Clinical Images*";
- quanto às operações executadas: "Storage", "Archive", "Archiving";
- quanto às tecnologias de base utilizadas: "Database", "Databases", "DBMS", "RDBMS";
- quanto ao contexto de uso: "PACS", "*Picture Archiving and Communication System*".

Os termos de busca escolhidos foram combinados logicamente de forma a se adequarem às limitações impostas pelas ferramentas de busca oferecidas pelas bibliotecas digitais. As expressões de busca geradas pelas bibliotecas podem ser visualizadas no APÊNDICE 1.

3.2.4 Critérios de aceitação das referências

São consideradas referências relevantes para esta revisão sistemática artigos de congressos e periódicos e capítulos de livros, publicados na língua inglesa no período de janeiro de 2004 a agosto de 2011, que apresentem uma definição clara da abordagem e tecnologia utilizadas para o armazenamento de imagens médicas no padrão DICOM em SGBDs no contexto de PACS. Toda referência recuperada a partir das ferramentas de busca das bibliotecas digitais consultadas foi avaliada individualmente com base na leitura de seu título e resumo; os trabalhos direcionados ao contexto de armazenamento em SGBDs de imagens DICOM em sistemas PACS foram mantidos, e os demais descartados.

3.2.5 Procedimentos para avaliação das referências

As referências remanescentes do processo de leitura de títulos e resumos foram avaliadas com base em partes de seu conteúdo (parágrafos que concentram a

ocorrência dos termos de busca); aquelas que não apresentaram qualquer uma das quatro características a seguir foram mantidas, e as demais descartadas.

1. A referência apenas cita a tecnologia utilizada para o armazenamento de imagens médicas, sem fornecer maiores detalhes sobre a estruturação ou composição do esquema de BD adotado;
2. a referência não utiliza o padrão DICOM como forma de representação das imagens médicas;
3. a referência não caracteriza um cenário de uso de uma arquitetura PACS;
4. o foco principal da referência está em etapas do processo inerente à adoção de uma arquitetura PACS que não o armazenamento das imagens (como comunicação, por exemplo).

3.2.6 Estratégias para a extração de dados

As referências selecionadas para análise foram avaliadas individualmente, a partir da extração e classificação dos seguintes dados:

- estudo – dados da referência incluindo autor(es), título e local/data de publicação;
- estratégia(s) de armazenamento – SGBD relacional, SGBD orientado a objetos, SGBDs não convencional, entre outras;
- tecnologia(s) de armazenamento – MySQL, PostgreSQL, DB2, entre outras;
- cenário de aplicação para o qual a estratégia/tecnologia foi projetada;
- modalidade(s) contemplada(s) – modalidade(s) de exame que se beneficia(m) da(s) estratégia(s)/tecnologia(s).

3.2.7 Estratégias para a síntese de dados

Os dados apresentados pelo quadro construído a partir das indicações da Seção 3.2.6 são complementados por uma síntese descritiva individual de cada referência selecionada.

3.2.8 Seleção de referências

A execução das pesquisas nas bibliotecas digitais previamente relacionadas retornou um total de 442 referências, distribuídas como segue: *IEEE Xplore* (140), *SpringerLink* (49), *ACM Digital Library* (6), *ScienceDirect* (57), *Wiley* (25) e *Web of Knowledge* (165).

3.2.9 Identificação de referências relevantes

As referências encontradas pela execução das pesquisas foram avaliadas com base em seus títulos e resumos, com a remoção de trabalhos considerados irrelevantes ou duplicados. Como resultado, foram mantidas 41 referências para análise posterior.

3.2.10 Exclusão de referências pelo não atendimento aos critérios de aceitação

As referências selecionadas após a aplicação dos critérios de inclusão e exclusão foram avaliadas com base nos critérios definidos na Seção 3.2.5. Do total, 39 trabalhos (relacionados no QUADRO 1) foram excluídos por não atenderem aos critérios especificados.

Referência	Resumo	Motivo(s) de exclusão (ver 3.2.5)
(ACUÑA, 2004)	O artigo propõe o desenvolvimento de um sistema de informações para a <i>web</i> que permita o gerenciamento de imagens médicas em XML (<i>Extensible Markup Language</i>), armazenando-as em um SGBD relacional. O uso de XML para a representação de conteúdo objetiva resolver problemas de interoperabilidade relacionados aos padrões DICOM e <i>Analyze</i> , permitindo que <i>softwares</i> menos especializados possam acessar os dados de interesse.	3

(AGOSTINHO; PEREIRA; FREIRE, 2007)	O artigo descreve uma extensão para um servidor DICOM <i>stand-alone</i> (desconectado de um sistema PACS), provendo um esquema de BD voltado ao alto desempenho de consultas e uma aplicação cliente para a definição dos critérios de busca e exibição de resultados.	3
(AHMED, 2007)	O trabalho apresenta uma proposta de arquitetura construída sobre <i>Grids</i> chamada SAAP (<i>Service Architecture for Archiving and Presentation</i>), cujo objetivo é prover uma estrutura para armazenamento e comunicação de dados médicos que possa ser utilizada em um hospital ou mesmo entre hospitais, permitindo o acesso e a troca de informações entre usuários com diferentes perfis. A proposta abrange quatro tópicos de interesse (armazenamento, comunicação, segurança e fluxo de trabalho), relacionando tecnologias úteis na sua implementação.	1,2,3
(ARMBRUST, 2009)	O artigo relaciona conceitos sobre HIS, RIS e PACS, caracterizando cada tipo de sistema e discutindo a necessidade de interconectá-los. Características sobre o padrão DICOM também são apresentadas, além de estratégias para o armazenamento <i>online</i> e <i>offline</i> das imagens adquiridas.	1
(BIAN; SEKER; TOPALOGU, 2010)	O trabalho apresenta um sistema de arquivos chamado JigDFS, construído com o propósito de armazenar imagens médicas de forma segura e eficiente. A abordagem adotada utiliza criptografia e particionamento de arquivos, distribuindo partições criptografadas entre os nodos do <i>cluster</i> onde a solução é disponibilizada.	1,4
(BLANQUER; HERNÁNDEZ; SEGRELLES, 2005)	O trabalho apresenta um <i>middleware</i> baseado em tecnologias de <i>Grid Computing</i> construído com o objetivo de resolver problemas de compartilhamento e processamento de imagens DICOM em ambientes distribuídos. A estratégia	1,3

	<p>adotada pelo <i>middleware</i> envolve a criação de repositórios virtuais, integrando diferentes repositórios físicos e oferecendo serviços de busca global, transmissão progressiva de conteúdo, encriptação automática e pseudo-anonimização.</p>	
(BROWN, 2005)	<p>O trabalho descreve um ambiente integrado voltado ao armazenamento de dados quantitativos relacionados a imagens médicas (como por exemplo área e volume de regiões de interesse, histogramas, dentre outros), permitindo a integração de dados de diferentes experimentos com o objetivo de responder a questões de pesquisa. O esquema de BD utilizado é baseado na hierarquia DICOM de quatro níveis, complementado por tabelas específicas voltadas ao armazenamento dos dados quantitativos de interesse (que podem variar de experimento para experimento).</p>	3
(CAICEDO, 2007)	<p>O trabalho apresenta a arquitetura de um sistema para administração de imagens voltado às práticas de ensino, diagnóstico e telemedicina. Desenvolvido pelo Centro de Telemedicina da Universidade Nacional da Colômbia, o sistema multicamadas voltado à <i>web</i> suporta a captura de imagens e metadados, o armazenamento do conteúdo capturado e a extração de características para uso em buscas por similaridade (CBIR – <i>Content-Based Image Retrieval</i>), foco principal da arquitetura proposta.</p>	1,3
(COWEN; DAVIES; KENGYELICS, 2007)	<p>O artigo apresenta uma revisão das tecnologias envolvidas na aquisição de imagens de CR (<i>Computed Radiology</i>), discutindo as variáveis que impactam na qualidade das imagens resultantes e nos benefícios oriundos da inserção de equipamentos que adotam essas tecnologias em <i>workflows</i> clínicos e hospitalares.</p>	1,2,4

(EVANGELISTA; CAMAPUM; AMEMIYA, 2005)	O trabalho apresenta uma aplicação voltada à comunicação e ao armazenamento de imagens DICOM, utilizando os SGBDs PostgreSQL, Firebird e Oracle na implementação de um esquema de BD que inclui tabelas para o armazenamento de um conjunto de dados referente a pacientes, estudos, séries e imagens. Testes da aplicação foram executados no Hospital Universitário de Brasília, utilizando imagens DICOM adquiridas a partir de um equipamento de ultrassom.	3
(HU, 2011)	O artigo descreve uma ferramenta para monitoração de produtividade de equipamentos médicos, baseada em dados representados de acordo com o padrão DICOM. A ferramenta possibilita a coleta de dados a partir de imagens DICOM, a recuperação desses dados para a geração de relatórios e a definição de alarmes vinculados a eventos temporais ocorridos durante a execução de exames.	1,4
(HUANG, 2005)	O artigo descreve a construção de uma estrutura de <i>Grid</i> para armazenamento de dados médicos entre instituições de saúde e centros de pesquisa, integrando PACS federados e provendo recursos para <i>backup off-site</i> .	1
(HUANG; LIU; ZHOU, 2004)	O trabalho descreve uma arquitetura de servidor para armazenamento e comunicação de dados médicos de disponibilidade contínua (CA – <i>Continuous Available</i>), construída com base na redundância de componentes de <i>hardware</i> e na monitoração dos mesmos via <i>software</i> .	1
(IGWE; ELMAGHRABY, 2010)	O artigo apresenta uma estratégia para a melhoria do processo de armazenamento e recuperação de imagens médicas, integrando metadados e dados descritivos de conteúdo das imagens em um modelo de dados construído sobre um SGBD relacional. Experimentos executados	3

	utilizando Oracle como SGBD demonstram a viabilidade da proposta, tanto em termos de desempenho na recuperação de imagens completas quanto na facilidade de indexação de conteúdo.	
(JERONIMO, 2004)	O trabalho descreve o processo de criação de um banco de dados de imagens de colo uterino, utilizadas para pesquisa, treinamento e educação. As imagens adquiridas são relacionadas com dados clínicos textuais, além de terem regiões de interesse demarcadas por especialistas com o objetivo de identificar incidência de câncer.	1,2,3
(KORENBLUM, 2011)	O artigo descreve a construção de um sistema chamado BIMM (<i>Biomedical Image Metadata Manager</i>), idealizado com o objetivo de prover recursos para a coleta de metadados, o estabelecimento de relações entre os metadados coletados com imagens médicas e a recuperação de imagens similares, utilizando critérios para comparação derivados dos metadados extraídos. Os metadados coletados e armazenados pelo sistema são provenientes de <i>input</i> manual, juntamente com metadados extraídos dos cabeçalhos dos arquivos DICOM relacionados.	1,4
(LANGER, 2011)	O artigo discorre sobre estratégias voltadas ao armazenamento de imagens médicas, a anonimização de dados sensíveis, o estabelecimento de relações entre dados em DICOM e outros dados não padronizados e a garantia de acesso a dados históricos, cujos formatos são descontinuados com o passar do tempo.	3
(LIU; ZHOU; DOCUMET, 2005)	O artigo descreve uma estrutura de arquivamento e <i>backup</i> de imagens médicas construída sobre <i>Grids</i> , integrando diferentes pontos de acesso. É apresentada uma breve revisão sobre estratégias e tecnologias utilizadas para <i>backup</i> e	1,4

	recuperação de conteúdo originalmente adquirido por sistemas PACS, seguida de uma proposta de implementação.	
(LOMBILLO-BIOSCA, 2005)	O trabalho descreve a arquitetura de um sistema PACS para a <i>web</i> chamado SAGIMA, desenvolvido pelo Grupo de Bioengenharia, Eletrônica e Telemedicina da Universidade Politécnica de Valência em parceria com a Unidade de Hemodinâmica do Hospital Clínico Universitário de Valência. O sistema é focado no gerenciamento de exames de angiografia digital, permitindo o acesso descentralizado a imagens, laudos e anotações.	1,4
(MACEDO, 2011)	O artigo propõe adaptações na implementação de um trabalho anterior, de forma a melhorar o desempenho no gerenciamento de imagens DICOM utilizando-se HDF5. São avaliadas operações de armazenamento (C-STORE) e pesquisa (C-FIND), tendo seus resultados comparados aos de um SGBD relacional.	1
(MINATI, 2007)	O trabalho propõe a construção de uma estrutura para armazenamento de dados médicos, tanto para exames originalmente adquiridos em formato DICOM quanto para modalidades não previstas pelo padrão. A proposta, batizada BIWS (<i>Bio-Image Warehouse System</i>), objetiva ser um complemento a sistemas PACS estabelecidos, permitindo consultas com base em diagnósticos classificados a partir de uma taxonomia.	1,4
(MONTAGNAT, 2008)	O artigo descreve a modelagem de um sistema para gerenciamento de dados médicos em <i>Grid</i> , com foco na segurança de acesso a dados sensíveis (como por exemplo metadados vinculados diretamente ao paciente). O modelo proposto mantém o armazenamento DICOM localizado em cada instituição, e cria catálogos de metadados e políticas de acesso para cada imagem	1,3

	mantida pelo sistema.	
(MOUGIAKAKOU, 2009)	O trabalho apresenta um sistema modular para gerenciamento e arquivamento de imagens médicas, suporte a diagnóstico e cooperação baseada em ferramentas de telemática chamado DIAGNOSIS. O sistema incorpora técnicas para processamento digital de imagens, que permitem a extração de características usadas no auxílio diagnóstico.	1,3
(OGIELA; TADEUSIEWICZ, 2004)	O trabalho descreve o uso de técnicas para indexação e busca de imagens médicas por conteúdo, através da aplicação de gramáticas livres de contexto. Segundo os autores, a aplicação dessas técnicas permitiria a identificação de imagens com patologias semelhantes, mesmo pertencendo a órgãos diferentes.	1,2,3
(OOIJEN; BHOMER; OUDKERK, 2005)	O artigo apresenta um acompanhamento feito no Hospital Universitário de Groningen, na Holanda, referente ao aumento no volume de dados adquiridos em relação ao tempo. Os autores argumentam que o aumento no volume de dados é decorrente não apenas do aumento no número de pacientes atendidos, mas também da substituição de equipamentos e da adoção de equipamentos para modalidades que, até então, trabalhavam com filme.	1
(PATANACHAI, 2008)	O artigo descreve o uso de um sistema PACS para o gerenciamento de imagens odontológicas, construído sobre um banco de dados relacional e um visualizador DICOM.	1,4
(PUTTEN, 2008)	O trabalho descreve a experiência de uso de uma infraestrutura para armazenamento de imagens relacionadas à cardiologia chamada THOPACS. Instalado em Rotterdam, o sistema permite o acesso local/remoto às imagens adquiridas a partir de equipamentos cardiológicos, além de prover níveis diferenciados de	1,4

	armazenamento para conteúdo <i>online</i> e histórico de exames.	
(REINER, 2011)	O artigo discute a possibilidade de uso de dados coletados durante o processo de avaliação de imagens radiológicas, com o objetivo de otimizar a construção e melhorar o desempenho de <i>workflows</i> voltados à análise dessas imagens. É proposto o armazenamento dos dados coletados em bancos de dados normalizados entre instituições, de forma que possam ser realizados estudos longitudinais de comportamento com base no histórico de avaliações executadas por radiologistas.	1,2,4
(ROSSET; ROSSET; RATIB, 2005)	O trabalho propõe o uso de ferramentas de <i>hardware</i> e <i>software</i> de uso comum integradas a sistemas PACS, de forma a flexibilizar o armazenamento, a transmissão de dados e o estabelecimento de conferências não-presenciais para discussão de casos. Como prova de conceito, um <i>software</i> de código aberto foi modificado para prover funcionalidades relacionadas a armazenamento e recuperação de imagens a partir de dispositivos portáteis, bem como compartilhar telas durante <i>chats</i> interativos, ponto-a-ponto.	1,3
(SELVER, 2007)	O artigo descreve as funcionalidades de um <i>software</i> construído para a edição de funções de transferência, úteis no processamento de imagens médicas para o destaque de órgãos e regiões de interesse.	1,3
(SLAVICEK, 2010)	O artigo apresenta o projeto MeDiMed, um sistema PACS regional instalado na República Tcheca que objetiva prover compartilhamento de imagens a um conjunto de hospitais da região de Brno.	1,4
(TENG, 2010)	O trabalho descreve um protótipo para armazenamento DICOM em <i>Cloud</i> , construído sobre a plataforma Windows Azure, provendo serviços escaláveis e	3

	replicáveis abrangendo armazenamento, indexação e acesso via <i>web</i> para consultas e recuperação de imagens.	
(VOGL, 2005)	O artigo descreve a experiência de implantação e uso de uma estrutura para arquivamento de dados médicos em formatos diversos, conhecida como AIM (<i>Advanced Image Management</i>), desenvolvida na Áustria. Além da estrutura de armazenamento, o projeto prevê a transmissão segura de dados entre as instituições parceiras e a utilização de um <i>software</i> unificado para a visualização de conteúdo.	1
(WARNOCK, 2007)	O trabalho apresenta as características do DCM4CHE, um projeto de código aberto para o gerenciamento de dados médicos envolvendo múltiplos padrões (como por exemplo DICOM e HL7). São apresentados detalhes de sua organização, bem como características do módulo utilizado especificamente para armazenamento de conteúdo (DCM4CHEE).	1,4
(YAKAMI, 2011)	O artigo descreve o projeto e a implementação de um sistema para armazenamento de imagens DICOM baseado em componentes de <i>hardware</i> comuns. A proposta utiliza sistemas de arquivos para a persistência das imagens, permitindo a criação de diferentes visões sobre a hierarquia de diretórios definida fisicamente.	1
(YANG, 2008)	O artigo apresenta um método para a execução de <i>queries</i> envolvendo imagens médicas chamado MIFAS (<i>Medical Image File Accessing System for Co-allocation Data Grids</i>), construído e disponibilizado para execução em <i>Grids</i> . O método proposto utiliza-se de recursos de co-alocação para o <i>download</i> paralelo de imagens, o que contribui para a redução do tempo total dispendido na localização e na	1,4

	recuperação de conteúdo para visualização e avaliação.	
(YU, 2008)	O trabalho descreve uma aplicação para a conversão de dados eletrocardiográficos adquiridos originalmente em DICOM para o formato XML, com o objetivo de indexá-los e armazená-los em um SGBD relacional para posterior distribuição via <i>web</i> .	1,3
(YUAN, 2008)	O artigo descreve um sistema para recuperação de imagens de ressonância magnética com base na extração e comparação de características. O sistema proposto pode ser integrado a um PACS existente, e disponibiliza aos usuários recursos para busca textual e por conteúdo de imagem.	1,4
(ZHANG, 2005)	O trabalho descreve a modelagem e a implementação de um protótipo chamado WebMIA, que objetiva prover um novo paradigma para armazenamento, manutenção, compartilhamento, atualização e recuperação de dados médicos. O protótipo engloba um conjunto de ferramentas para armazenamento e processamento de dados, anotação e fusão de imagens médicas pertencentes a diferentes modalidades de exame e indexação/busca por conteúdo textual e de imagem.	1,2,3

QUADRO 1 - TRABALHOS EXCLUÍDOS DA REVISÃO SISTEMÁTICA PELO NÃO ATENDIMENTO AOS CRITÉRIOS DEFINIDOS. FONTE: o autor (2012)

3.2.11 Síntese do conteúdo das referências selecionadas

Os dois trabalhos remanescentes do processo de inclusão, exclusão e aplicação dos critérios de aceitação foram classificados de acordo com as características identificadas como as mais relevantes para esta revisão sistemática (relacionadas na Seção 3.2.6); essas características foram compiladas no QUADRO 2 e complementadas por sínteses individuais dos trabalhos.

Estudo	Estratégia(s)	Tecnologia(s)	Cenário(s)	Modalidade(s)
(POWER, 2004)	Banco de dados objeto-relacional	SGBD DB2 – IBM	Projeto eDiaMoND – Reino Unido	Mamografia
(CHANDRASHEKAR, 2006)	DAL (<i>Data Access Layer</i>)	Pode ser implementado sobre diferentes estruturas de armazenamento	Abrange diferentes pontos de um <i>workflow</i> hospitalar	Não há restrições

QUADRO 2 - SÍNTESE DAS CARACTERÍSTICAS RELEVANTES PARA OS TRABALHOS AVALIADOS. FONTE: o autor (2012)

3.2.11.1 *A relational approach to the capture of DICOM files for Grid-enabled medical imaging databases* (POWER, 2004)

O artigo apresenta a arquitetura definida e adotada para o banco de imagens médicas do projeto eDiaMoND, um protótipo construído para o armazenamento e a organização de dados provenientes de exames de mamografia no Reino Unido. O projeto objetiva o estabelecimento de uma base para o auxílio diagnóstico, o treinamento de novos profissionais de saúde e a geração de dados epidemiológicos.

Apesar de ser direcionada inicialmente a exames de mamografia, a estrutura do banco de dados para o projeto eDiaMoND foi definida de forma a permitir sua utilização com outras modalidades DICOM a partir de um esforço mínimo de expansão. Para tanto, o modelo objeto-relacional foi escolhido e implementado com base no SGBD DB2, da IBM (*International Business Machines*). Uma arquitetura puramente orientada a objetos foi descartada por não apresentar um modelo de dados reconhecido universalmente e por limitar a otimização de consultas pela adoção de encapsulamento de dados.

Em termos de modelagem e implementação, o banco de dados foi dividido em duas partes: a primeira, chamada simplesmente de *repositório*, armazena os dados extraídos dos arquivos no formato DICOM de uma forma não-estruturada; a segunda, chamada de *repositório de informações clínicas*, armazena os dados estruturados extraídos a partir do repositório. A segunda parte armazena apenas os dados relevantes para o contexto do projeto eDiaMoND (um subconjunto dos dados armazenados na primeira parte). O esquema do repositório de informações clínicas (que pode ser visualizado na FIGURA 8 – superior) permite visualizar claramente o mo-

delo hierárquico inerente ao padrão DICOM (FIGURA 8 – inferior), representado como um conjunto de relacionamentos entre entidades.

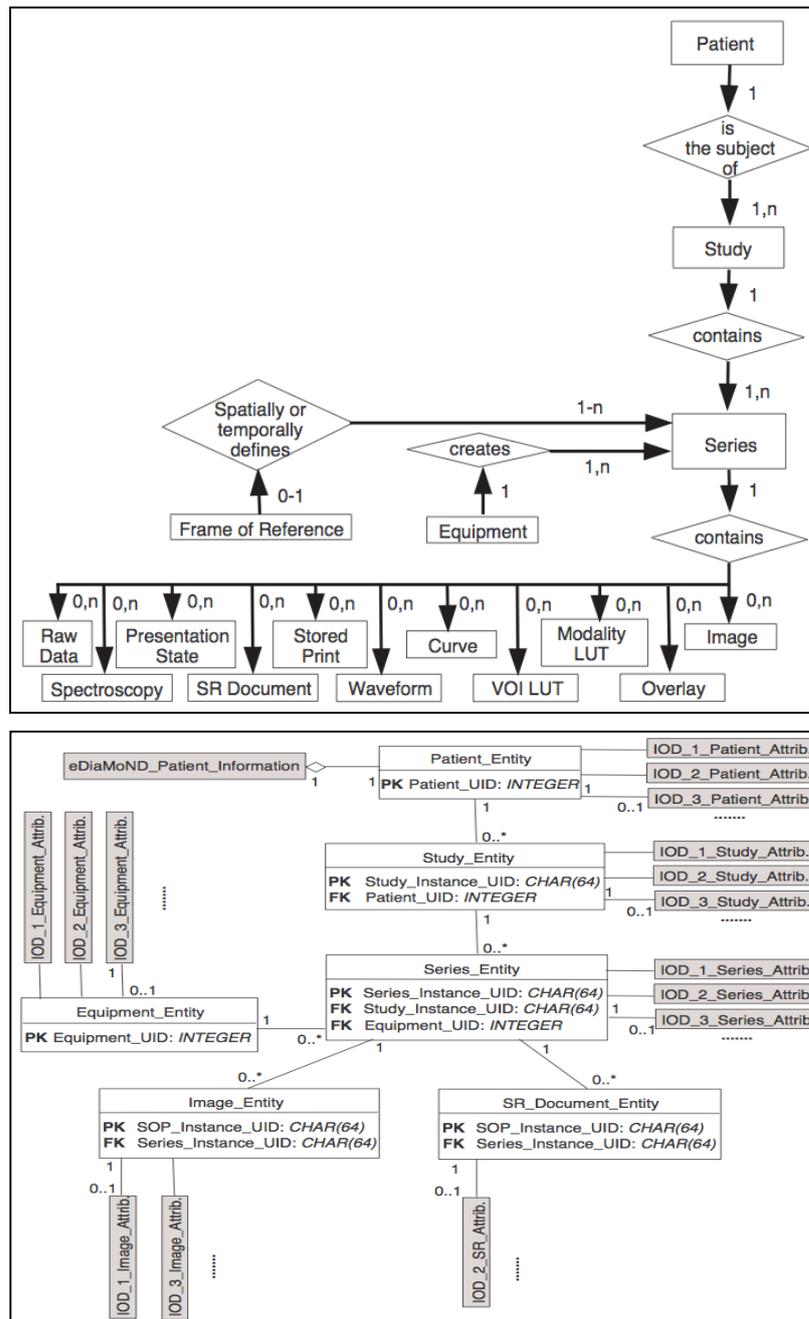


FIGURA 8 – REPOSITÓRIO DICOM PARA O PROJETO eDiaMoND. NA PARTE SUPERIOR: EXEMPLO DE ORGANIZAÇÃO HIERÁRQUICA PARA O PADRÃO DICOM. NA PARTE INFERIOR: MODELO RELACIONAL PARA A ORGANIZAÇÃO HIERÁRQUICA. FONTE: (POWER, 2004)

3.2.11.2 COTS-Like Generic Medical Image Repository (CHANDRASHEKAR, 2006)

O artigo apresenta um repositório genérico para o armazenamento de imagens médicas, passível de ser utilizado em diferentes pontos de um *workflow* hospitalar, eliminando a necessidade de se ter uma estrutura de armazenamento específica para modalidades, *workstations* e PACS. O repositório é implementado como uma camada de acesso a dados (DAL), de forma a abstrair das aplicações cliente o conhecimento sobre a estrutura efetiva de armazenamento (que pode utilizar um banco de dados ou mesmo um sistema de arquivos).

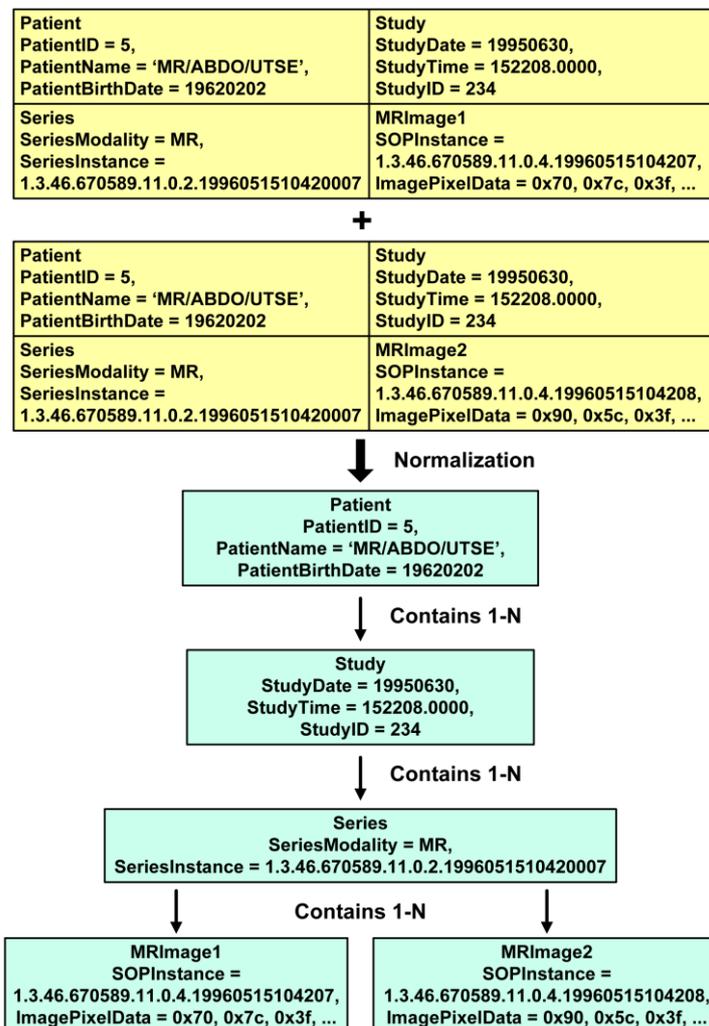


FIGURA 9 – REPOSITÓRIO DICOM CONSTRUÍDO COMO DAL (DATA ACCESS LAYER) ACESSANDO UM BANCO DE DADOS RELACIONAL. FONTE: (CHANDRASHEKAR, 2006) (modificado pelo autor)

Para os casos nos quais o repositório é construído sobre um banco de dados relacional, o modelo proposto gera o esquema de BD com base em um modelo de informação baseado no conteúdo proveniente de diferentes modalidades de exame. O esquema pode ser gerado com diferentes níveis de normalização (como exemplificado na FIGURA 9), objetivando um maior desempenho de acesso.

3.2.12 Discussão

Apesar do número elevado de referências encontradas por meio da pesquisa executada a partir da combinação dos termos de busca, os critérios de inclusão, exclusão e de aceitação reduziram o escopo da revisão sistemática a apenas dois artigos cujo conteúdo é utilizável como resposta às questões de pesquisa. As referências remanescentes indicam duas situações distintas quanto ao armazenamento de imagens médicas no padrão DICOM: uma delas baseada em uma abordagem há tempos consolidada no contexto de banco de dados (o modelo relacional), e outra baseada na pluralidade de estruturas de armazenamento (com um forte apelo quanto à possibilidade de escolha de onde e como armazenar os dados disponíveis).

Em (POWER, 2004), o BD relacional estendido pelo modelo orientado a objetos é utilizado como repositório; fica clara a capacidade de adaptação do esquema de BD, visto que o mesmo é capaz de representar a estrutura hierárquica do padrão DICOM por meio de relacionamentos simples entre entidades. Apesar de se basear nas mesmas premissas definidas por (CODD, 1970), o modelo relacional continua sendo uma das principais opções de escolha quando o armazenamento de dados simples e complexos é considerado, dada a flexibilidade na aplicação de suas regras de normalização e a disponibilidade de um volume cada vez maior de recursos voltados à manutenção de dados não convencionais.

Em (CHANDRASHEKAR, 2006), a abordagem proposta possibilita a adoção de diferentes estruturas de armazenamento – o que permite a escolha entre tecnologias variadas – juntamente com a abstração de como o processo de persistência será executado. Essa estratégia reflete uma característica do padrão DICOM, onde não existe uma especificação que defina como o armazenamento deve ser feito, tampouco a tecnologia que o viabilizará.

A aplicação dos critérios de inclusão e exclusão para a seleção de referências, por sua vez, permitiu identificar que apesar de um número considerável de tra-

balhos acadêmicos (442) citar os termos de busca utilizados, menos de 10% (41) foram classificados em primeira instância; desses, menos de 5% (2) foram relacionados como relevantes. Os percentuais relativos à inclusão e exclusão de referências podem ser visualizados na FIGURA 10 – esquerda. A análise dos critérios de aceitação (cuja ocorrência no processo de avaliação das referências pode ser visualizada na FIGURA 10 – direita) leva à conclusão de que, apesar de extensivamente citado, o processo de armazenamento de imagens médicas no contexto de SGBDs em cenários de PACS é um tema pouco explorado, podendo ser aprofundado em termos de concepção de estratégias, desenvolvimento de tecnologias e utilização em cenários diversos. A maioria das referências foi descartada por não apresentar uma especificação de como a estratégia para armazenamento foi concebida, ou mesmo de como a tecnologia relacionada foi utilizada. A ausência de um modelo de dados bem definido, nesses casos, impossibilita uma análise mais criteriosa.

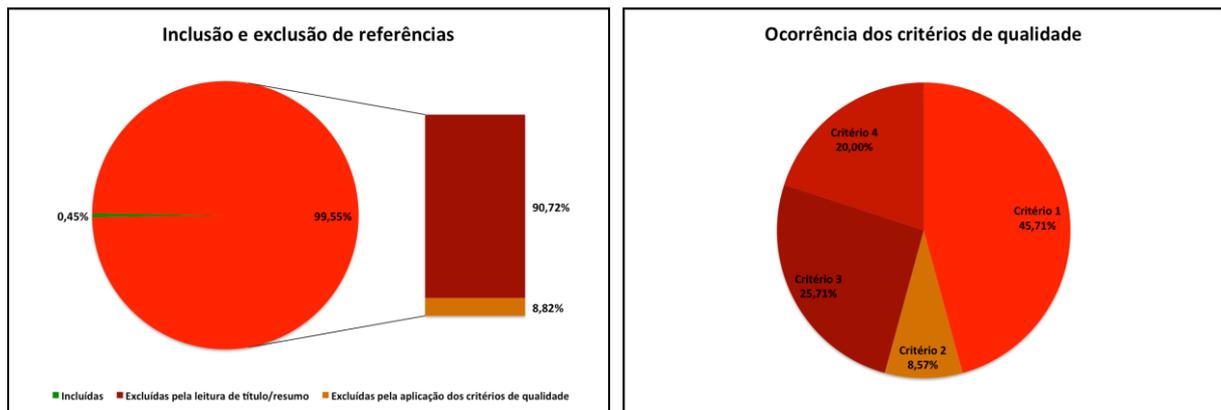


FIGURA 10 – REVISÃO SISTEMÁTICA DA LITERATURA EM NÚMEROS. À ESQUERDA: RESULTADO DA APLICAÇÃO DE CRITÉRIOS DE INCLUSÃO, EXCLUSÃO E ACEITAÇÃO. À DIREITA: PESO DOS CRITÉRIOS DE ACEITAÇÃO NA ESCOLHA DE REFERÊNCIAS RELEVANTES. FONTE: o autor (2012)

A leitura das duas referências selecionadas permite concluir que, em se tratando de banco de dados, o modelo relacional se apresenta como uma opção interessante mesmo em situações que envolvam a manutenção de dados complexos – fato que pode ser comprovado pela utilização do modelo nos dois trabalhos selecionados. Esse modelo pode ser estendido pela adoção de características do paradigma orientado a objetos, que tende a facilitar a representação hierárquica de dados (presente no padrão DICOM). Outra abordagem é a de se utilizar diferentes estruturas de armazenamento (envolvendo desde banco de dados até sistemas de ar-

quivos) de forma abstrata, por meio de camadas de acesso a dados. Essa abordagem é coerente com o padrão DICOM no que diz respeito à inexistência de uma especificação para armazenamento das imagens médicas, com a possibilidade de utilização de diferentes opções escolhidas e implementadas de maneira *ad hoc*.

Uma alternativa à adoção de um modelo puramente relacional ou baseado em sistemas de arquivos está na união das tecnologias, de forma a explorar suas melhores características; assim, a organização do modelo relacional (estendido ou não) garantida pela normalização de dados pode encontrar o alto desempenho e a escalabilidade oriunda de outras tecnologias. O presente trabalho se diferencia de outros encontrados na literatura por prover essa união: nele, SGBDs relacionais são estendidos para incorporar funcionalidades de comunicação específicas a sistemas PACS, utilizando-se de esquemas customizados para armazenamento de dados. Esses esquemas são flexíveis o suficiente para persistir imagens médicas com características diversas, permitindo sua distribuição entre estruturas de armazenamento externas ao esquema de BD por meio de extensões integrantes do padrão SQL. Considera-se a abordagem utilizada como inédita, visto que não foram encontrados trabalhos similares na literatura relacionada.

A revisão de literatura apresentada neste capítulo permite identificar a extensa citação do uso de SGBDs em sistemas PACS, porém sem o aprofundamento necessário ao entendimento de como esses cenários são construídos. Como forma de contribuir com uma descrição detalhada, o próximo capítulo relaciona os materiais e métodos utilizados no desenvolvimento do presente trabalho (incluindo o esquema de BD, as extensões de SGBD e a integração de ambos à arquitetura PACS proposta).

4 MATERIAIS E MÉTODOS

Em consonância com as hipóteses e objetivos específicos introduzidos na Seção 1.3, este capítulo descreve a proposta de arquitetura PACS simplificada e de alto desempenho a ser desenvolvida no presente trabalho. Para cada hipótese/objetivo é apresentada a metodologia de construção dos componentes que integram os protótipos a serem usados nos experimentos (incluindo os recursos de *hardware* e *software* empregados).

4.1 ARQUITETURA PROPOSTA

Em linhas gerais, a arquitetura proposta se baseia na incorporação de serviços pela camada de armazenamento PACS, resultando em uma simplificação pela redução do número de componentes de mais alto nível. Em comparação ao cenário apresentado na FIGURA 1, serviços originalmente oferecidos pelos servidores DICOM e HL7, distribuição de imagens e banco de dados são incorporados por módulos e extensões disponibilizadas pela nova estrutura de banco de dados (como segue) (vide FIGURA 11):

1. o SGBD é estendido para incorporar dois módulos, responsáveis pela gerência de comunicação de dados entre a instância PACS e sistemas externos (HIS/RIS/PACS) e pela gerência de distribuição de dados entre tecnologias heterogêneas de armazenamento;
2. tecnologias heterogêneas de armazenamento são consideradas para o roteamento e persistência de dados convencionais (alfanuméricos), dados não convencionais (binários) e combinações (como imagens DICOM na íntegra), objetivando complementar o modelo relacional com recursos de alto desempenho (como escalabilidade, tolerância a falhas e processamento distribuído de consultas);
3. a comunicação e integração de dados pode ser dividida em dois cenários específicos: a comunicação e integração interna (do SGBD para outras AEs na mesma instituição de saúde), e a comunicação e integração externa (do SGBD para sistemas HIS, RIS e PACS da mesma instituição,

ou mesmo de outras instituições de saúde), ambos implementados a partir da construção e validação de mensagens de acordo com o padrão DICOM.

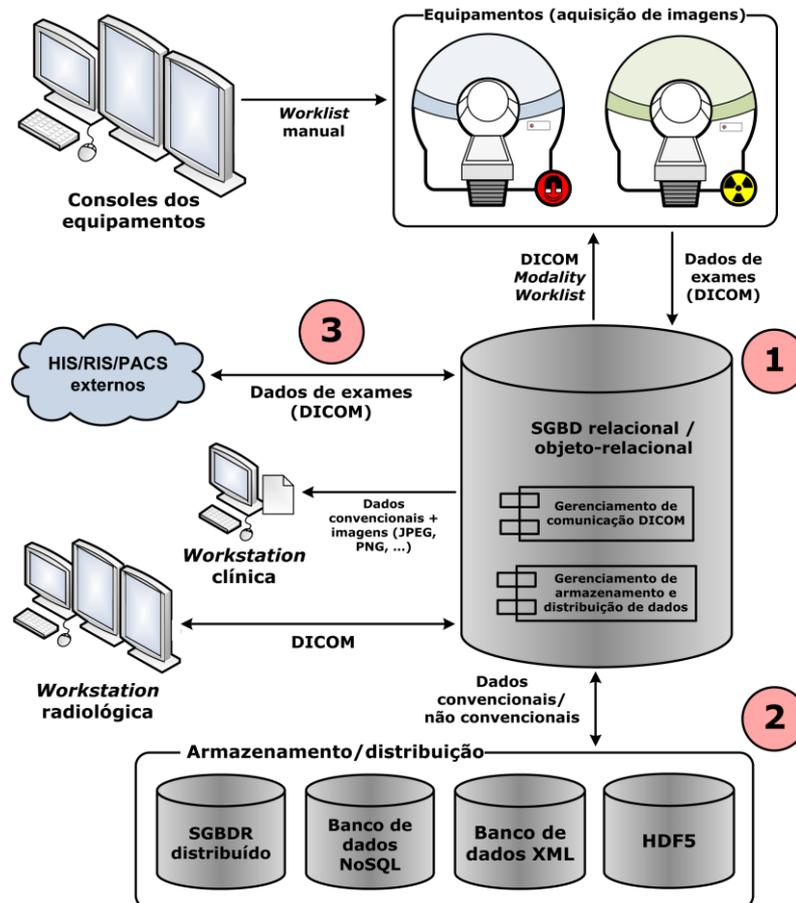


FIGURA 11 – PROPOSTA DE ARQUITETURA SIMPLIFICADA E DE ALTO DESEMPENHO PARA SISTEMAS PACS. DESTAQUE PARA A INCORPORAÇÃO DE SERVIÇOS NA CAMADA DE ARMAZENAMENTO (SGBD), PELA ADOÇÃO DE MÓDULOS E EXTENSÕES. FONTE: o autor (2014)

Um maior detalhamento da camada de armazenamento proposta (FIGURA 12) permite visualizar a relação estabelecida entre os módulos e extensões, além da estratégia de comunicação entre o SGBD e entidades externas.

1. A comunicação entre o SGBD e outras AEs em uma mesma instância PACS, bem como a comunicação entre o SGBD e sistemas externos (HIS, RIS e PACS na mesma instituição, ou mesmo em outras instituições de saúde), é executada através do módulo de gerenciamento de comunicação DICOM (responsável pela construção e *parsing* de mensagens de acordo com o padrão).

2. Operações que demandam persistência ou recuperação de conteúdo são inicialmente tratadas pelo módulo de gerenciamento de comunicação DICOM, que utiliza os serviços oferecidos pelo módulo de gerenciamento de armazenamento e distribuição de dados para efetivar toda e qualquer gravação ou leitura que se torne necessária. Nesse momento, ocorre a tradução das mensagens recebidas originalmente em DICOM para instruções suportadas pela camada de armazenamento (como por exemplo, consultas em SQL).
3. O roteamento dos dados a serem persistidos nas tecnologias heterogêneas de armazenamento, bem como dos dados recuperados como resultado de pesquisas, é de responsabilidade do módulo de gerenciamento de armazenamento e distribuição de dados. Esse módulo comunica-se diretamente com o módulo de gerenciamento de comunicação DICOM, assumindo o papel de servidor de dados para o mesmo.

É importante esclarecer que a visão da arquitetura proposta apresentada nas FIGURAS 11 e 12 é conceitual, e busca isolar os módulos e extensões de acordo com seus respectivos papéis. Em tempo de implementação, pode ocorrer a sobreposição e interseção dos limites conceituais, com determinados serviços sendo disponibilizados pelo próprio SGBD (e não por um módulo em separado), como indicado na proposta.

4.2 EXTENSÃO DE SGBD PARA RECEPÇÃO E ENVIO DE MENSAGENS NO PADRÃO DICOM

As hipóteses #1 e #2 preveem a incorporação de rotinas de *parsing*, interpretação e composição de mensagens no padrão DICOM à camada de armazenamento PACS por meio de uma extensão de SGBD, baseada em um esquema de BD simplificado e de alto desempenho. Essas hipóteses são atendidas, conceitualmente, pelo módulo de gerenciamento de comunicação DICOM (cuja integração à arquitetura proposta pode ser visualizada na FIGURA 12).

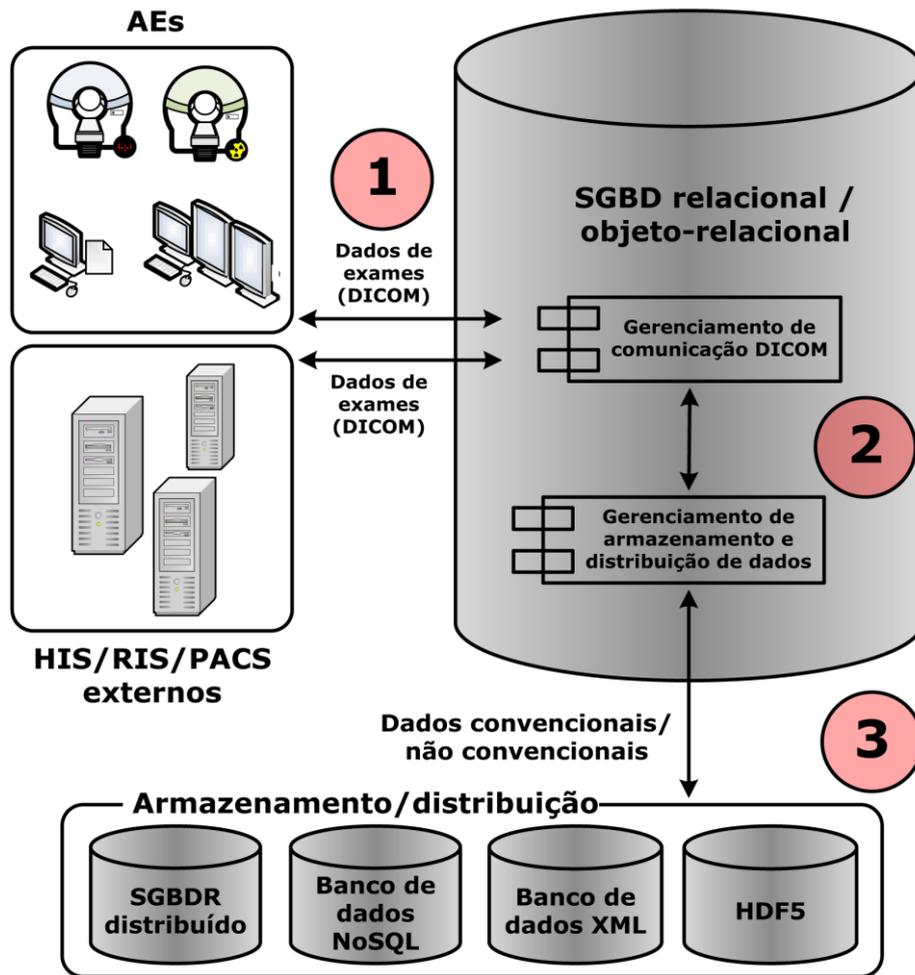


FIGURA 12 – DETALHAMENTO DOS ELEMENTOS QUE COMPÕEM A ARQUITETURA PROPOSTA. DESTAQUE PARA O FLUXO DE DADOS PREVISTO ENTRE OS ELEMENTOS EXTERNOS E INTERNOS AO SGBD. FONTE: o autor (2014)

Neste trabalho, a extensão proposta para o gerenciamento de comunicação foi implementada utilizando-se recursos de SGBDs de código aberto, recursos esses que podem ser considerados mais completos e abrangentes quando comparados a ADTs e UDFs; neles, as funcionalidades implementadas são organizadas como programas completos (e não apenas como funções capazes de receber argumentos e gerar resultados), característica presente tanto nos tipos de dados customizados quanto em funções definidas pelo usuário. Os protótipos implementados respeitam as particularidades de nomenclatura e APIs disponibilizadas pelos SGBDs hospedeiros, sem restringir as funcionalidades necessárias à validação da hipótese #2, e assumem o papel de SCPs para as seguintes primitivas de serviço DICOM:

- C-ECHO: primitiva utilizada para verificar a conectividade de uma AE;
- C-STORE: primitiva utilizada para o armazenamento de imagens;

- C-FIND: primitiva utilizada para a pesquisa de imagens e metadados;
- C-GET: primitiva utilizada para a recuperação de imagens.

4.2.1 MySQL *Daemon Plugins*

A primeira versão do módulo de gerenciamento de comunicação DICOM foi prototipada como uma extensão do SGBD MySQL (MYSQL, 2014a), utilizando-se de sua arquitetura de *plugins* (GOLUBCHIK; HUTCHINGS, 2010a). Essa arquitetura permite que conjuntos de rotinas complexas sejam escritas e compiladas como bibliotecas compartilhadas, instaladas junto ao SGBD e gerenciadas quanto à sua execução pelo próprio SGBD. Como pontos favoráveis, podem ser citados:

- os metadados dos *plugins* são definidos e armazenados no próprio código (e não no dicionário de dados do SGBD), o que facilita a instalação e reduz a incidência de problemas derivados de configurações incorretas;
- *plugins* seguem o ciclo de vida do processo principal do SGBD – são inicializados e interrompidos juntamente com esse processo, de forma automatizada;
- o código de um *plugin* é versionado com relação à API do SGBD onde o mesmo é instalado. Essa restrição garante que *plugins* instanciados pelo processo principal sigam estritamente a API disponibilizada pela versão do SGBD que está sendo executado.

Há, também, limitações a serem observadas. Um *plugin* não tem acesso direto aos dados de uma instância de BD; para acessá-los, torna-se necessário criar uma conexão ao BD desejado utilizando-se de uma API de acesso provida pelo MySQL (de mais baixo nível quando comparada às APIs ODBC – *Open Database Connectivity* ou JDBC – *Java Database Connectivity*).

4.2.2 PostgreSQL *Custom Background Workers*

Objetivando maximizar a experiência com extensibilidade em SGBDs *open source*, uma segunda versão do módulo de gerenciamento de comunicação foi prototipada para PostgreSQL (POSTGRESQL, 2014a), explorando a API para construção de *background workers* customizados (POSTGRESQL, 2014b). Um *background*

worker é um processo filho do processo principal do SGBD (*postmaster*), gerenciado por este em termos de ciclo de vida. Apesar das semelhanças com relação aos *plugins* MySQL, *background workers* são mais livres com relação à gerência de memória e acesso a BDs. Esses processos podem ser configurados para acessar regiões de memória compartilhada (resultando em um melhor desempenho), bem como acessar qualquer BD instanciado no *cluster* em que os mesmos estão sendo executados de forma direta usando SPI (*Server Programming Interface*), a API de acesso e manipulação de dados mais básica disponibilizada pelo SGBD. Tais características favorecem essas extensões no que tange ao desempenho de execução e manipulação de dados; porém, erros na modelagem e programação das extensões podem comprometer a execução do SGBD como um todo, além de contribuir para a corrupção dos dados armazenados.

4.2.3 Esquema de BD

Os protótipos desenvolvidos foram projetados para utilizar um esquema de BD relacional simplificado, organizado de forma a gerenciar imagens DICOM heterogêneas, sendo capaz de armazenar dados convencionais e não convencionais provenientes de equipamentos de diferentes fabricantes e modalidades de exame. O esquema foi criado visando adaptar-se a conteúdos variáveis sem a necessidade de alterações estruturais, sendo uma base efetiva para a execução de pesquisas e recuperação de conteúdo de acordo com as premissas do padrão DICOM.

Para que o esquema de BD pudesse atender aos requisitos de heterogeneidade de armazenamento, adaptação ao conteúdo e desempenho, imagens no padrão DICOM foram avaliadas com relação à sua organização estrutural. Após essa avaliação, concluiu-se que o conteúdo de uma imagem DICOM é estruturado em nível de atributo e semiestruturado em nível de imagem. Em seu nível organizacional mais baixo (o atributo), uma imagem DICOM:

- organiza seus dados através de *tags* identificadas por um número de grupo e um número de elemento, na forma de um par ordenado;
- cada *tag* existente em uma imagem é caracterizada por uma representação de valor (VR) e por uma multiplicidade de valor (VM), que juntas especificam o tipo de dado e o domínio válido para o conteúdo da *tag*, e o número de ocorrências do conteúdo dentro da *tag*;

- as *tags* são definidas em um dicionário de dados padrão, com números de grupo e elemento reservados, cujo conteúdo pode ser estendido por *tags* proprietárias.

Em seu nível organizacional mais alto (a imagem), por sua vez, dados em DICOM:

- estruturam-se como um conjunto de *tags*;
- o conteúdo desse conjunto de *tags* varia de acordo com a disponibilidade de informação durante o agendamento e execução do exame, bem como a modalidade de exame a ser executada e o fabricante do equipamento utilizado.

Essas características, consideradas juntamente com as especificações existentes no padrão DICOM acerca dos procedimentos para pesquisa e recuperação de conteúdo, foram usadas na escolha da estratégia a ser adotada no momento da construção do esquema de BD relacional. Optou-se assim pela customização do chamado *modelo de armazenamento decomposto*, descrito na Seção 2.5. A adoção desse modelo se deu pela possível redução/eliminação da esparsidade dos dados armazenados (uma característica presente no padrão DICOM, independentemente da modalidade de exame), bem como pela possível otimização no acesso a atributos pontuais – uma característica das consultas feitas sobre metadados – onde um conjunto reduzido de atributos é recuperado como resultado de cada pesquisa executada.

4.2.3.1 Esquema de BD baseado no modelo de armazenamento decomposto

A proposta de um esquema de BD baseado no modelo de armazenamento decomposto adapta as características do DSM original a um cenário onde imagens DICOM são armazenadas em sua completude, a uma razão de 1:n entre armazenamento e pesquisa/recuperação. Assumindo que uma imagem é armazenada uma vez, e suas *tags* pesquisadas e seu conteúdo recuperado sempre que necessário, a construção do esquema prioriza o desempenho de consulta em detrimento do desempenho de gravação. As principais diferenças deste esquema de BD com relação a outras estratégias reconhecidas para armazenamento de imagens médicas são as seguintes:

- todas as *tags* definidas no padrão, juntamente com todas as *tags* proprietárias presentes nas imagens, são armazenadas e indexadas no esquema de BD,

provendo total flexibilidade na construção e execução de consultas. Essa generalização permite o armazenamento de conteúdo heterogêneo sem que haja necessidade de modificações no esquema;

- o acesso aos metadados da imagem (*tags* com conteúdo alfanumérico, em sua maioria) é feito pela combinação de qualquer número de chaves de busca requeridas ou opcionais;
- a recuperação de conteúdo pode ser feita no nível de imagem (acessando apenas as *tags* que armazenam as imagens em formato binário – *pixeldata*, na nomenclatura do padrão DICOM), ou em nível de metadados + imagens (combinando dados convencionais e não convencionais).

O esquema de BD proposto é construído sobre o modelo relacional padrão, sendo fisicamente centralizado em uma tabela chamada *chave_hierarquica*. Essa tabela é responsável pelo gerenciamento das chaves substitutas (característica mantida do DSM original), relacionando-as com os valores das quatro *tags* definidas pelo padrão DICOM para identificação dos seus quatro níveis hierárquicos (*patientid*, *studyinstanceuid*, *seriesinstanceuid* e *sopinstanceuid*). Para cada imagem cujo conteúdo é armazenado no BD, um novo registro é criado na tabela *chave_hierarquica* com uma nova chave substituta.

Em uma analogia direta, as *tags* DICOM poderiam ser consideradas como os atributos do DSM original. De acordo com as especificações do modelo (ver Seção 2.5), cada *tag* seria armazenada em uma tabela física específica (o que demandaria um esquema de BD com mais de 6.000 tabelas apenas para o dicionário de dados padrão), e implicaria na necessidade de criação de uma nova tabela para cada nova *tag* inserida no dicionário ou definida de forma proprietária por um fabricante de equipamentos. Visando reduzir essa complexidade, as *tags* extraídas das imagens em tempo de *parsing* são armazenadas em tabelas de acordo com seus VRs, uma tabela por VR. A organização do esquema de BD com base nos VRs das *tags* a serem armazenadas, além de simplificar em muito a persistência, é bastante consistente no que diz respeito a possíveis modificações. Alterações de esquema tornam-se necessárias apenas com a adoção de novos VRs pelo padrão DICOM, fato raro considerando que a adoção de um conjunto reduzido e bem-definido de representações de dados é um dos fatores que garantem a interoperabilidade entre equipamentos e fornecedores de *hardware* e *software*. O benefício imediato na adoção de um esquema de BD com essas características está na possibilidade de armazenar

diferentes combinações de *tags*, sem que haja a necessidade da criação de novas tabelas sempre que novas *tags* padrão ou proprietárias sejam definidas.

Além da adaptação feita com relação à criação de tabelas com base em VRs, uma diferença relevante entre o esquema de BD proposto e o DSM original está no número de atributos por tabela. Enquanto o DSM estabelece a criação de relações binárias, para o armazenamento DICOM são utilizadas relações *n*-árias, com *n* variando de acordo com cada VR. Fisicamente, cada tabela é composta por um atributo que se relaciona via chave estrangeira com a chave substituta da tabela *chave_hierarquica*; esse relacionamento permite, em tempo de execução de consultas, recuperar os valores das *tags* que compõem cada imagem armazenada. As tabelas são complementadas com tantos atributos quanto necessários ao armazenamento mínimo do conteúdo definido por cada VR. Índices são criados em todos os atributos usados como chave primária, chave estrangeira e predicados de consulta.

Como complemento, o esquema de BD prevê a criação de uma tabela para o armazenamento do conteúdo completo das imagens DICOM, sem modificação. Para cada imagem cujo conteúdo é armazenado no BD um novo registro é criado na tabela *conteudo_original*, sendo vinculado via chave estrangeira com a tabela *chave_hierarquica*; o conteúdo originalmente lido a partir das imagens é armazenado, na íntegra, em um campo do tipo BLOB (*Binary Large Object*), sem *parsing* ou extração de *tags*. Essa estratégia otimiza consideravelmente o processo de recuperação do conteúdo completo de imagens individuais ou estudos/séries inteiras; com ela, basta ler o conteúdo armazenado na tabela e encaminhá-lo como resposta a operações de busca/recuperação, ou mesmo persisti-lo em arquivos em disco. Seria possível obter o mesmo resultado a partir das tabelas utilizadas para o armazenamento das *tags* individuais por VR; porém, a reconstrução de uma imagem a partir de suas *n tags* armazenadas em diferentes tabelas é um processo custoso, que implica a transposição dos dados armazenados por VR em uma representação agrupada por imagem. Tal custo pode se tornar proibitivo em cenários onde ocorre interatividade (como por exemplo a recuperação de estudos DICOM completos para diagnóstico). A FIGURA 13 exibe parte do diagrama ER (Entidade-Relacionamento) para o esquema de BD proposto (o dicionário de dados completo pode ser consultado no APÊNDICE 2); o QUADRO 3 resume as diferenças entre o DSM original e a adaptação feita para o presente trabalho.

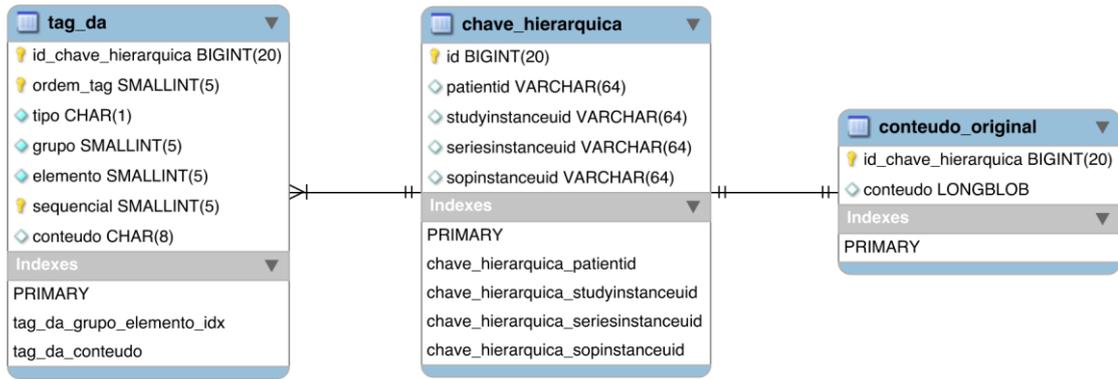


FIGURA 13 – EXTRATO DO ESQUEMA DE BD BASEADO NO MODELO DE ARMAZENAMENTO DECOMPOSTO. O ESQUEMA É CENTRALIZADO NA TABELA CHAVE_HIERARQUICA, COM A QUAL AS DEMAIS TABELAS SE RELACIONAM. FONTE: o autor (2014)

DSM original	Esquema de BD proposto
Para cada atributo do modelo conceitual, estabelece a criação de relações binárias formadas por uma chave substituta e pelo atributo.	Para cada VR DICOM, prevê a criação de uma relação n -ária formada por uma chave substituta e por atributos relacionados ao VR.
Duas relações binárias são previstas para cada atributo: uma delas ordenada pela chave substituta, e outra ordenada pelo valor do atributo.	Índices são usados para a ordenação do conteúdo das relações.
-	Uma relação é criada especificamente para representar a hierarquia do padrão DICOM. Essa relação é formada por uma chave substituta e por quatro atributos, correspondentes aos quatro níveis hierárquicos do padrão.
-	Uma relação é criada especificamente para armazenar o conteúdo completo das imagens DICOM, objetivando melhorar o desempenho das operações de recuperação.

QUADRO 3 – DIFERENÇAS ENTRE O DSM ORIGINAL E O ESQUEMA DE BD PROPOSTO NO PRESENTE TRABALHO. FONTE: o autor (2014)

4.3 DISTRIBUIÇÃO DE DADOS ENTRE TECNOLOGIAS HETEROGÊNEAS DE ARMAZENAMENTO

A hipótese #3 prevê a distribuição de dados entre tecnologias heterogêneas de armazenamento, através do roteamento de dados convencionais e não convencionais para alternativas baseadas em NoSQL, XML, HDF, entre outras (LITH; MATTSSON, 2010; NOSQL, 2012), tendo como ponto de entrada o SGBD relacional ou objeto-relacional. Essa hipótese, apesar de atendida conceitualmente pelo módulo de gerenciamento de armazenamento e distribuição de dados (conforme FIGURA 11), é implementada sobre o esquema de BD usado pelo módulo de gerenciamento de comunicação DICOM de acordo com as premissas da extensão SQL/MED.

4.3.1 PostgreSQL FDWs

A implementação SQL/MED no SGBD PostgreSQL segue as especificações de modularização definidas conceitualmente, com as responsabilidades relativas ao acesso a dados externos divididas entre FSs, FDWs e FTs. Esses módulos são responsáveis, respectivamente, pela conexão do SGBD à fonte de dados que será acessada, pela definição de funções usadas na manutenção dos dados e pela representação interna (relacional) dos dados acessados; como complemento, *user mappings* são disponibilizados para configurar FSs que suportam a especificação de credenciais de acesso.

Inicialmente limitados a acessos somente leitura, PostgreSQL FDWs evoluíram para prover também, a partir de sua API, funções para escrita de dados via FSs. Atualmente, diversas opções de acesso a dados hospedados originalmente em outros SGBDs, bancos de dados NoSQL, arquivos, sistemas de arquivos e serviços na *web* estão disponíveis para uso (POSTGRESQL, 2014c), permitindo a especificação, modelagem e construção de bancos de dados federados heterogêneos, configuráveis e escaláveis de acordo com o volume e tipo de conteúdo a ser gerenciado.

4.3.2 MySQL *Storage Engines*

O acesso a dados externos feito a partir de instâncias de BD hospedadas em *clusters* MySQL é definido pela *Storage Engine API* (MYSQL, 2014b), que corresponde à implementação SQL/MED provida pelo SGBD. Essa implementação simplifica a modularização originalmente proposta, resumindo-se a módulos customizados e incorporados aos processos do SGBD em tempo de execução. Dada a flexibilidade da API, é possível (e recomendado) que os módulos implementem rotinas para as operações CRUD (*Create, Read, Update, Delete*) a serem executadas sobre os dados acessados, utilizando-se de toda e qualquer estrutura de indexação e persistência que possa ser incorporada em tempo de compilação ou execução (GOLUBCHIK; HUTCHINGS, 2010b; PACHEV, 2007). A integração dos módulos customizados à arquitetura do SGBD pode ser visualizada na FIGURA 14.

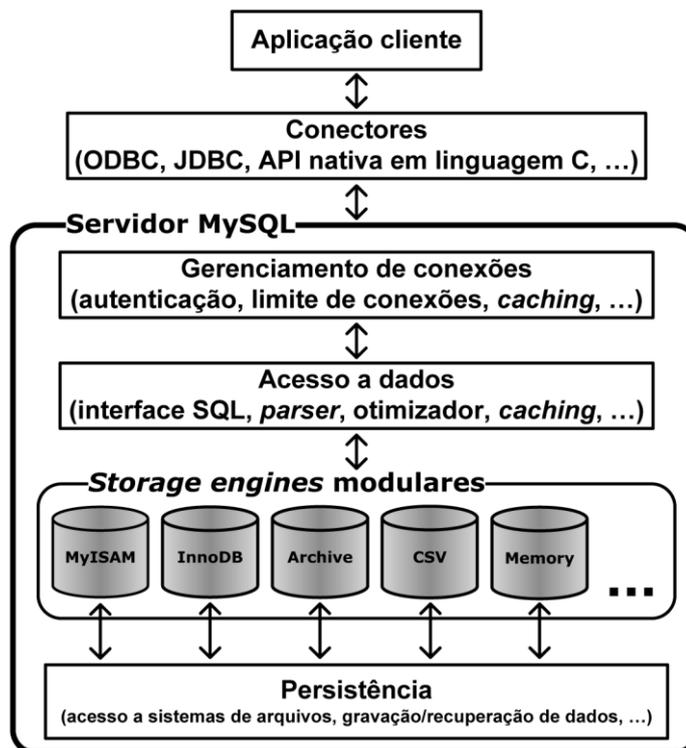


FIGURA 14 – ARQUITETURA MODULAR DO SGBD MYSQL, INCLUINDO *STORAGE ENGINES* CUSTOMIZADOS. FONTE: o autor (2012)

Uma vez compilado e instalado junto ao servidor MySQL, um *storage engine* pode ser utilizado durante a criação de tabelas (conceito equivalente à definição de FTs). Todo e qualquer acesso às tabelas criadas com base em um *storage engine*

específico é traduzido, em tempo de execução, a um conjunto de chamadas de função cuja implementação é prevista pela API e implementada pelo módulo correspondente.

4.4 CONSIDERAÇÕES SOBRE O CAPÍTULO

O primeiro passo para a proposta de simplificação das arquiteturas PACS convencionais consistiu da identificação de um componente ou grupo de componentes capaz de ser estendido, assumindo um maior número de responsabilidades sem comprometimento de funcionalidades já executadas ou perda de desempenho. Pelo fato de as responsabilidades a serem englobadas envolverem comunicação, processamento e armazenamento de dados, a camada de armazenamento PACS foi identificada como o local ideal para as alterações propostas.

Dentre seus elementos componentes, os SGBDs utilizados pelas camadas de armazenamento foram identificados como as melhores opções para customização, dados os seus recursos nativos voltados à extensibilidade e à flexibilidade relacionada à distribuição de conteúdo. Essas duas características, essenciais ao atendimento da proposta, foram exploradas em sua completude; como resultado, obteve-se um modelo (vide FIGURAS 11 e 12) capaz de nortear a construção de instâncias PACS simplificadas, de instalação mais fácil e ágil (graças à redução do número de AEs componentes), com características voltadas ao alto desempenho de armazenamento graças à escalabilidade, tolerância a falhas e distribuição de recursos. Por manter todas as funcionalidades de uma arquitetura PACS convencional, a proposta pode ser considerada como uma substituta viável aos cenários de aquisição e gerenciamento de imagens médicas digitais existentes atualmente.

Os materiais e métodos relacionados neste capítulo serviram como insumos para a construção de protótipos utilizados na verificação das hipóteses formuladas na Seção 1.3. Experimentos foram executados utilizando esses protótipos, gerando os resultados que podem ser visualizados no capítulo a seguir.

5 EXPERIMENTOS, RESULTADOS E DISCUSSÕES

As hipóteses introduzidas na Seção 1.3 foram verificadas a partir da construção de ambientes experimentais, utilizados na execução de testes visando comprovar a viabilidade da arquitetura proposta em termos de configuração e desempenho, bem como atender aos objetivos específicos relacionados. Neste capítulo, esses ambientes são descritos a partir de sua organização de *hardware* e *software*, dos *datasets* utilizados na execução de testes e da metodologia empregada na mensuração do desempenho de cada teste. Os resultados obtidos em cada experimento realizado são apresentados e discutidos, comparando a arquitetura proposta com instâncias baseadas em arquiteturas PACS convencionais.

5.1 AMBIENTE EXPERIMENTAL #1

Uma vez definidos e implementados, o esquema e a extensão de BD (hipóteses #1 e #2) foram submetidos a uma série de experimentos, com o intuito de observar seu comportamento com relação ao armazenamento de imagens heterogêneas, à pesquisa de metadados usando predicados simples e compostos e à recuperação de conteúdo para os níveis hierárquicos estudo, série e imagem. A primeira rodada de testes utilizou *datasets* heterogêneos, disponíveis gratuitamente na internet para fins de pesquisa, cujas características podem ser observadas na TABELA 1 (DICOM, 2014).

TABELA 1 – DATASETS UTILIZADOS NA PRIMEIRA RODADA DE TESTES DO ESQUEMA + EXTENSÃO DE BD

Modalidade de exame	# de estudos	# de séries	Média			Tamanho médio (em bytes) por imagem		Tamanho em disco (MB)
			Imagens por estudo	Imagens por série	Tags por imagem	Metadados	Imagem	
<i>Computed Radiology</i> (CR)	1	6	6	1	80	802	2278594	14
<i>X-Ray Angiography</i> (XA)	2	3	30	20	120	5662	1442097	83
<i>Secondary Capture</i> (SC)	4	6	229	153	64	932	168897	151
<i>Positron Emission Tomography</i> (PET)	9	23	578	226	161	3085	16211	111
<i>Magnetic Resonance</i> (MR)	16	140	307	35	159	2704	72006	363
<i>Computed Tomography</i> (CT)	35	105	846	282	132	3888	109054	3272

FONTE: o autor (2014)

Com fins comparativos, as mesmas operações de armazenamento, pesquisa de metadados e recuperação de conteúdo foram executadas em um servidor de armazenamento DICOM de código aberto (DCM4CHEE versão 2.17.1) (DCM4CHE, 2014), aderente à arquitetura PACS convencional, usado extensivamente por instituições de saúde como parte da infraestrutura para execução de aplicações em ambientes de produção e pesquisa. O servidor emprega uma abordagem híbrida para o armazenamento DICOM, com um SGBD relacional persistindo metadados e um sistema de arquivos persistindo imagens completas. Apesar de utilizar um esquema de BD fixo, horizontal (que permite armazenar um número reduzido e pré-definido de *tags*), elementos de dados não previstos são concatenados e armazenados como BLOBs, seguindo o modelo de atributos interpretados (BECKMANN, 2006). Duas configurações do servidor foram usadas durante os testes:

- *Configuração Padrão (CP)* – *tags* previstas para armazenamento pelo esquema de BD são persistidas diretamente, *tags* não previstas são ignoradas e imagens completas são armazenadas em sistema de arquivos;
- *Configuração Estendida (CE)* – *tags* previstas para armazenamento pelo esquema de BD são persistidas diretamente, *tags* não previstas são concatenadas em campos do tipo BLOB relacionados a um dos quatro níveis hierárquicos do padrão DICOM e imagens completas são armazenadas em sistema de arquivos.

Os experimentos foram executados em um equipamento com a seguinte configuração de *hardware* e *software*: Intel® Core i7 2.7GHz, 4GB DDR3 RAM, 500GB SATA HD, OS X 10.8.4. O SGBD adotado na implementação do esquema de BD baseado no modelo de armazenamento decomposto e na instalação e configuração do servidor de armazenamento DICOM foi MariaDB versão 10.0.3 (MARIADB, 2014a), sem qualquer ajuste de configuração visando melhorias de desempenho.

5.1.1 Resultados e discussão – armazenamento (C-STORE)

Testes de armazenamento foram executados com o objetivo de verificar a adaptabilidade da proposta do esquema de BD a *datasets* heterogêneos. Como resultado, a íntegra do volume de dados descrito na TABELA 1 foi armazenada sem comprometimento ou perda de conteúdo – indicando que o uso de VRs como base para modelagem e construção de tabelas não compromete o processo.

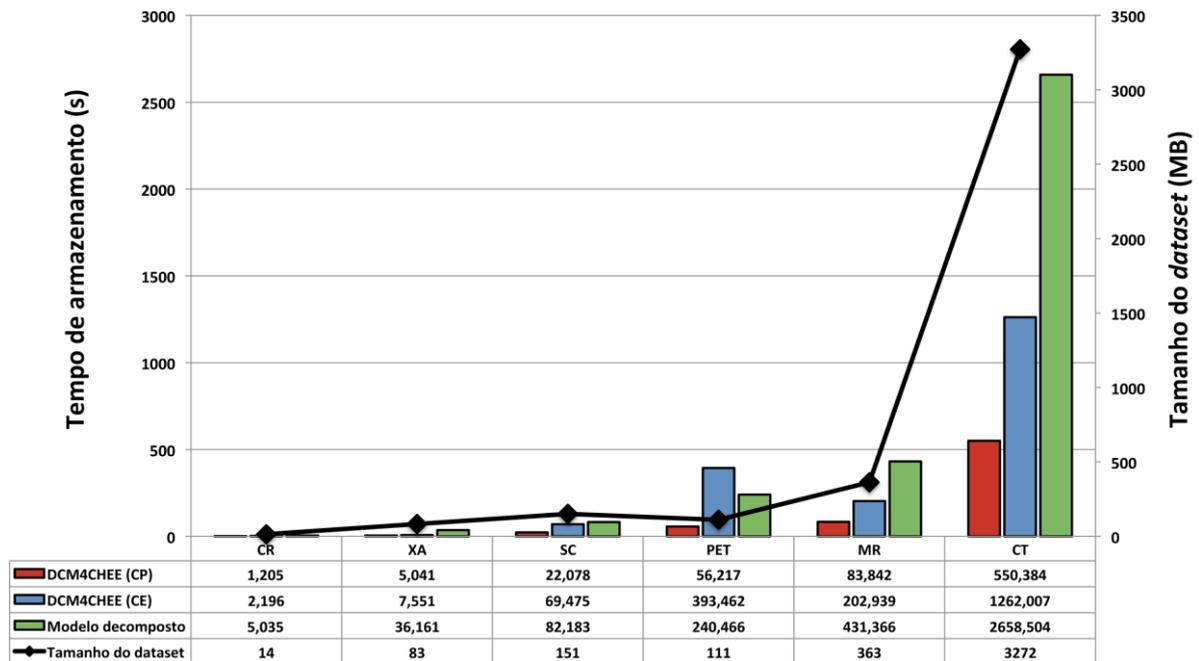


FIGURA 15 – TEMPO DE ARMAZENAMENTO DE *DATASETS* COMPLETOS POR CONFIGURAÇÃO E MODALIDADE DE EXAME. FONTE: o autor (2014)

Em termos de desempenho, a configuração padrão do servidor de armazenamento sobressaiu-se em relação às demais estratégias. Em números gerais, essa configuração foi em média 57,9% mais rápida do que a configuração estendida, e 78,6% mais rápida do que o esquema de BD proposto. Aqui, o melhor desempenho é derivado direto do número reduzido de inserções no BD (decorrência do esquema horizontal adotado pelo DCM4CHEE, em contraponto às tabelas baseadas em VRs do esquema de BD), somado ao fato de que um número reduzido de *tags* é persistido.

Configurado de forma estendida, o servidor de armazenamento apresentou um decréscimo em seu desempenho para todas as modalidades de exame. Considerando que o esquema de BD é o mesmo da configuração padrão (assim como o número de inserções), a queda no desempenho é resultado do processamento necessário à composição dos campos BLOB, envolvendo a concatenação de todas as *tags* não previstas no esquema físico. Modalidades de exame caracterizadas por um maior número de *tags* por imagem são as maiores penalizadas: no armazenamento de imagens PET, por exemplo, a configuração estendida foi respectivamente 85,7%

e 38,9% mais lenta do que a configuração padrão e o esquema proposto, baseado no DSM.

Apesar de consistente, o esquema de BD baseado no modelo de armazenamento decomposto é o mais lento de todos os esquemas testados. A decomposição do esquema conceitual em tabelas físicas mais simples tem um preço, que surge no momento das inserções: para uma imagem formada por n tags, $n+2$ inserções são necessárias (uma inserção para a chave hierárquica, uma inserção para cada tag e uma inserção para o conteúdo completo da imagem). Essa característica (presente no DSM original) é o principal fator responsável pelo esquema proposto ser de 3,7 a 7,2 vezes mais lento do que a configuração padrão do servidor de armazenamento, e de 0,6 a 4,8 mais lento do que a configuração estendida. Um resumo dos resultados discutidos pode ser visualizado na FIGURA 15.

5.1.2 Resultados e discussão – pesquisa (C-FIND)

Experimentos envolvendo a pesquisa de metadados foram executados com base na hierarquia de quatro níveis definida pelo padrão DICOM. Para cada nível, um conjunto de tags foi pesquisado e seus respectivos valores recuperados, uma tag por operação de busca. O número de tags pesquisado variou de acordo com o nível da hierarquia, e as tags usadas nos testes foram escolhidas a partir de listagens presentes na documentação oficial do padrão (uma listagem completa das tags pesquisadas pode ser visualizada no ANEXO 1) (PS 3.4-2011, 2012). De acordo com as listagens utilizadas, foram selecionadas 16 tags para o nível paciente, 27 tags para o nível estudo, quatro tags para o nível série e 25 tags para o nível imagem. O método de pesquisa adotado foi o *método de busca hierárquico* (também definido no padrão), onde predicados construídos para pesquisas em um determinado nível hierárquico devem incluir as tags definidas como *únicas* (analogia às chaves primárias do modelo relacional) do nível em questão, juntamente com as tags únicas dos seus níveis superiores. Após 10 repetições usando valores escolhidos aleatoriamente para as tags de predicado (sendo que esses valores existem nos *datasets*), os tempos médios de execução foram calculados e usados como base de comparação.

Em uma primeira avaliação, percebe-se que o número de tags pesquisado por nível hierárquico é o principal fator relacionado a um melhor ou pior desempe-

no. Independentemente do tamanho do *dataset* para cada modalidade de exame ou do número médio de *tags* por imagem, mais *tags* a serem pesquisadas e recuperadas individualmente implicam diretamente em um maior tempo necessário à execução das operações.

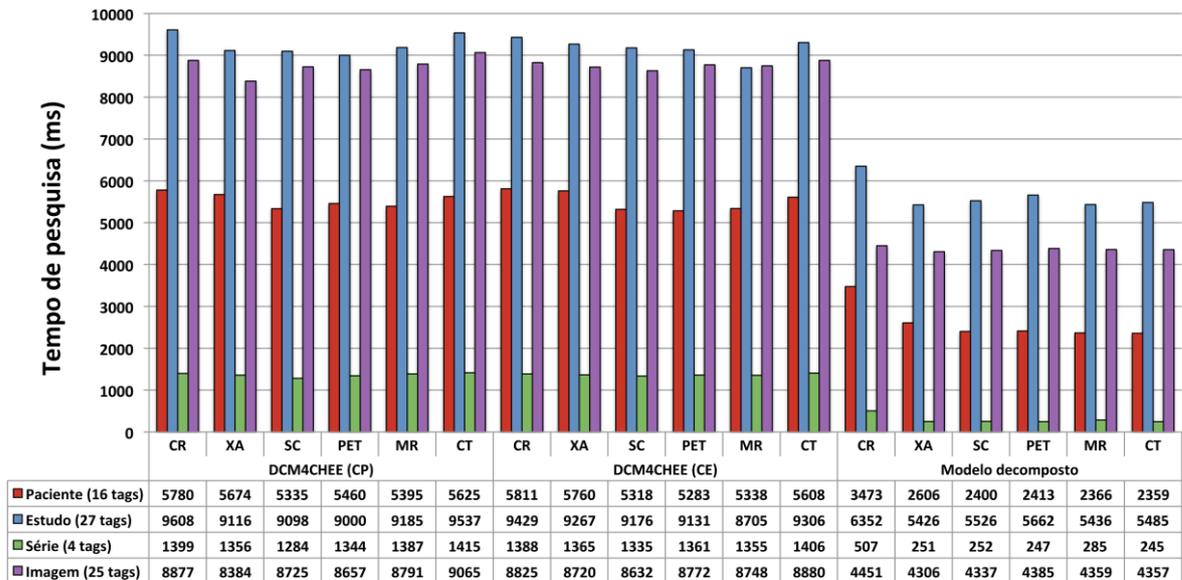


FIGURA 16 – TEMPO DE PESQUISA DE METADADOS POR ESTRATÉGIA DE ARMAZENAMENTO E MODALIDADE DE EXAME. FONTE: o autor (2014)

A diferença de desempenho entre as duas configurações do servidor de armazenamento foi inferior a 1%, considerando o tempo acumulado de pesquisa. Esse resultado é um claro indicativo de que o tempo de processamento empregado na manutenção dos campos BLOB (configuração estendida) é fator secundário para os resultados; como fatores principais, destacam-se a execução das consultas e a recuperação dos dados encontrados.

Devido à sua estrutura física simplificada, o esquema de BD proposto foi em média 48% mais rápido quando comparado ao servidor de armazenamento. O melhor resultado obtido envolveu a pesquisa de *tags* para o nível *série* das tomografias computadorizadas, com uma redução de 82,6% no tempo de pesquisa quando comparado à configuração estendida do servidor. Esse ganho significativo é derivado direto da diferença estrutural dos esquemas físicos dos BDs. Por implementar tabelas com poucos atributos, estratégias derivadas do DSM original (adotando uma organização verticalizada) levam menos tempo para ler e disponibilizar os dados encontra-

dos pela pesquisa, quando um número reduzido de atributos é considerado em cada operação; estratégias horizontalizadas, por sua vez, organizam os dados armazenados em tabelas de estrutura mais complexa, com um maior número de campos – demandando um tempo maior para a recuperação física dos dados. Um compilado dos resultados dos experimentos de pesquisa executados pode ser visualizado na FIGURA 16.

Objetivando avaliar o comportamento do esquema de BD proposto em situações onde grupos de *tags* são pesquisadas simultaneamente, experimentos foram executados a partir dos dados de tomografia computadorizada (a modalidade de exame com o maior volume de dados). Para cada nível da hierarquia DICOM, as *tags* relacionadas foram organizadas em grupos múltiplos de cinco, com cada grupo sendo usado na composição de uma consulta. Após 10 execuções do método de busca hierárquico, com predicados de busca sendo construídos a partir de valores aleatórios extraídos do *dataset*, o tempo médio obtido foi usado como base de comparação.

O comportamento esperado para o servidor de armazenamento foi confirmado: dado o seu esquema de BD horizontal, o tempo acumulado de pesquisa e *fetching* diminui conforme o número de *tags* coexistentes nas mesmas tabelas aumenta. A configuração padrão do servidor é mais rápida em aproximadamente 1,3% quando comparada à configuração estendida. Em uma comparação com o esquema proposto, porém, são observadas variações entre os diferentes níveis hierárquicos. Para o nível *estudo*, por exemplo, o esquema proposto foi mais lento quando comparado a ambas as configurações do servidor. Pelo fato do número de *tags* pesquisadas (27) estar distribuído entre várias tabelas (construídas a partir dos VRs das *tags*), mais tempo é dispendido no acesso, junção e recuperação do seu conteúdo.

A organização (vertical/horizontal) dos esquemas de BD tem relevância nos resultados finais; porém, outros fatores são igualmente importantes (e determinantes) nos tempos de resposta obtidos. A seletividade dos predicados usados nas consultas é um exemplo: no nível mais baixo da hierarquia (imagem), onde os predicados são formados pela combinação das *tags* únicas de todos os níveis, o esquema proposto foi 79,1% mais rápido que o servidor de armazenamento (no melhor caso). Outro exemplo é o de cenários intermediários, onde um número médio de *tags* por nível hierárquico é equilibrado com uma seletividade também média. Para esses casos, o esquema proposto foi, respectivamente, 64,1% e 44,2% mais rápido em

resposta às pesquisas do que as configurações do servidor de armazenamento para os níveis *paciente* e *série*. A FIGURA 17 exibe os tempos obtidos com esses experimentos.

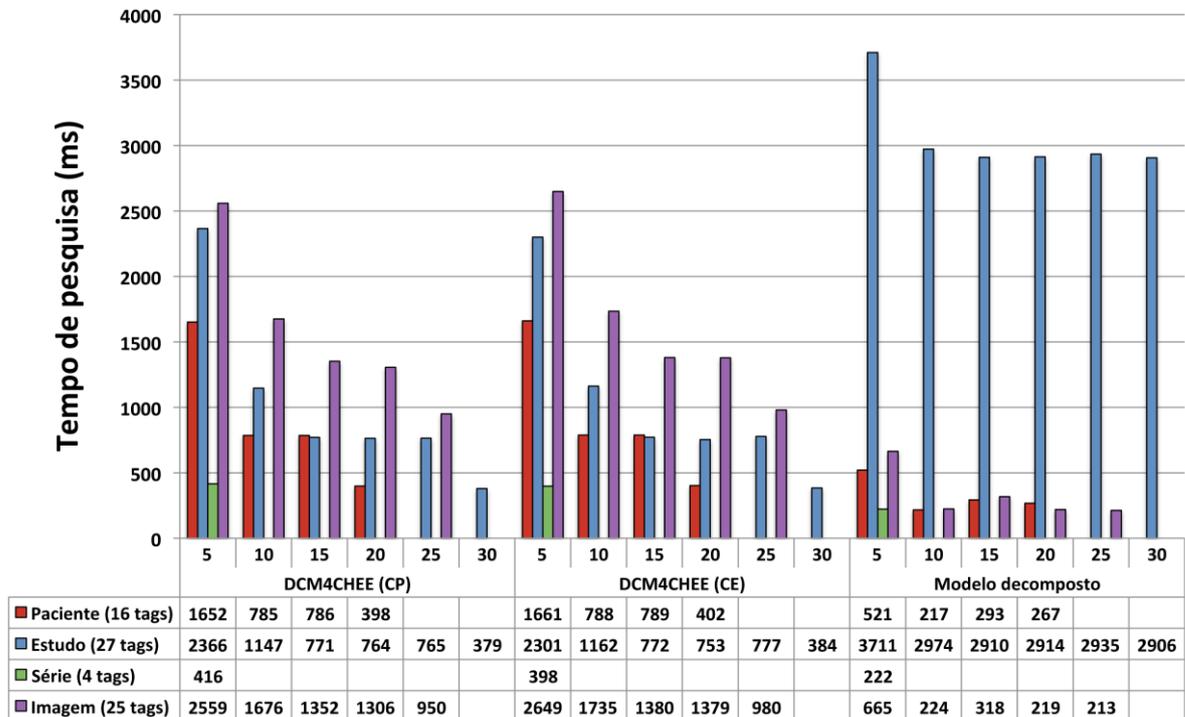


FIGURA 17 – TEMPO DE PESQUISA DE METADADOS POR ESTRATÉGIA DE ARMAZENAMENTO E GRUPOS DE TAGS (TOMOGRAFIA COMPUTADORIZADA). FONTE: o autor (2014)

5.1.3 Resultados e discussão – recuperação (C-GET)

Os testes envolvendo recuperação de conteúdo foram realizados de forma similar aos testes de pesquisa: a hierarquia do padrão DICOM foi usada como base (com as operações sendo executadas para os níveis *estudo*, *série* e *imagem*), e as imagens encontradas pela execução do método de busca foram persistidas em um sistema de arquivos, uma imagem por arquivo físico. O tempo médio de 10 execuções com predicados de busca aleatórios foi assumido como resultado.

Através dos experimentos executados foi possível observar que, para o desempenho de busca, a relação entre o número total de imagens recuperadas por nível hierárquico é tão determinante quanto o tamanho físico da imagem e a complexidade do seu conteúdo. Apesar do senso comum de que imagens maiores impliquem em um maior tempo de resposta, cenários onde modalidades de exame cujas ima-

gens são mais simples (porém, mais numerosas) podem ser mais lentos. Um exemplo foi observado com as modalidades de exame PET (imagens simples) e XA (imagens complexas): considerando o nível hierárquico *estudo*, o volume de imagens foi determinante para que a recuperação de conteúdo da primeira levasse 2,7 vezes mais tempo do que da segunda.

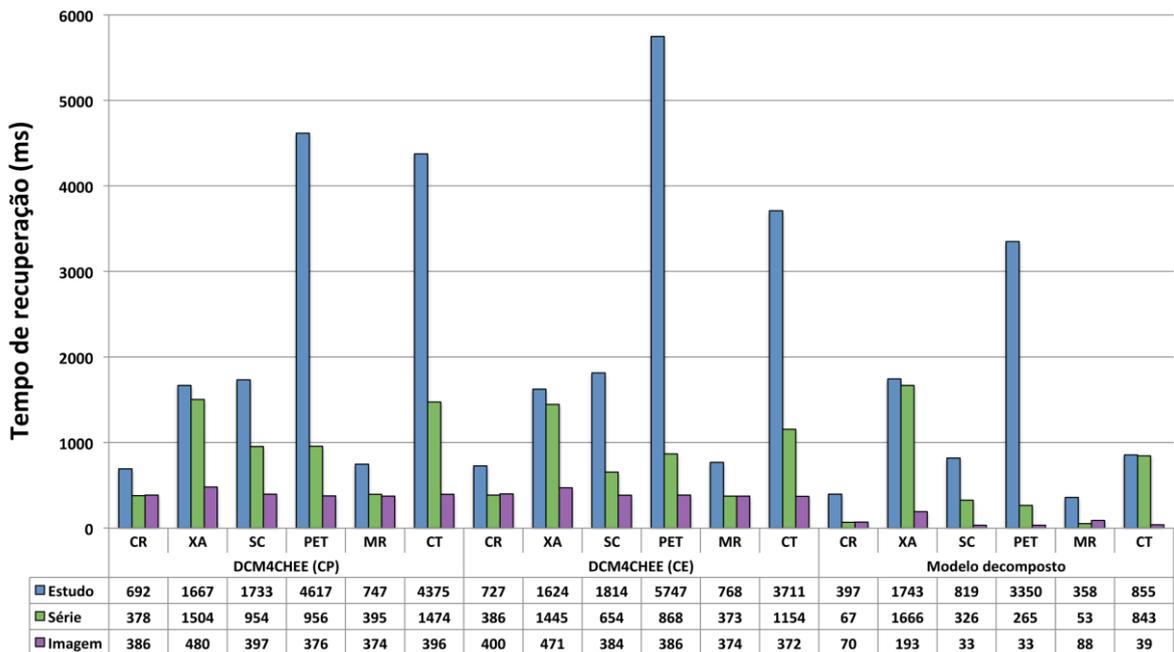


FIGURA 18 – TEMPO DE RECUPERAÇÃO DE IMAGENS COMPLETAS POR ESTRATÉGIA DE ARMAZENAMENTO E MODALIDADE DE EXAME. FONTE: o autor (2014)

A organização das estratégias quanto ao formato de armazenamento também influenciou nos resultados obtidos. A recuperação de conteúdo a partir do servidor de armazenamento implica em uma consulta inicial em sua instância de BD, com o objetivo de identificar o *path* em um sistema de arquivos onde as imagens relacionadas estão efetivamente armazenadas; uma vez encontradas, as imagens são recuperadas e enviadas ao cliente que as solicitou. Como não há qualquer processamento envolvendo conteúdo armazenado como BLOB, ambas as configurações do servidor geraram resultados muito similares (diferença de aproximadamente 1,1% em tempo de execução). Com relação ao esquema de BD proposto, a recuperação de conteúdo implica em uma junção entre as tabelas *chave_hierarquica* e *conteudo_original*; não há a necessidade de acessar o sistema de arquivos, e o con-

teúdo extraído diretamente do BD é enviado ao cliente. Essa simplificação no acesso aos dados resulta em uma redução média de 48,3% no tempo de recuperação dos mesmos, com o melhor caso sendo 91,3% mais rápido (quando comparado com a configuração estendida, para o nível hierárquico *imagem*). Tais resultados justificam a adoção de uma tabela exclusiva para o armazenamento (redundante) do conteúdo completo das imagens, apesar do seu impacto no tempo de armazenamento e no tamanho final da instância de BD. Na FIGURA 18, os tempos de recuperação obtidos por estratégia de armazenamento e modalidade de exame são discretizados.

5.2 AMBIENTE EXPERIMENTAL #2

Uma vez testado com relação ao suporte a dados heterogêneos, o esquema de BD proposto foi submetido a novos experimentos em um cenário voltado à replicação e distribuição de um volume maior de imagens. O objetivo dos testes executados foi avaliar o comportamento da proposta em um ambiente de *cluster*, bem como a possibilidade de manter o esquema físico do BD inalterado.

O cenário de testes foi construído utilizando-se Postgres-XC versão 1.1 (POSTGRES-XC, 2014), uma variante do SGBD PostgreSQL com recursos de escalabilidade implementados em uma arquitetura multimestre síncrona. Nela, acessos de escrita e leitura podem ser executados em diferentes nodos em paralelo, com a garantia de que inserções, atualizações e exclusões serão replicadas de forma síncrona entre os nodos responsáveis pela persistência de conteúdo. *Clusters* Postgres-XC são organizados em três níveis, com cada nível assumindo um papel relacionado ao gerenciamento, comunicação ou armazenamento de dados (como segue):

- *GTM (Global Transaction Manager)* – nodo(s) responsável(is) pelo gerenciamento do ciclo de vida de transações globais, bem como pelo gerenciamento de conteúdo de objetos compartilhados (como por exemplo sequências usadas na geração de chaves substitutas);
- *Coordinator* – nodo(s) responsável(is) pela análise de instruções SQL, geração de planos de execução, encaminhamento de consultas aos nodos de dados e consolidação de resultados (quando necessário). A conexão das aplicações cliente é feita com nodos pertencentes a este nível;

- *Datanode* – nodo(s) responsável(is) pelo armazenamento efetivo dos dados do *cluster*. Nesse nível, dados são replicados ou distribuídos de acordo com o esquema físico do BD.

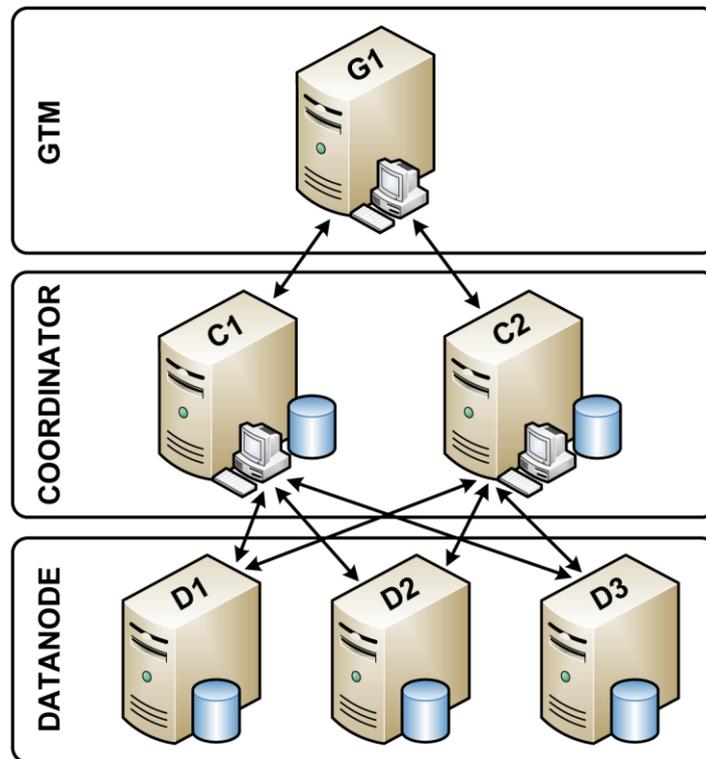


FIGURA 19 – CLUSTER HOMOGÊNEO UTILIZADO NOS EXPERIMENTOS DE REPLICAÇÃO E DISTRIBUIÇÃO DE CONTEÚDO. FONTE: o autor (2014)

Em termos de alto desempenho, *clusters* Postgres-XC são escaláveis nos níveis *coordinator* e *datanode*, sendo possível configurar e inserir novos nodos sob demanda. Novos *coordinators* permitem um balanceamento de carga no estabelecimento de conexões ao *cluster*, suportando um maior número de acessos simultâneos por diferentes aplicações cliente; novos *datanodes*, por sua vez, aumentam a capacidade de armazenamento e permitem distribuir a carga de trabalho em configurações que utilizam particionamento de dados, através da execução de consultas distribuídas. A tolerância a falhas é obtida nos três níveis: pela adoção de dois ou mais GTMs (um ativo e os demais em *standby*), pela configuração de dois ou mais *coordinators* (para que a interrupção dos serviços de um único nodo não impeça o estabelecimento de conexões pelos clientes) e pela configuração de dois ou mais *datanodes* (para que o provimento dos dados se mantenha caso um nodo se torne inoperante). É importante ressaltar que uma configuração tolerante a falhas só será

obtida para os cenários de replicação (em que os dados são copiados para todos os *datanodes*); cenários de particionamento são comprometidos pelo fato de cada *datanode* receber parte dos dados armazenados (que ficarão inacessíveis caso o nodo apresente problemas).

Para os experimentos, um *cluster* homogêneo com seis nodos foi configurado. O acesso ao *cluster* foi executado a partir de um sétimo equipamento (cliente), responsável pelo envio das imagens DICOM para armazenamento, pela execução de pesquisas sobre os metadados e pela recuperação de conteúdo. A organização do *cluster* pode ser visualizada na FIGURA 19; as configurações de *hardware* e *software* dos equipamentos utilizados, por sua vez, podem ser visualizadas no QUADRO 4.

Configurações		Processador	Memória	Armazenamento	Sistema Operacional
Cliente		Intel® Core™ i7 - 2,7GHz	4GB DDR3	500GB SATA	OS X 10.8.3
Cluster	G1	Intel® Xeon® X3440 - 2,53GHz (x8) (compartilhados por virtualização)	4GB DDR3 (por nodo)	859GB SATA (por nodo)	Ubuntu 10.04.1
	C1				
	C2				
	D1				
	D2				
D3					

QUADRO 4 – CONFIGURAÇÕES DE *HARDWARE* E *SOFTWARE* PARA OS EXPERIMENTOS DE REPLICAÇÃO E DISTRIBUIÇÃO DE CONTEÚDO. FONTE: o autor (2014)

Com relação à distribuição de dados no *cluster*, duas estratégias opostas foram utilizadas: *replicação* e *particionamento por round-robin*. Na replicação, há a garantia de que cada *datanode* receba uma cópia de cada registro armazenado; assim, consultas executadas sobre essa configuração podem ser encaminhadas a apenas um nodo (o que permite um balanceamento de carga feito pelos *coordinators* no momento de rotear os planos de execução). No particionamento por *round-robin*, o conteúdo da tabela *chave_hierarquica* continua sendo replicado em todos os *datanodes* (para permitir a execução correta de JOINS); o conteúdo das demais tabelas, porém, é distribuído registro a registro entre os *datanodes* disponíveis, sem replicação. Nessa configuração, os planos de execução gerados pelos *coordinators* são roteados a todos os *datanodes*, que os executam gerando resultados locais cor-

respondentes aos seus conjuntos de dados armazenados. É papel dos *coordinators* consolidar os resultados locais parciais, gerando uma resposta única aos clientes que requisitaram os dados.

Os testes realizados no ambiente experimental #1 focaram a observação do comportamento do esquema de BD proposto com relação ao armazenamento de dados heterogêneos. Considerando os resultados obtidos (o esquema comprovou ser consistente), o ambiente experimental #2 pôde ser direcionado à uma avaliação focada em volume de conteúdo. Para tanto, exames de CT disponíveis publicamente foram utilizados (NATIONAL BIOMEDICAL IMAGING ARCHIVE, 2013); detalhes do *dataset* escolhido podem ser observadas na TABELA 2.

TABELA 2 – DATASETS UTILIZADOS NA SEGUNDA RODADA DE TESTES DO ESQUEMA DE BD BASEADO NO MODELO DE ARMAZENAMENTO DECOMPOSTO

Modalidade de exame	# de estudos	# de séries	Média			Tamanho médio (em <i>kbytes</i>) por imagem		Tamanho em disco (GB)
			Imagens por estudo	Imagens por série	Tags por imagem	Metadados	Imagem	
<i>Computed Tomography</i> (CT)	694	1825	222	84	84	1	491	72

FONTE: o autor (2014)

5.2.1 Resultados e discussão – armazenamento (C-STORE)

Os experimentos executados mensuraram o tempo médio para o armazenamento de uma imagem de CT, dividindo-o em três componentes: o tempo de *parsing* necessário à extração das *tags* individuais das imagens, o tempo de inserção acumulado das *tags* extraídas (correspondente à soma do tempo individual para cada *tag*) e o tempo de inserção do conteúdo completo da imagem. Com fins comparativos, o *cluster* foi utilizado para hospedar as duas estratégias definidas anteriormente: replicação e particionamento (*round-robin*); um terceiro cenário foi construído, provendo uma instância de BD de nodo único configurada sobre PostgreSQL 9.3 (POSTGRESQL, 2014a). Todos os cenários foram submetidos a acessos de escrita concorrente (com o número de processos variando de um a cinco), visando avaliar o impacto da arquitetura de três níveis usada no *cluster* quando comparada a uma instalação independente, que não necessita de uma coordenação remota para o gerenciamento de transações.

Para todas as combinações de *cluster*, nodo único e número de processos concorrentes, percebe-se um padrão relacionado à distribuição do tempo entre os três componentes destacados. A inserção individual de *tags* é a etapa mais lenta (correspondendo a, em média, 59% do tempo total), seguida pelo processo de *parsing* do conteúdo dos arquivos (28,4%) e pela inserção do conteúdo completo dos arquivos (12,6%). Esse é um claro indicativo de que o número de *tags* usadas para armazenar metadados (geralmente em formato convencional – alfanumérico) é tão relevante quanto o tamanho do conteúdo binário (a imagem propriamente dita), em se tratando de tempo de armazenamento. De forma complementar, o aumento no número de processos concorrentes implica em um também aumento do tempo dispendido na persistência do conteúdo. Por se tratar de armazenamento em SGBD relacional, esse aumento é diretamente relacionado ao controle de concorrência multiversão (MVCC – *Multi-Version Concurrency Control*), implementado pelo SGBD e utilizado para garantir o isolamento entre as diferentes transações em execução.

Com relação ao tempo total de execução, a instância de nodo único apresentou os melhores resultados. Pelo fato de não depender de comunicação interno-dos e de gerenciamento de transações distribuídas, essa configuração foi 51,8% e 34,8% mais rápida quando comparada, respectivamente, às configurações de *cluster* para replicação e particionamento. Considerando apenas os ambientes distribuídos, particionar se mostrou 26,1% mais rápido do que replicar conteúdo. Uma compilação dos resultados obtidos pode ser visualizada na FIGURA 20.

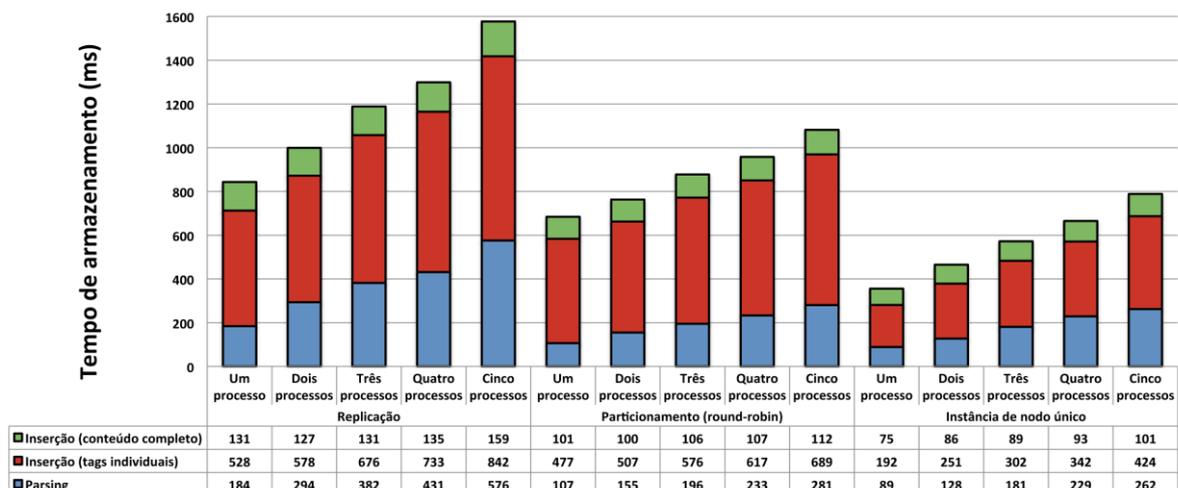


FIGURA 20 – TEMPO DE ARMAZENAMENTO PARA IMAGENS DE CT POR CONFIGURAÇÃO, NÚMERO DE PROCESSOS CONCORRENTES E ETAPA DE GRAVAÇÃO (PARSING, TAGS INDIVIDUAIS E CONTEÚDO COMPLETO). FONTE: o autor (2014)

5.2.2 Resultados e discussão – pesquisa (C-FIND)

Experimentos de pesquisa seguiram a mesma metodologia empregada no ambiente experimental #1, com conjuntos de *tags* relacionadas aos quatro níveis hierárquicos do padrão DICOM tendo seus valores recuperados, uma *tag* por vez. A principal diferença esteve na construção dos predicados de busca, que exploraram quatro das alternativas previstas pelo padrão no que compete à execução da primitiva C-FIND. Essas alternativas englobaram o uso de predicados para buscas por comparação de igualdade usando identificadores únicos, comparação de igualdade usando valores de *tags*, comparação por intervalo de valores de *tags* e comparação por correspondência de padrão (feita com base nos valores de *tags*). Exemplos de instruções SQL que atendem às alternativas de busca podem ser visualizados no APÊNDICE 3.

Considerando-se todas as combinações feitas entre níveis hierárquicos, número de processos sendo executados de forma concorrente e alternativas para construção de predicados, o particionamento por *round-robin* apresentou os piores resultados. Essa é uma decorrência direta da necessária consolidação dos resultados parciais obtidos pelos *datanodes*, feita pelos *coordinators* sempre que uma consulta é executada. Para a consolidação, é necessário que todos os resultados parciais sejam recebidos – o que depende do processamento executado por cada *datanode* do *cluster*; assim, o *datanode* com pior desempenho (seja por um *hardware* menos potente ou pelo armazenamento de um maior volume de dados) acaba por determinar quando a consolidação será iniciada.

A pesquisa das 16 *tags* relacionadas ao primeiro nível hierárquico (paciente) permite observar que as alternativas de busca se comportam de forma similar, independentemente do número de processos dentro de uma mesma configuração. A instância de nodo único destaca-se como a melhor opção, sendo no geral 35,4% mais rápida do que o *cluster* com replicação; já os resultados obtidos com a configuração de particionamento destacam a problemática derivada da consolidação de resultados parciais: sendo em média três ordens de magnitude mais lenta quando comparada à instância de nodo único, torna-se inviável em ambientes que demandem interação. Os resultados individualizados por alternativa de busca, configuração e número de processos concorrentes podem ser visualizados na FIGURA 21.

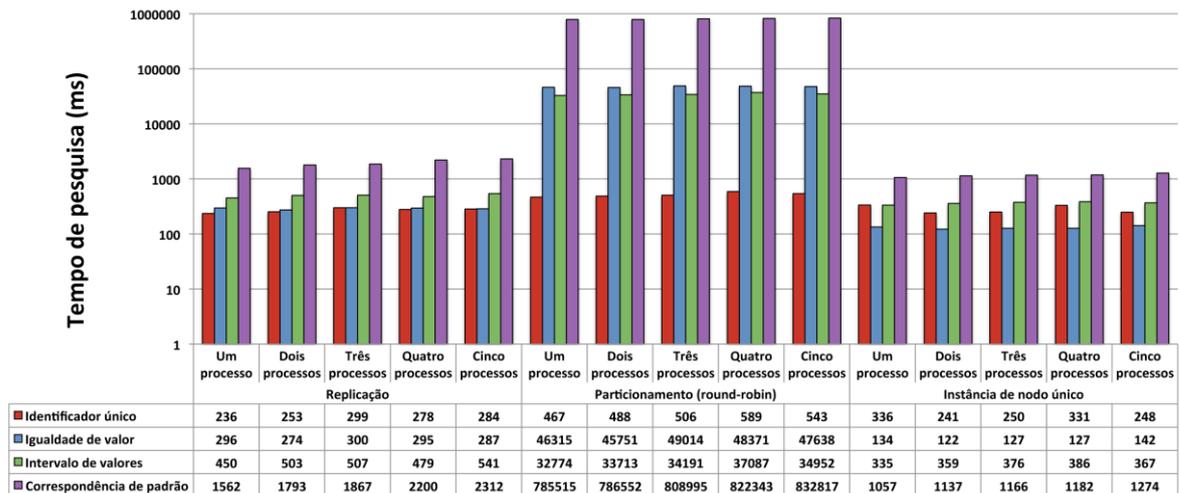


FIGURA 21 – TEMPO DE PESQUISA DE METADADOS POR CONFIGURAÇÃO, NÚMERO DE PROCESSOS CONCORRENTES E ALTERNATIVA DE BUSCA (NÍVEL HIERÁRQUICO DE PACIENTE). FONTE: o autor (2014)

Em nível de estudo, o comportamento observado se repete: o tempo de cada alternativa de busca é bastante similar apesar da variância no número de processos. Mais uma vez a instância de nodo único se sobressai, com um desempenho 45,9% superior quando comparada à configuração de replicação. A configuração de particionamento, por sua vez, obteve os piores resultados de toda a bateria de experimentos: a diferença se manteve em três ordens de magnitude com relação à instância de nodo único, porém com valores individuais mais elevados para as alternativas de busca. Uma análise da FIGURA 22 permite identificar que consultas baseadas em predicados que usam correspondência de padrão são as principais responsáveis pelo comprometimento do desempenho dessa configuração.

Os resultados obtidos para os níveis hierárquicos *série* e *imagem* são bastante similares, em termos de comportamento, aos níveis *paciente* e *estudo*. Apesar das diferenças numéricas, percebe-se que no geral:

- predicados para buscas por comparação de igualdade usando identificadores únicos apresentam um melhor desempenho graças à sua indexação e ao menor volume dados existente na tabela *chave_hierarquica*;
- predicados para buscas por comparação de igualdade usando valores de *tags* e predicados para buscas por intervalo de valores de *tags* apresentam desempenho intermediário, alternando-se como segunda e terceira melhor opção de acordo com a configuração de distribuição de dados;

- predicados para buscas por correspondência de padrão apresentam o pior desempenho. Para esse caso em particular, uso de índices baseados em estruturas como GiST (*Generalized Search Tree*) e GIN (*Generalized Inverted Index*) devem ser considerados como substitutos aos índices baseados em *B-Tree*, usados como padrão.

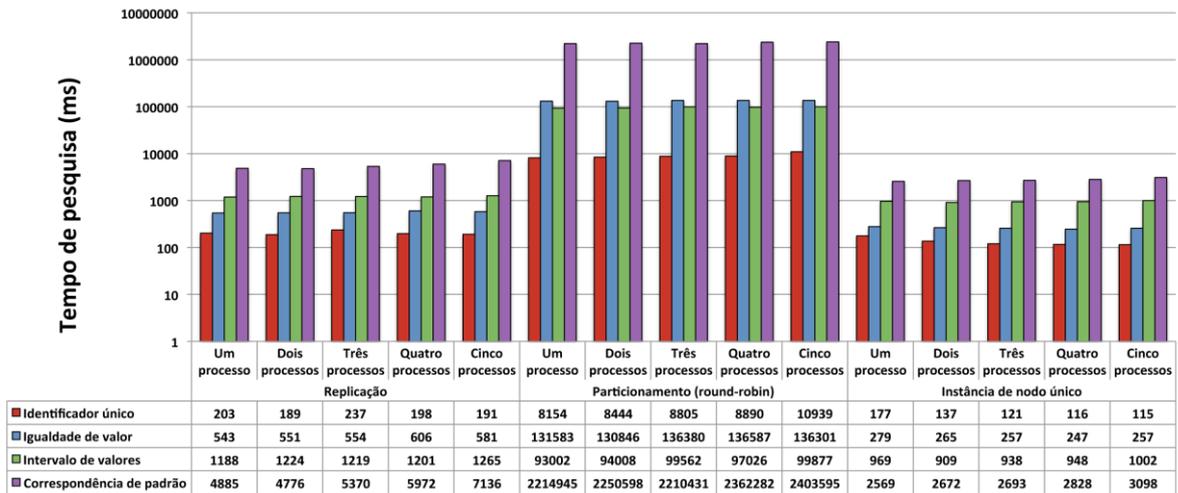


FIGURA 22 – TEMPO DE PESQUISA DE METADADOS POR CONFIGURAÇÃO, NÚMERO DE PROCESSOS CONCORRENTES E ALTERNATIVA DE BUSCA (NÍVEL HIERÁRQUICO DE ESTUDO). FONTE: o autor (2014)

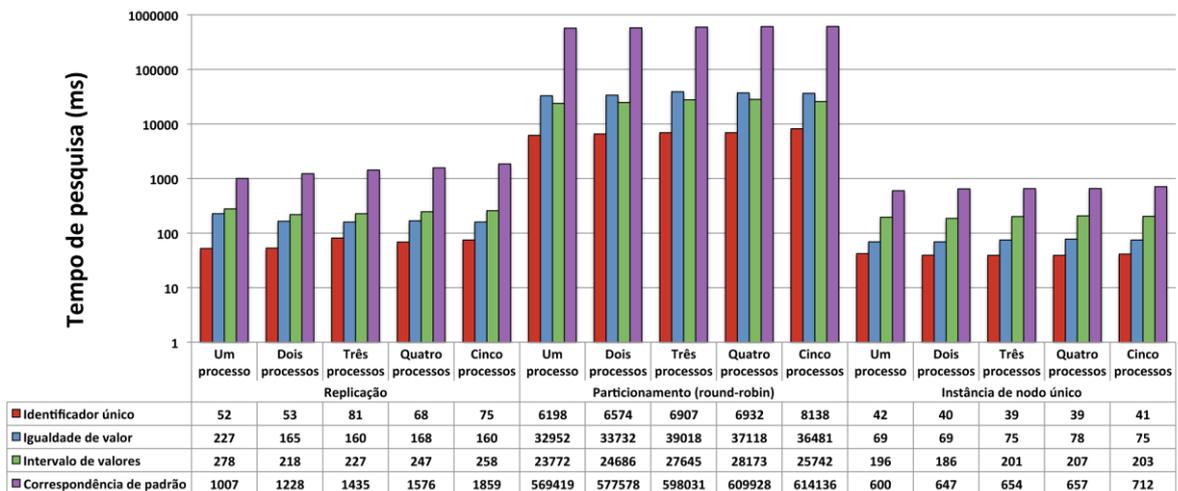


FIGURA 23 – TEMPO DE PESQUISA DE METADADOS POR CONFIGURAÇÃO, NÚMERO DE PROCESSOS CONCORRENTES E ALTERNATIVA DE BUSCA (NÍVEL HIERÁRQUICO DE SÉRIE). FONTE: o autor (2014)

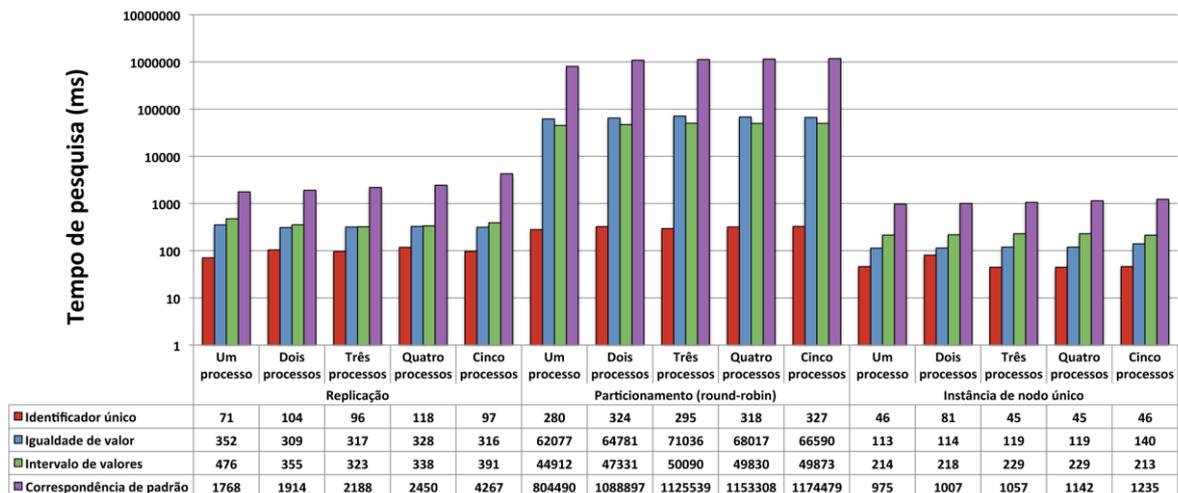


FIGURA 24 – TEMPO DE PESQUISA DE METADADOS POR CONFIGURAÇÃO, NÚMERO DE PROCESSOS CONCORRENTES E ALTERNATIVA DE BUSCA (NÍVEL HIERÁRQUICO DE IMAGEM). FONTE: o autor (2014)

Os gráficos das FIGURAS 23 e 24 exibem, respectivamente, os resultados obtidos para os níveis série e imagem. Em ambos, seguindo o padrão já identificado nos experimentos anteriores, a instância de nó único apresenta um desempenho superior quando comparada à configuração de replicação: 49,4% mais rápida para séries, e 55,4% para imagens.

5.2.3 Resultados e discussão – recuperação (C-GET)

Da mesma forma que os experimentos de pesquisa, os experimentos de recuperação seguiram a metodologia empregada no ambiente experimental #1 – conjuntos de imagens foram recuperadas para os níveis hierárquicos *estudo* e *série*, e imagens individuais para o nível hierárquico *imagem*. Para tanto, os predicados usados nas consultas foram construídos a partir dos identificadores únicos relacionados a cada nível, como pode ser observado no APÊNDICE 4.

Uma diferença fundamental entre os experimentos de pesquisa e recuperação é a razão entre o número de consultas executadas e o volume de dados recuperado em cada execução. Em uma pesquisa (C-FIND), n instruções são executadas (uma por *tag*), devolvendo ao cliente um volume reduzido de dados (comumente mensurados em *bytes*). Em uma recuperação (C-GET), por sua vez, uma única instrução é executada, devolvendo ao cliente dados cujo volume varia de acordo com o nível hierárquico; comumente, esse volume é mensurado em *megabytes*. Ademais,

comparando as instruções dos APÊNDICES 3 e 4, percebe-se que recuperar imagens envolve consultas mais simples do que pesquisar conteúdo de grupos de *tags* individuais.

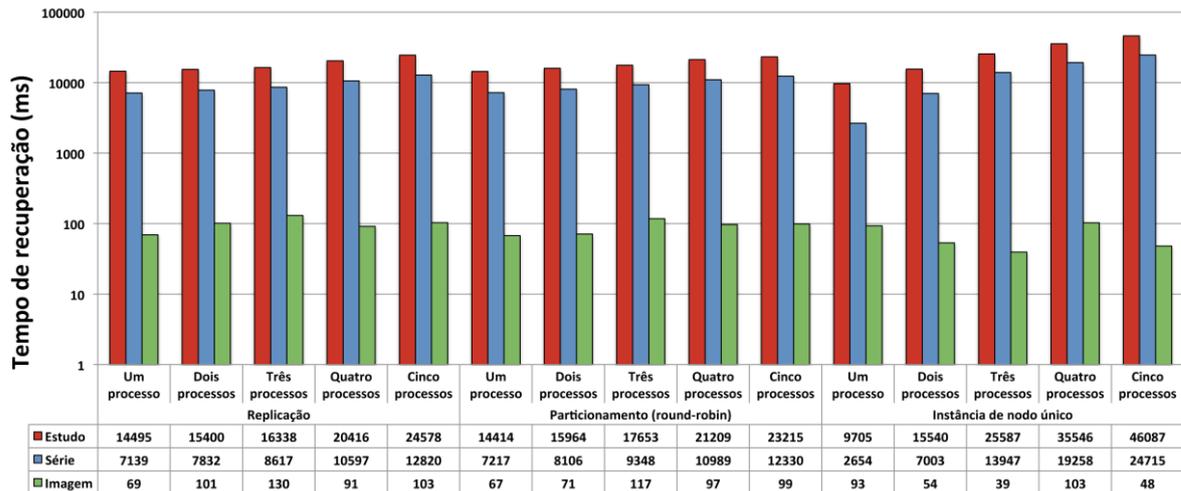


FIGURA 25 – TEMPO DE RECUPERAÇÃO DE CONTEÚDO POR CONFIGURAÇÃO, NÚMERO DE PROCESSOS CONCORRENTES E NÍVEL HIERÁRQUICO. FONTE: o autor (2014)

Na FIGURA 25, é possível perceber que quanto maior a seletividade (um maior número de cláusulas nos predicados de busca), menor é o tempo dispendido na localização, *fetching* e envio do conteúdo das imagens ao cliente. Nesse cenário, as configurações de *cluster* se sobressaem: com particionamento, foi observado um ganho de 29,7% em tempo de recuperação; com replicação (o melhor caso), um ganho de 30,8%.

5.3 AMBIENTE EXPERIMENTAL #3

Os experimentos envolvendo a distribuição de dados entre tecnologias heterogêneas foram executados em um cenário composto por uma instância de BD relacional, interligada a um *cluster* homogêneo através da implementação da extensão SQL/MED disponível no SGBD. A escolha dos componentes de *software* se deu, à época da execução dos experimentos, por uma avaliação empírica de disponibilidade de recursos, robustez de código-fonte e características aderentes ao resultado

pretendido – armazenamento de conteúdo no padrão DICOM em nível de atributo e imagem.

A instância relacional foi disponibilizada em um servidor MariaDB versão 10.0.3, hospedando uma releitura do esquema de BD descrito na Seção 4.2.3.1. Para permitir uma comparação do custo (medido em tempo de execução) das operações de persistência, pesquisa e recuperação de dados entre o modelo puramente relacional e a proposta de armazenamento heterogêneo, o esquema físico implementado foi construído de forma a rotear 100% dos dados a serem armazenados para a estrutura de *cluster*, usando o SGBD como uma “ponte” entre as aplicações cliente e o armazenamento propriamente dito. Essa ponte possibilitou o uso de instruções SQL convencionais para armazenar e recuperar os dados de interesse, abstraindo a real tecnologia de persistência usada pelo *cluster*.

No *cluster*, os dados foram persistidos em uma instância de BD NoSQL implementada sobre Apache Cassandra (CASSANDRA, 2014), seguindo a estratégia de armazenamento particionado de linhas (registros). A definição do esquema de BD utilizado baseou-se nas características do padrão DICOM quanto à estruturação de atributos e semiestruturação de imagens, resultando em um esquema similar ao esquema baseado no modelo de armazenamento decomposto no que compete à capacidade de persistir qualquer combinação de *tags*, para qualquer modalidade de exame e fabricante de equipamento. A organização do esquema de BD, seguida de especificações quanto aos seus componentes, pode ser visualizada na FIGURA 26. O esquema armazena uma *tag* por linha, sendo responsabilidade da *chave da linha* identificar unicamente cada registro. Em teoria, essa chave poderia ser composta pela combinação dos identificadores únicos de cada nível hierárquico do padrão DICOM, juntamente com os valores de grupo e elemento de cada *tag*; porém, é comum ocorrerem multiplicidades na combinação desses valores em uma mesma imagem. Objetivando prover um identificador realmente único, três colunas foram adicionadas ao esquema (todas integrando a chave da linha), de forma a atender às seguintes situações:

- os pares de valores para grupo e elemento podem se repetir dentro de uma mesma imagem através de *sequências de elementos*. Sequências são construções delimitadas por *tags* específicas, e que podem ocorrer *n* vezes em uma mesma imagem. Atributos que estejam aninhados às sequências podem se repetir, também, *n* vezes. A coluna *ordem_tag* visa prover uma distinção

entre *tags* com o mesmo número de grupo e elemento; seu conteúdo é gerado de forma sequencial, em tempo de *parsing*, no contexto individual de cada imagem;

- algumas *tags* DICOM possuem VMs maiores do que 1 (um). Os valores para essas *tags* podem ser divididos em *n* componentes, o que gera múltiplas ocorrências da mesma *tag* para o mesmo valor da coluna *ordem_tag*. A coluna *item*, nesse cenário, é usada como um identificador único para cada parte componente do valor original de uma *tag*. Seu valor também é gerado de forma sequencial, no contexto de cada *tag*, em tempo de *parsing*;
- em algumas circunstâncias (comumente relacionadas a imagens armazenadas de forma compactada), um componente do valor de uma *tag* pode novamente ser dividido em partes. A coluna *subitem* atende à essa condição, por estabelecer um valor único sequencial para cada parte de um componente.

Nível	Coluna	Chave da linha	Chave de particionamento	Chave de clusterização	Índice secundário
Imagem	patientid	*	*		
	studyinstanceuid	*		*	
	seriesinstanceuid	*		*	
	sopinstanceuid	*		*	
	ordem_tag	*		*	
Atributo	grupo				*
	elemento				*
	item	*		*	
	subitem	*		*	
	conteudo_text ¹				*
	conteudo_int ²				*
	conteudo_float ³				*
	conteudo_double ⁴				*
	conteudo_blob ⁵				
	tamanho				

¹ VRs: Application Entity (AE), Age String (AS), Code String (CS), Date (DA), Decimal String (DS), Date Time (DT), Integer String (IS), Long String (LO), Long Text (LT), Person Name (PN), Short String (SH), Short Text (ST), Time (TM), Unique Identifier (UI), Unlimited Text (UT).

² VRs: Signed Long (SL), Sequence of Items (SQ), Signed Short (SS), Unsigned Long (UL), Unsigned Short (US).

³ VRs: Floating Point Single (FL).

⁴ VRs: Floating Point Double (FD).

⁵ VRs: Attribute Tag (AT), Other Byte String (OB), Other Float String (OF), Other Word String (OW), Unknown (UN).

FIGURA 26 – ESQUEMA DE BD IMPLEMENTADO SOBRE APACHE CASSANDRA E ACESSADO VIA INSTÂNCIA RELACIONAL, POR MEIO DE SUA IMPLEMENTAÇÃO SQL/MED. FONTE: o autor (2014)

Funcionalmente, as colunas que compõem uma chave de linha assumem dois papéis distintos. A primeira coluna é utilizada como *chave de particionamento*, e estabelece como os dados são divididos e roteados para os nodos componentes do

cluster. Registros que possuem valores idênticos para chaves de particionamento são roteados para os mesmos nodos. As demais colunas (se houver) são utilizadas como *chaves de clusterização*: além de garantirem a unicidade das chaves de linha, são mantidas ordenadas para agilizar pesquisas.

A coluna *tamanho* registra o chamado *tamanho DICOM* do conteúdo armazenado por cada *tag*. Esse valor é sempre equivalente a um número par, e é garantido pela complementação do conteúdo original (quando tiver tamanho ímpar) com espaços em branco ou valores nulos. Valores armazenados nessa coluna são usados na composição de mensagens, permitindo saber exatamente quantos *bytes* deverão ser enviados às AEs que estejam executando primitivas do tipo C-FIND.

As colunas restantes são dedicadas ao armazenamento do conteúdo das *tags*, bem como dos valores de grupo e elemento. Por uma questão de simplicidade, valores de *tags* cujos VRs sejam compatíveis são armazenados em uma mesma coluna (respeitando a sua tipologia básica), podendo receber índices secundários sempre que passíveis de uso na construção de predicados de busca.

A integração entre a instância relacional e o *cluster* é efetivada por um *storage engine* modular (MARIADB, 2014b), capaz de traduzir operações CRUD a partir do SGBD para execução na instância NoSQL. Essa interação é feita em duas direções: instruções SQL executadas no lado do SGBD são traduzidas para uma (ou mais) instruções equivalentes de acordo com o *framework* Apache Thrift (APACHE THRIFT, 2014) (suportado pelo Apache Cassandra); os resultados obtidos (se houverem) são devolvidos a partir da instância NoSQL, sendo traduzidos para a representação relacional reconhecida pelo SGBD. O cenário completo do ambiente experimental pode ser visualizado na FIGURA 27.

O *storage engine* utilizado para a integração entre as instâncias permite que a tabela criada no SGBD conecte-se a um dos nodos do *cluster*, essa conexão é configurada no momento da criação da tabela, e passa a fazer parte dos metadados da mesma. Pelo fato da arquitetura do Apache Cassandra não estabelecer uma diferenciação entre seus nodos (todos podem ser usados para escrita e leitura), qualquer nodo pode ser escolhido como ponto de conexão. Com relação ao *hardware*, os equipamentos relacionados no QUADRO 4 foram remanejados e reconfigurados de acordo com o QUADRO 5, para atender aos seguintes cenários de experimentação:

- *Instância de Nodo Único* (INU) – composta por um único equipamento, foi utilizada como base para a instalação de uma instância relacional a ser comparada com a instância NoSQL; para alguns experimentos, também serviu de base à uma instalação simplificada do Apache Cassandra;
- *Cluster* (CL) – composto por cinco nodos homogêneos, foi utilizado para a instalação da instância NoSQL; configurações adotadas especificamente para escrita e leitura serão discutidas, respectivamente, na apresentação dos resultados de armazenamento, pesquisa e recuperação.

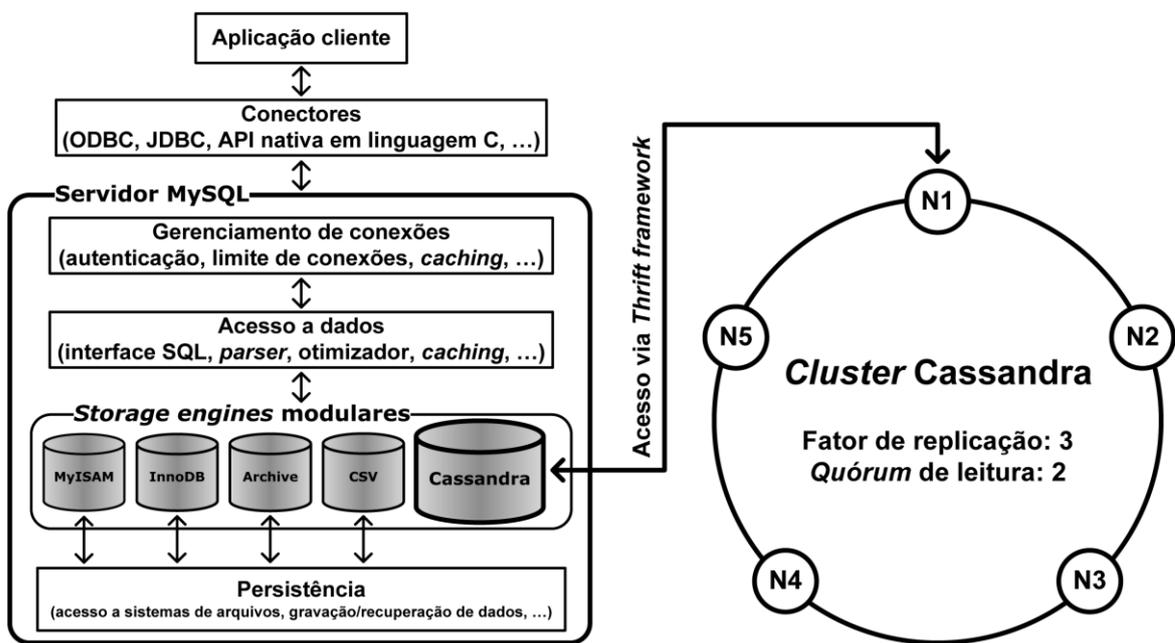


FIGURA 27 – CENÁRIO COMPLETO PARA O AMBIENTE EXPERIMENTAL #3, CONTEMPLANDO A INTEGRAÇÃO ENTRE AS INSTÂNCIAS RELACIONAL E NoSQL. FONTE: o autor (2014)

Configurações	Processador	Memória	Armazenamento	Sistema Operacional	
Cliente	Intel® Core™ i7 - 2,7GHz	4GB DDR3	500GB SATA	OS X 10.8.3	
Nodo único	Intel® Xeon® X3440 - 2,53GHz	4GB DDR3	859GB SATA	Ubuntu 10.04.1	
Cluster	N1	Intel® Xeon® X3440 - 2,53GHz (x8) (compartilhados por virtualização)	4GB DDR3 (por nodo)	859GB SATA (por nodo)	Ubuntu 10.04.1
	N2				
	N3				
	N4				
	N5				

QUADRO 5 – CONFIGURAÇÕES DE HARDWARE E SOFTWARE PARA OS EXPERIMENTOS ENVOLVENDO TECNOLOGIAS HETEROGÊNEAS. FONTE: o autor (2014)

5.3.1 Resultados e discussão – armazenamento (C-STORE)

Para os testes de armazenamento, foram utilizados os mesmos *datasets* descritos na TABELA 1; a escolha se deu pela heterogeneidade de conteúdo, necessária para atestar a adaptabilidade do esquema de BD NoSQL proposto a diferentes combinações de *tags*. Duas modalidades de escrita foram assumidas: escrita por *Processo de Escrita Único* (PEU), sequencial e sem concorrência acessando a instância NoSQL via *storage engine*, e *Múltiplos Processos de Escrita* (MPE), com cinco processos sendo executados de forma concorrente acessando as instâncias relacional (via extensão de BD) e NoSQL (via *storage engine*). Para o armazenamento na instância NoSQL, foi utilizado um fator de replicação igual a três – indicando que cada registro inserido foi roteado para três nodos distintos, de acordo com sua chave de particionamento.

O tempo requerido para armazenamento é derivado de duas características combinadas: o tamanho físico dos *datasets* e a complexidade das imagens em termos de número de *tags*. Um exemplo claro dessa relação pode ser visualizado em se comparando as modalidades de exame SC e PET: apesar de o *dataset* PET ser menor do que o *dataset* SC, demora-se mais para armazená-lo (devido ao seu número de *tags* ser quase três vezes maior). Considerando que o número de *tags* utilizadas especificamente para o armazenamento do conteúdo binário das imagens é o mesmo em ambos os *datasets*, a responsabilidade pela maior ou menor demora é dos metadados – uma característica sempre presente quando se trabalha com armazenamento de imagens completas em nível de *tag*.

Em uma comparação global é possível perceber a repetição de um padrão no tempo de armazenamento, tanto entre diferentes modalidades de exame quanto dentro de uma mesma modalidade (variando os cenários de teste). Para as variantes NoSQL, instâncias de nodo único (acessadas por um ou múltiplos processos) demandam menos tempo para armazenar os *datasets* escolhidos. Com base no tempo acumulado, a redução chega a 89,8% quando comparada ao *cluster* (consequência direta da ausência de replicação e de comunicação entre nodos). O contraponto ao bom desempenho é a escalabilidade: INUs não preveem a incorporação de novos recursos ao cenário, e limitam-se à capacidade do único nodo disponível; além disso, o nodo utilizado torna-se um ponto único de falha (*Single Point Of*

Failure – SPOF), sobre o qual recai toda a responsabilidade sobre a disponibilidade do serviço oferecido.

Como forma de otimizar o tempo de armazenamento, o uso de MPEs é uma alternativa viável. Tanto no ambiente de nodo único quanto no *cluster*, experimentos envolvendo escritas em paralelo geraram resultados superiores em comparação às escritas sequenciais, com um ganho geral de 77,9%. Esse ganho é correspondente ao acesso a um único nodo do *cluster* para escrita a partir do *storage engine* (uma limitação existente no módulo, como pode ser visualizado na FIGURA 27), e em teoria pode ser melhorado com o acesso sendo feito a outros nodos também em paralelo. Há, porém, situações em que o modelo relacional ainda prevalece: como pode ser visualizado na FIGURA 28, o mesmo tem um desempenho superior em armazenar imagens para as modalidades de exame PET, MR e CT.

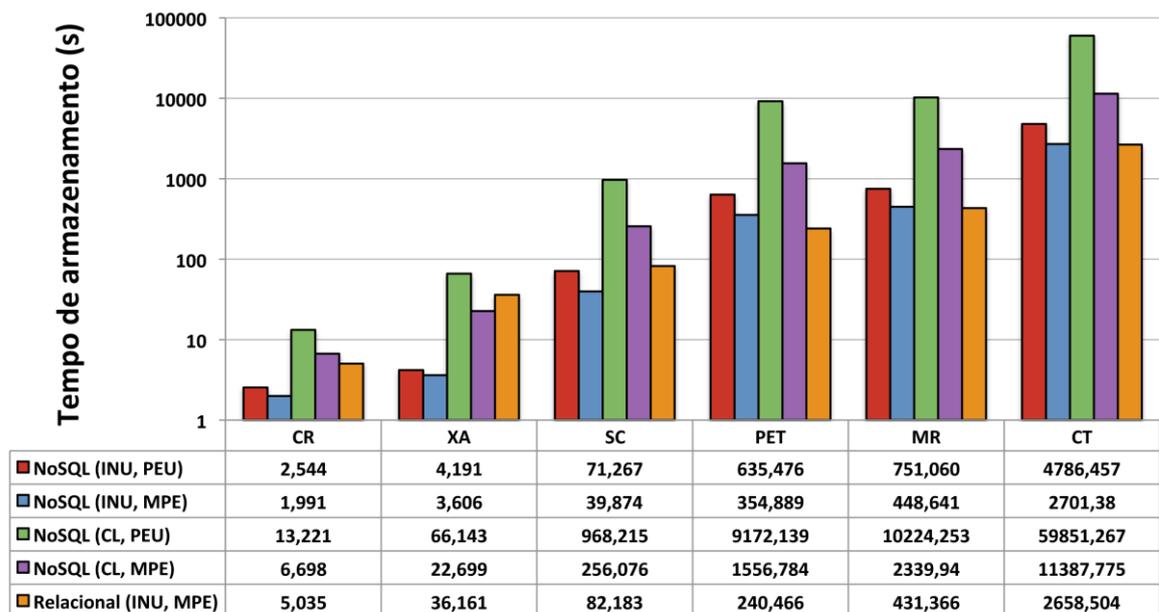


FIGURA 28 – TEMPO DE ARMAZENAMENTO DE *DATASETS* COMPLETOS POR MODALIDADE DE EXAME, EM CENÁRIOS COM VARIÂNCIA NO NÚMERO DE PROCESSOS DE ESCRITA E DISTRIBUIÇÃO DE DADOS. FONTE: o autor (2014)

5.3.2 Resultados e discussão – pesquisa (C-FIND)

Para a pesquisa de metadados, os mesmos procedimentos definidos na Seção 5.1.2 foram seguidos: execução do método de busca hierárquico para os quatro níveis definidos no padrão DICOM, incluindo conjuntos de *tags* relacionadas a cada nível, com 10 repetições utilizando valores aleatórios (porém existentes nos dados

armazenados) para a composição dos predicados de busca. A consistência de leitura é garantida por um *quórum* igual a dois, significando que cada valor retornado está persistido em pelo menos dois nodos do *cluster*.

Como primeira observação, percebe-se a relação direta entre o tempo necessário à recuperação do conteúdo das *tags* para cada nível e o número de *tags* por nível: mais *tags* demandam mais tempo. Apesar de esse comportamento ser observado na grande maioria dos experimentos, há exceções: pesquisas para a modalidade de exame CT, nas variantes NoSQL, demandam mais tempo no nível *série* (quatro *tags*) do que no nível *imagem* (25 *tags*). Essa ocorrência em específico é derivada da organização do padrão DICOM quanto à obrigatoriedade de conteúdo. Na prática, o número de *tags* existentes para o nível *imagem* nos *datasets* escolhidos é menor do que 25 (devido às *tags* não serem obrigatórias); como o esquema de BD implementado pela instância NoSQL recupera dados físicos com base em colunas pontuais (acessando-as individualmente), e não em registros inteiros (recuperando-os na íntegra e escolhendo as colunas relevantes em uma segunda etapa), menos colunas a serem recuperadas resultam em uma redução no tempo de busca total.

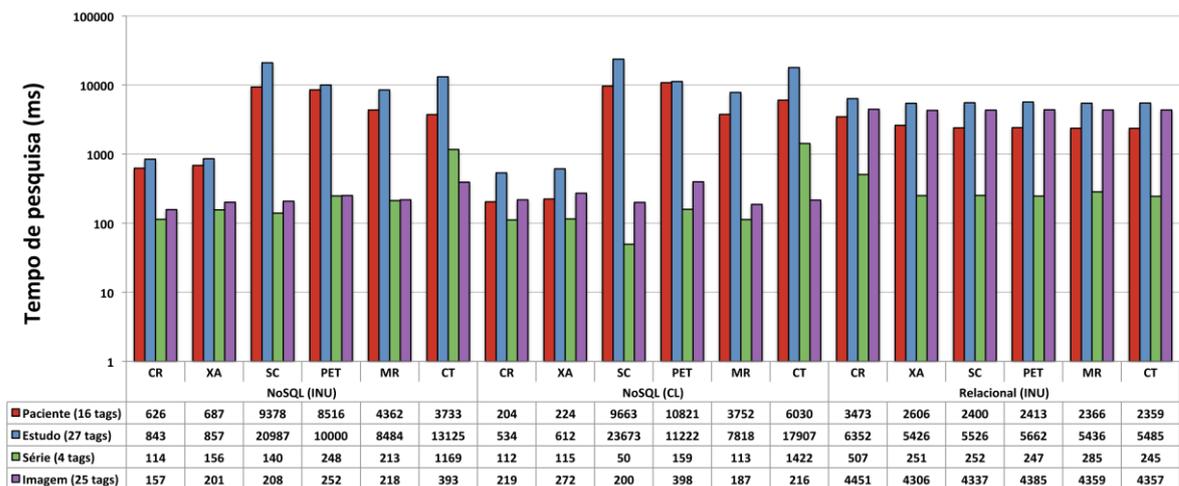


FIGURA 29 – TEMPO DE PESQUISA DE METADADOS POR MODALIDADE DE EXAME, RELACIONADA À DISTRIBUIÇÃO DE DADOS E NÍVEL HIERÁRQUICO. FONTE: o autor (2014)

A diferença de seletividade existente entre os níveis hierárquicos, de forma complementar ao número de *tags* pesquisadas por nível, impacta diretamente no tempo de resposta das operações de busca. Níveis mais baixos na hierarquia são beneficiados por demandarem predicados de busca mais completos, formados pelos

identificadores únicos dos níveis superiores em complemento ao identificador único do próprio nível. Essa particularidade resulta em situações nas quais a relevância do número de predicados é superior ao número de *tags* pesquisadas. Como exemplo, podem ser comparados os níveis *estudo* e *imagem*: apesar de possuírem um número de *tags* bastante parecido, o nível mais baixo (dada a maior seletividade de seus predicados) tem um tempo de resposta relacionado consideravelmente menor.

Visualmente, como pode ser observado na FIGURA 29, a instância relacional apresenta um padrão melhor definido com relação aos tempos de busca por nível hierárquico quando comparada às variantes NoSQL. Ela é, inclusive, mais lenta para algumas combinações de modalidade de exame e nível hierárquico; em números totais, porém, é a abordagem mais interessante: quando comparada às configurações de instância única e *cluster*, respectivamente, os ganhos são de 8,9% e 19,2%.

5.3.3 Resultados e discussão – recuperação (C-GET)

A recuperação do conteúdo completo das imagens DICOM armazenadas seguiu os procedimentos adotados na Seção 5.1.3: conjuntos de imagens para os níveis *estudo* e *série* foram acessados, bem como imagens individuais para o nível *imagem*. Novamente, 10 repetições foram executadas com predicados construídos a partir dos identificadores únicos de cada nível, em conjunto com os identificadores únicos dos níveis superiores.

Como esperado, o tempo de recuperação diminui conforme a seletividade dos níveis hierárquicos aumenta. Esse comportamento é observável em ambas as variantes NoSQL, para todas as modalidades de exame, com destaque para o *cluster*. Um dos fatores relevantes para o estabelecimento dessa relação está na estratégia de particionamento adotada (feita a partir do conteúdo da *tag patientid*); como essa *tag* é usada em todos os predicados de busca (por ser a identificação única do primeiro nível hierárquico), é possível à instância NoSQL rotear consultas para apenas um dos nodos que receberam cópias dos dados para o paciente especificado. Assim, não há a necessidade de consolidar resultados provenientes de múltiplas fontes, e execuções feitas em nodos específicos acessam apenas um subconjunto do volume total dos dados armazenados.

Com relação ao volume de conteúdo, quanto maiores as imagens recuperadas, mais tempo é gasto na execução das consultas. Esse fator impacta diretamente nos níveis hierárquicos intermediários (*estudo* e *série*), fazendo com que pesquisas executadas em ambos os níveis apresentem um tempo variado não apenas pelo número de imagens recuperadas, mas também pelo tamanho de cada imagem considerada individualmente.

Uma vez mais, a instância relacional se beneficia de não depender de resoluções de particionamento para encontrar e recuperar dados de imagens individuais ou conjuntos de imagens: apesar de ser mais lenta em casos específicos (como por exemplo, modalidade XA nos níveis *estudo* e *série*), na média geral é 81,7% e 83,2% mais rápida, respectivamente, que as variantes de instância única e *cluster* NoSQL. A relação entre os tempos que levam à essa conclusão pode ser visualizada na FIGURA 30.

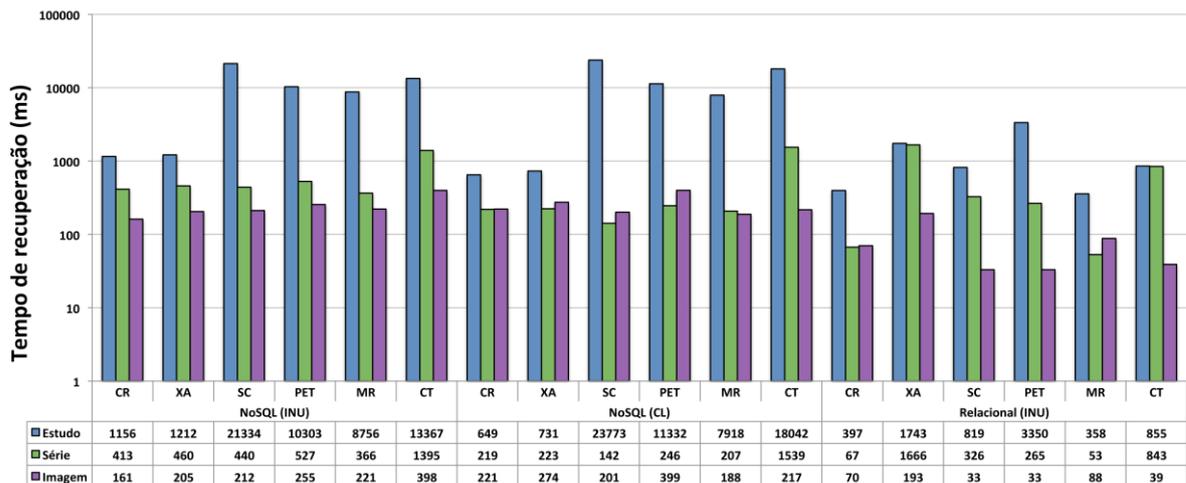


FIGURA 30 – TEMPO DE RECUPERAÇÃO DE CONTEÚDO POR MODALIDADE DE EXAME, RELACIONADA À DISTRIBUIÇÃO DE DADOS E NÍVEL HIERÁRQUICO. FONTE: o autor (2014)

5.4 CONSIDERAÇÕES SOBRE O CAPÍTULO

Os resultados apresentados neste capítulo indicam a viabilidade da adoção da arquitetura PACS proposta, considerando as hipóteses formuladas e os ganhos de desempenho nas operações de pesquisa de metadados e recuperação de conteúdo – operações essas as mais comumente executadas em cenários envolvendo

imagens médicas. Através de experimentos, foi possível verificar a adaptação do esquema de BD proposto a instâncias relacionais de nodo único e *clusters*, bem como de releituras desse esquema para uso com tecnologias não-relacionais. A adoção desse esquema em conjunto às extensões de SGBD implementadas, configuradas em cenários que provem escalabilidade e tolerância a falhas (aqui, *clusters* Postgres-XC e Cassandra), indicam que a arquitetura proposta é uma alternativa relevante ao estado da arte em sistemas PACS.

Apesar do senso comum de que instâncias de nodo único se sobressaem em termos de desempenho com relação a *clusters* (pelo fato de não necessitarem de comunicação internodos), alguns experimentos demonstram que, para determinados cenários e objetivos, a adoção de configurações distribuídas é a melhor opção. É o caso do cluster Postgres-XC (relacional) respondendo às operações de recuperação de conteúdo (C-GET) de forma mais rápida quando comparado à uma instância simples de BD. Os resultados obtidos indicam que, em situações onde se busca a recuperação de imagens completas, obtém-se um melhor desempenho com essas imagens sendo distribuídas em mais de uma instância de BD relacional, instâncias essas organizadas de forma particionada ou replicada. Considerando esse resultado em conjunto aos melhores desempenhos obtidos com as instâncias de nodo único em pesquisas de metadados (C-FIND), pode-se vislumbrar um cenário em que tecnologias diferentes são usadas para dados de complexidades diferentes; assim, metadados seriam mantidos em instâncias de BD simples (escolha justificada pelo desempenho e pelo volume reduzido dos dados a serem armazenados), enquanto que imagens completas seriam destinadas a *clusters* (devido à sua maior capacidade de armazenamento e escalabilidade, considerando volumes de dados mais expressivos).

O capítulo a seguir apresenta as conclusões derivadas dos resultados obtidos com os experimentos, detalhando-as em nível de hipótese e sugerindo possíveis atividades como trabalhos futuros.

6 CONCLUSÕES E TRABALHOS FUTUROS

Com base nos resultados apresentados e discutidos na Seção 5, este capítulo conclui o presente trabalho atestando a validade das hipóteses formuladas na Seção 1.3, bem como o atendimento aos objetivos específicos relacionados. A discussão é feita hipótese a hipótese, seguida de uma indicação das contribuições da tese à área de informática médica, especificamente no que compete ao gerenciamento de imagens digitais. Como complemento, são relacionados possíveis direcionamentos quanto à execução de trabalhos futuros.

6.1 HIPÓTESE #1 – ESQUEMA DE BD

O esquema de BD baseado no modelo de armazenamento decomposto descrito na Seção 4.2.3.1 validou a hipótese #1, provendo *guidelines* para a construção de esquemas físicos de BD aderentes às características do padrão DICOM. A separação de *tags* por VR em tabelas distintas atendeu à semiestruturação do padrão em nível de imagem, permitindo que elementos de dados conhecidos (integrantes do dicionário de dados DICOM) pudessem ser armazenados conjuntamente a elementos de dados desconhecidos (*tags* proprietárias), sem prejuízo de desempenho ou geração de demandas relacionadas à atualização da estrutura física dos esquemas de BD. O uso de campos comuns para armazenar o conteúdo dos componentes das *tags*, por sua vez, atendeu à estruturação do padrão em nível de elemento de dados. Por fim, o uso de uma tabela específica para persistir imagens completas mostrou-se adequado, considerando a facilidade obtida na recuperação de conteúdo e o seu desempenho para imagens individuais e conjuntos de imagens.

Com relação a desempenho, o esquema proposto mostrou-se apto a atender os cenários PACS mais comuns, onde a razão de uma escrita para várias leituras é bem conhecida. Mesmo quando conjuntos de *tags* têm seus valores recuperados simultaneamente, o desempenho do esquema foi superior às alternativas horizontalizadas, amplamente adotadas por servidores de armazenamento DICOM baseados no modelo relacional. Em conjunto com o suporte à heterogeneidade de conteúdo, o desempenho superior do esquema de BD proposto atendeu ao objetivo específico

relacionado à primeira hipótese, tornando-se uma alternativa viável quando SGBDs relacionais são usados como integrantes das camadas de persistência.

6.2 HIPÓTESE #2 – EXTENSÃO DE SGBD

Os mecanismos oferecidos pelos SGBDs escolhidos para os experimentos mostraram-se adequados como base à extensibilidade da camada de armazenamento da arquitetura PACS proposta, validando assim a hipótese #2 e atendendo ao objetivo específico relacionado. Dada a sua flexibilidade quanto ao código passível de ser executado sob a coordenação dos processos nativos dos SGBDs, foi possível implementar as primitivas mínimas necessárias ao estabelecimento de uma comunicação DICOM (atendendo aos quatro serviços básicos que envolvem armazenamento: verificação de conectividade, persistência, pesquisa e recuperação).

Uma particularidade não explicitada na descrição dos testes executados diz respeito à diferença na metodologia de acesso à instância de BD relacional da arquitetura proposta, quando comparada à arquitetura convencional. Na arquitetura convencional, pelo fato de o servidor DICOM ser um módulo à parte, sua conexão com o BD ocorre, invariavelmente, via *drivers* ODBC ou JDBC. Na arquitetura proposta, como o servidor DICOM é executado de forma incorporada ao SGBD, tais *drivers* são desnecessários; a conexão ao BD é persistente, e o acesso às tabelas é feito via chamadas da API de mais baixo nível fornecida pelo SGBD. A manutenção de uma conexão persistente e a eliminação dos *drivers* de conexão contribui para reduzir o tempo de acesso aos dados, sobretudo na pesquisa de metadados e recuperação de conteúdo – as operações executadas com mais frequência.

A agregação de serviços de comunicação DICOM ao SGBD pode gerar questionamentos com relação à quebra de modularização implementada pela arquitetura proposta. Essa agregação cria uma dependência dos serviços DICOM ao SGBD, de forma que se o último estiver indisponível, os serviços DICOM também não poderão ser executados. Funcionalmente, essa dependência já existe na arquitetura PACS convencional, visto que se a camada de persistência estiver indisponível, não é possível armazenar conteúdo, pesquisar metadados ou recuperar imagens para visualização ou manipulação. A principal diferença entre as arquiteturas reside no tipo de erro gerado, e em quem o reporta: na arquitetura convencional, o

erro é reportado pelo servidor DICOM, indicando a impossibilidade de acessar a camada de persistência; na arquitetura proposta, o erro ocorre na tentativa de associação entre o AE SCU e o AE SCP (nesse caso, a extensão do SGBD). Como a associação não é estabelecida, a comunicação não ocorre. Dadas essas similaridades funcionais entre arquiteturas, no contexto de armazenamento de conteúdo DICOM, a concentração de responsabilidades na camada de persistência não pode ser vista como um fator limitador e de propensão a erros.

6.3 HIPÓTESE #3 – ARMAZENAMENTO HETEROGÊNEO DE ALTO DESEMPENHO

A distribuição de dados convencionais e não convencionais foi viabilizada pelas implementações SQL/MED disponíveis nos SGBDs utilizados nos experimentos, validando assim a hipótese #3 e atendendo ao objetivo específico correlato. Essas implementações permitiram, de forma bastante direta, complementar o esquema de BD relacional com recursos de alto desempenho incluindo escalabilidade, particionamento, replicação e tolerância a falhas, através de sua interligação a uma tecnologia NoSQL. Como resultado, observou-se não apenas a flexibilização do armazenamento de dados em si, mas também ganhos de desempenho em operações de busca sem abrir mão das características do esquema de BD relacional proposto (como por exemplo, a capacidade de armazenar conteúdo heterogêneo).

Um benefício implícito da adoção de tecnologias heterogêneas interligadas via SQL/MED está na padronização da linguagem de acesso e manipulação de dados. O uso da linguagem SQL permite abstrair APIs de mais baixo nível, repassando a responsabilidade de tradução de consultas expressas em alto nível aos respectivos FDWs. Para as aplicações cliente (neste trabalho, as extensões de SGBD responsáveis pela escrita e leitura de dados), nada muda: o acesso continua sendo feito via esquema relacional, sobre as mesmas estruturas de tabela e campo; a persistência efetiva, porém, pode ser distribuída entre uma ou mais tecnologias, com estruturas físicas de armazenamento diversas.

Além de facilitar o acesso aos dados distribuídos, a adoção da extensão SQL/MED em conjunto a tecnologias NoSQL traz benefícios explícitos relacionados a desempenho e disponibilidade de recursos. Como exemplo, pode ser citado o am-

biente experimental #3 (Seção 5.3); nele, o *cluster* construído com Apache Cassandra permite particionar e replicar dados simultaneamente, o que é relevante em termos de alta disponibilidade e tolerância a falhas (considerando que a interrupção no funcionamento de um nodo não compromete o restante da estrutura). Em termos de escalabilidade, novos nodos podem ser adicionados sob demanda, sendo incorporados automaticamente pelo *cluster* e recebendo cópias de partições de dados de acordo com os critérios de distribuição definidos no momento de sua construção. Essas características complementam o modelo relacional, unindo a facilidade de acesso provida por uma linguagem de alto nível (SQL) com a segurança e os recursos de alto desempenho implementados por tecnologias diversas.

6.4 CONTRIBUIÇÕES

A principal contribuição do presente trabalho está na reorganização de arquiteturas PACS convencionais, feita com base na redistribuição de serviços e responsabilidades entre os seus componentes. Como resultado, obtém-se uma arquitetura mais enxuta, com um maior desempenho no acesso aos dados de interesse e uma maior flexibilidade de escolha quanto a opções de persistência de dados, sem comprometer o funcionamento de outras camadas de *hardware* e *software* envolvidas. Essa contribuição é baseada em estratégias pontuais que, individualmente, também apresentam novidades quanto à sua construção e/ou organização (como segue):

- desconhece-se, na literatura relacionada, uma abordagem semelhante à da utilização de um modelo de dados decomposto para o armazenamento de imagens no padrão DICOM. As pesquisas realizadas sobre esse tema possuem um foco quase que inteiramente voltado ao nível de imagem, tratando os seus elementos de dados componentes como estruturas a serem recuperadas, sob demanda de *parsing*, sempre que o conteúdo de um elemento ou conjunto de elementos é requisitado. A necessidade de se ter um acesso ágil a esses dados pontuais, utilizando-se de estratégias de busca flexíveis e contextualizadas, motivou o desenvolvimento da proposta de esquema de BD apresentada nesta tese, gerando como resultado um esquema diferenciado, robusto no armazenamento de conteúdo heterogêneo e rápido nas respostas a buscas por metadados e recuperação de conteúdo;

- apesar de bastante evoluídas, as APIs de extensibilidade disponibilizadas pelos SGBDs são relegadas, atualmente, a servirem de base para a construção de módulos de composição simplificada, usados geralmente na implementação de serviços de monitoramento. Nesta tese, essas APIs foram usadas em sua completude, de forma a prover os recursos necessários à integração de um servidor de armazenamento DICOM ao SGBD. Essa integração comprova que serviços baseados em persistência de dados, hoje sendo executados como componentes externos às suas respectivas camadas de armazenamento, podem ser refatorados como extensões a essas camadas, contribuindo assim para a simplificação das arquiteturas que englobam esses componentes e para o ganho de desempenho obtido através das alterações necessárias no acesso aos dados de interesse;
- as implementações da extensão SQL/MED oferecidas pelos SGBDs relacionais, desde a sua efetiva incorporação ao padrão SQL, têm sido pouco empregadas como *middleware* para distribuição de dados entre tecnologias heterogêneas, servindo mais como base teórica e de implementação para bancos de dados federados (homogeneamente construídos sobre o modelo relacional) (JOSIFOVSKI, 2002; ORDONEZ; CHEN; GARCÍA-GARCÍA, 2007). Nesta tese, o emprego da extensão é fundamentalmente heterogêneo, de forma a se beneficiar de características específicas encontradas em diferentes tecnologias (sem excluir, é claro, outros SGBDs). O principal diferencial, no entanto, não consiste em simplesmente empregar a extensão; o presente trabalho contribui por fornecer uma visão de uso da extensão SQL/MED onde tecnologias diferentes coexistem e se comunicam, de forma a atender a cenários onde os dados, com o passar do tempo, são migrados entre tecnologias de acordo com regras temporais e acessados de maneiras diversas. Em se tratando de PACS, essa característica é bem exemplificada pelos níveis de armazenamento e acesso *online*, *near-line* e *offline*. Cada um desses níveis pode ser atendido por uma ou mais ferramentas (*softwares*) para persistência, devidamente integradas para compor uma visão global do volume de dados disponível. Essa possibilidade de integração é o fator motivador principal para o uso de SQL/MED na arquitetura proposta.

O trabalho contribui, também com as seguintes publicações:

- (SAVARIS; WANGENHEIM, 2011) – relatório técnico com uma revisão sistemática de literatura, relacionando trabalhos que englobam o uso de SGBDs como meio para persistência de imagens médicas no padrão DICOM no contexto de sistemas PACS;
- (SAVARIS; HARA; WANGENHEIM, 2012) – trabalho que explora a modelagem e implementação de uma extensão de SGBD para acesso a dados armazenados em nuvem. A partir de sua execução, foi possível verificar a possibilidade de implementar extensões de SGBD de complexidade variada;
- (SAVARIS; HÄRDER; WANGENHEIM, 2014a) – trabalho que descreve o modelo de dados decomposto para DICOM; relaciona os resultados obtidos para as operações de armazenamento, pesquisa de metadados e recuperação de imagens completas;
- (SAVARIS; HÄRDER; WANGENHEIM, 2014b) – trabalho que apresenta os resultados obtidos com a adoção da extensão SQL/MED para a persistência de imagens DICOM em Apache Cassandra, correspondendo à extensão de SGBD para gerenciamento de distribuição de dados.

6.5 TRABALHOS FUTUROS

Como trabalhos futuros, sugere-se o aperfeiçoamento da arquitetura proposta pela incorporação de novos serviços de comunicação, processamento e armazenamento (como segue) (vide FIGURA 31):

- incluir rotinas de *parsing* e geração de mensagens de acordo com o padrão HL7 no módulo de gerenciamento de comunicação DICOM, transformando-o no *módulo de gerenciamento de comunicação DICOM/HL7*. Com essa alteração, o módulo passaria a receber e enviar dados de/para sistemas HIS e RIS não apenas em DICOM, mas também em HL7 (padrão de comunicação mais aderente aos tipos de dados suportados por esses sistemas);
- incorporar à arquitetura um *módulo para gerenciamento de processamento DICOM de alto desempenho*, responsável pela disponibilização de serviços de processamento baseados em GPGPU (*General-Purpose computation on Graphics Processing Units*) (GPGPU, 2014). Esse módulo seria utilizado,

sempre que necessário, pelos módulos de gerenciamento de comunicação e armazenamento/distribuição de dados, provendo rotinas para a compressão de imagens e reconstrução 3D (tridimensional) a partir de séries de imagens dos exames armazenados;

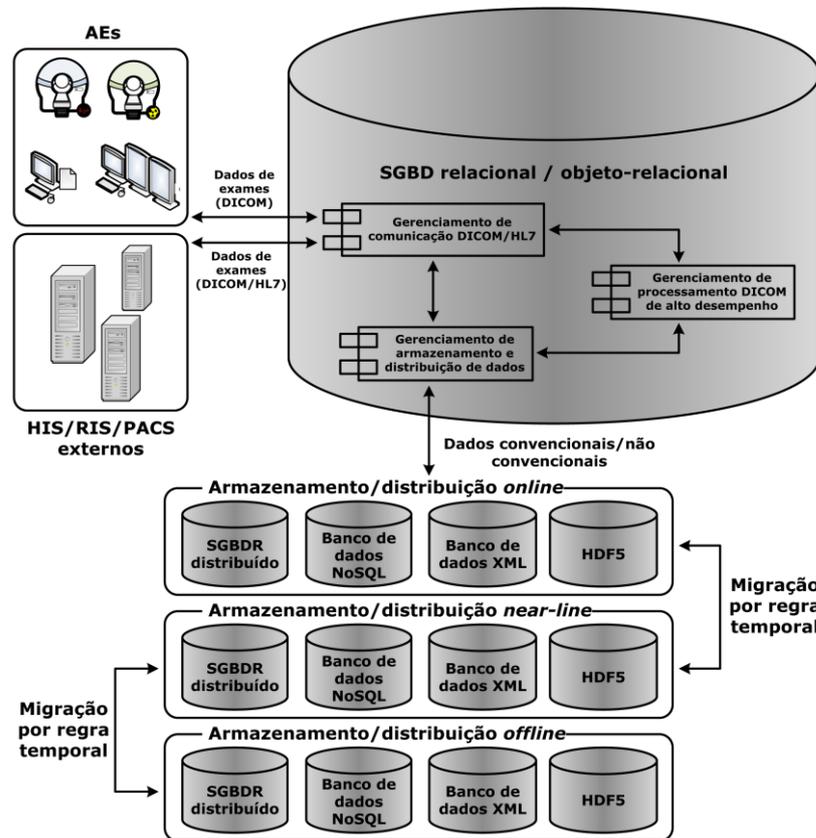


FIGURA 31 – TRABALHOS FUTUROS INTEGRADOS À PROPOSTA DE ARQUITETURA PACS. DESTAQUE PARA A INCORPORAÇÃO DE SERVIÇOS DE PROCESSAMENTO DE ALTO DESEMPENHO E PARA A DISTRIBUIÇÃO TEMPORAL DE DADOS *ONLINE*, *NEAR-LINE* E *OFFLINE*. FONTE: o autor (2014)

- incluir rotinas para controle temporal do conteúdo gerido pelo módulo de gerenciamento de armazenamento e distribuição de dados, permitindo ao mesmo rotear os dados de interesse para diferentes tecnologias de persistência com base em regras derivadas da legislação vigente, das políticas adotadas pela instituição usuária dos serviços com relação à manutenção das imagens DICOM adquiridas e da disponibilidade de recursos de *hardware* e *software*;
- complementar o módulo de gerenciamento de armazenamento e distribuição de dados com bibliotecas de indexação e busca externas, tais como o Apache Lucene (APACHE LUCENE CORE, 2014). Essas bibliotecas flexibilizariam a

construção de instruções de busca, disponibilizando operadores específicos que seriam úteis em determinadas situações (tais como buscas por correspondência de padrão), auxiliando na redução do tempo necessário à localização de metadados individuais ou grupos de metadados, além de imagens individuais ou grupos de imagens pertencentes a estudos DICOM completos.

REFERÊNCIAS

ABADI, D. Column Stores for Wide and Sparse Data. In: CIDR 2007, Pacific Grove, California, USA. **Proceedings of the Third Biennial Conference on Innovative Data Systems Research**. 2007. p. 292-297.

ABBASI, H. *et al.* DataStager: Scalable Data Staging Services for Petascale Applications. In: HPDC '09, Munich, Germany. **Proceedings of the 18th ACM International Symposium on High Performance Distributed Computing**. New York, NY, USA: ACM, 2009. p. 39-48.

ACUÑA, C. J. *et al.* A Web Information System for Medical Image Management. **Lecture Notes in Computer Science**, v. 3337, p. 49-59, 2004.

AGOSTINHO, R.; PEREIRA, M.; FREIRE, M. Advanced Querying Architecture for DICOM Systems. In: ICSNC '07, Cap Esterel, France. **Proceedings of the Second International Conference on Systems and Networks Communications**. Washington, DC, USA: IEEE Computer Society, 2007. p. 53-56.

AHMED, S. Grid Services Architecture for Archiving and Presentation of Medical Images. In: HONET 2007, Dubai, United Arab Emirates. **International Symposium on High Capacity Optical Networks and Enabling Technologies**. 2007. p. 1-8.

ALLCOCK, B. *et al.* Data Management and Transfer in High-Performance Computational Grid Environments. **Parallel Computing – Parallel Data Intensive Algorithms and Applications**, v. 28, n. 5, p. 749-771, 2002.

APACHE LUCENE CORE. Disponível em: <<http://lucene.apache.org/core/>>. Acesso em: 01/08/2014.

APACHE THRIFT. Disponível em: <<http://thrift.apache.org/>>. Acesso em: 04/08/2014.

ARMBRUST, L. J. PACS and Image Storage. **Veterinary Clinics of North America: Small Animal Practice**, v. 39, n. 4, p. 711-718, 2009.

BATORY, D. S. Principles of Database Management System Extensibility. **IEEE Database Engineering**, v. 10, n. 2, p. 40-46, 1987.

BATORY, D. S. *et al.* GENESIS: An Extensible Database Management System. **IEEE Transactions on Software Engineering**, v. 14, n. 11, p. 1711-1730, 1988.

BECKMANN, J. L. *et al.* Extending RDBMSs to support sparse datasets using an interpreted attribute storage format. In: ICDE 2006, Atlanta, GA, USA. **Proceedings of the 22nd International Conference on Data Engineering**. Washington, DC, USA: IEEE Computer Society, 2006. p. 58-67.

BELL, G.; GRAY, J. What's Next in High-Performance Computing? **Communications of the ACM**, v. 45, n. 2, p. 91-95, 2002.

BELLON, E. *et al.* Trends in PACS architecture. **European Journal of Radiology**, v. 78, n. 2, p. 199-204, 2011.

BHATTACHARYA, S. *et al.* Coordinating backup/recovery and data consistency between database and file systems. In: SIGMOD '02, Madison, Wisconsin, USA. **Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data**. New York, NY, USA: ACM, 2002. p. 500-511.

BIAN, J.; SEKER, R.; TOPALOGLU, U. A Secure Distributed File System for Medical Image Archiving. In: SOCIALCOM '10, Minneapolis, MN, USA. **Proceedings of the 2010 IEEE Second International Conference on Social Computing**. Washington, DC, USA: IEEE Computer Society, 2010. p. 961-966.

BIDGOOD JR, W. D. *et al.* Understanding and Using DICOM, the Data Interchange Standard for Biomedical Imaging. **Journal of the American Medical Informatics Association**, v. 4, n. 3, p. 199-212, 1997.

BLANQUER, I.; HERNÁNDEZ, V.; SEGRELLES, D. Creating Virtual Storages and Searching DICOM Medical Images Through a Grid Middleware based in OGSA. In: CCGRID 2005, Cardiff, UK. **IEEE International Symposium on Cluster Computing and the Grid**. Washington, DC, USA: IEEE Computer Society, 2005. p. 504-511.

BONDI, A. B. Characteristics of Scalability and Their Impact on Performance. In: WOSP '00, Ottawa, Canada. **Proceedings of the 2nd International Workshop on Software and Performance**. New York, NY, USA: ACM, 2000. p. 195-203.

BOSILCA, G. *et al.* Algorithm-based Fault Tolerance Applied to High Performance Computing. **Journal of Parallel and Distributed Computing**, v. 69, n. 4, p. 410-416, 2009.

BOURNE, R. **Fundamentals of Digital Imaging in Medicine**. London, UK: Springer-Verlag, 2010. p. 31-54.

BROWN, M. S. *et al.* Database Design and Implementation for Quantitative Image Analysis Research. **IEEE Transactions on Information Technology in Biomedicine**, v. 9, n. 1, p. 99-108, 2005.

CAICEDO, J. C. *et al.* Design of a Medical Image Database With Content-based Retrieval Capabilities. In: PSIVT '07, Santiago, Chile. **Proceedings of the 2nd Pacific Rim Conference on Advances in Image and Video Technology**. Berlin, Heidelberg: Springer-Verlag, 2007. p. 919-931.

CAPPELLO, F. Fault Tolerance in Petascale/Exascale Systems: Current Knowledge, Challenges and Research Opportunities. **International Journal of High Performance Computing Applications**, v. 23, n. 3, p. 212-226, 2009.

CAREY, M. J. *et al.* The architecture of the EXODUS extensible DBMS. In: OODS '86, Pacific Grove, California, USA. **Proceedings on the 1986 International Workshop on Object-oriented Database Systems**. Los Alamitos, CA, USA: IEEE Computer Society Press, 1986. p. 52-65.

CAREY, M.; HAAS, L. Extensible Database Management Systems. **SIGMOD Record**, v. 19, n. 4, p. 54-60, 1990.

CASSANDRA. Disponível em: <<http://cassandra.apache.org/>>. Acesso em: 04/08/2014.

CHANDRASHEKAR, N. *et al.* COTS-Like Generic Medical Image Repository. In: ICCBSS '06, Orlando, Florida, USA. **Proceedings of the Fifth International Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems**. Washington, DC, USA: IEEE Computer Society, 2006. p. 199-205.

CODD, E. F. A Relational Model of Data for Large Shared Data Banks. **Communications of the ACM**, v. 13, n. 6, p. 377-387, 1970.

COPELAND, G. P.; KHOSHAFIAN, S. N. A decomposition storage model. **SIGMOD Record**, v. 14, n. 4, p. 268-279, 1985.

CORRIERO, N. *et al.* HSFS: A Compress Filesystem for Metadata Files. **Digital Information Processing and Communications**, v. 189, p. 289-300, 2011.

COSTA, C.; SILVA, A.; OLIVEIRA, J. L. Current Perspectives on PACS and a Cardiology Case Study. **Studies in Computational Intelligence**, v. 65, p. 79-108, 2007.

COWEN, A. R.; DAVIES, A. G.; KENGYELICS, S. M. Advances in Computed Radiography Systems and their Physical Imaging Characteristics. **Clinical Radiology**, v. 62, n. 12, p. 1132-1141, 2007.

DANTAS, M. **Computação Distribuída de Alto Desempenho – Redes, Clusters e Grids Computacionais**. Rio de Janeiro, RJ, Brazil: Axcel Books do Brasil Editora, 2005. p. 1-59.

DCM4CHE. Disponível em: <<http://www.dcm4che.org/confluence/display/ee2/Home>>. Acesso em: 04/08/2014.

DICOM. Disponível em: <<http://medical.nema.org/>>. Acesso em: 11/07/2012.

DICOM sample image sets – OsiriX Imaging Software Web site. Disponível em: <<http://www.osirix-viewer.com/datasets/>>. Acesso em: 15/05/2014.

DOLIN, R. H. *et al.* The HL7 Clinical Document Architecture. **Journal of the American Medical Informatics Association**, v. 8, n. 6, p. 552-569, 2001.

DOLIN, R. H. *et al.* HL7 Clinical Document Architecture, Release 2. **Journal of the American Medical Informatics Association**, v. 13, n. 1, p. 30-39, 2006.

DONGARRA, J. *et al.* High-Performance Computing: Clusters, Constellations, MPPs, and Future Directions. **Computing in Science & Engineering**, v. 7, n. 2, p. 51-59, 2005.

DONGARRA, J. Trends in High-Performance Computing: a Historical Overview and Examination of Future Developments. **IEEE Circuits and Devices Magazine**, v. 22, n. 1, p. 22-27, 2006.

DUBOC, L.; ROSENBLUM, D. S.; WICKS, T. A Framework for Modelling and Analysis of Software Systems Scalability. In: ICSE '06, Shanghai, China. **Proceedings of the 28th International Conference on Software Engineering**. New York, NY, USA: ACM, 2006. p. 949-952.

EICHELBERG, M.; ADEN, T.; RIESMEIER, J. A Survey and Analysis of Electronic Healthcare Record Standards. **ACM Computing Surveys**, v. 37, n. 4, p. 277-315, 2005.

EVANGELISTA, N. D.; CAMAPUM, J. F.; AMEMIYA, E. Communication and Storage of Digital Medical Images in Database. In: IEEE-EMBS 2005, Shanghai, China. **27th Annual International Conference of the Engineering in Medicine and Biology Society**. 2005. p. 5471-5474.

FADIKA, Z. *et al.* MARIANE: Using MapReduce in HPC Environments. **Future Generation Computer Systems**, v. 36, p. 379-388, 2014.

FAGG, G. E. Process Fault Tolerance: Semantics, Design and Applications for High Performance Computing. **International Journal of High Performance Computing Applications**, v. 19, n. 4, p. 465-477, 2005.

FU, S.; XU, C.-Z. Proactive Resource Management for Failure Resilient High Performance Computing Clusters. In: ARES '09, Fukuoka, Japan. **International Conference on Availability, Reliability and Security**. Washington, DC, USA: IEEE Computer Society, 2009. p. 257-264.

GALE, M. E.; GALE, D. R. DICOM Modality Worklist: An Essential Component in a PACS Environment. **Journal of Digital Imaging**, v. 13, n. 3, p. 101-108, 2000.

GAO, X.; NACHANKAR, V.; QIU, J. Experimenting Lucene Index on HBase in an HPC Environment. In: HPCDB '11, Seattle, WA, USA. **Proceedings of the first Annual Workshop on High Performance Computing Meets Databases**. New York, NY, USA: ACM, 2011. p. 25-28.

GARDARIN, G. *et al.* Extending a Relational DBMS to Support Complex Objects. In: DKSME 1989, Gaithersburg, MD, USA. **Proceedings of the Second International Conference on Data and Knowledge Systems for Manufacturing and Engineering**. Los Alamitos, CA, USA: IEEE Computer Society Press, 1989. p. 131-137.

GODFREY, M. *et al.* Secure and Portable Database Extensibility. In: SIGMOD '98, Seattle, Washington, USA. **Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data**. New York, NY, USA: ACM, 1998. p. 390-401.

GOLUBCHIK, S.; HUTCHINGS, A. Daemon Plugins. In: _____. **MySQL 5.1 Plugin Development**. Birmingham, UK: Packt Publishing Ltd., 2010. p. 51-81.

GOLUBCHIK, S.; HUTCHINGS, A. Storage Engine Plugins. In: _____. **MySQL 5.1 Plugin Development**. Birmingham, UK: Packt Publishing Ltd., 2010. p. 161-179.

GPGPU. Disponível em: <<http://gpgpu.org/>>. Acesso em: 04/08/2014.

GROSSMAN, R. L. *et al.* Compute and Storage Clouds Using Wide Area High Performance Networks. **Future Generation Computer Systems**, v. 25, n. 2, p. 179-183, 2009.

HABABEH, I. O.; RAMACHANDRAN, M.; BOWRING, N. A High-Performance Computing Method for Data Allocation in Distributed Database Systems. **The Journal of Supercomputing**, v. 39, n. 1, p. 3-18, 2007.

HASKIN, R. L.; LORIE, R. A. On extending the functions of a relational database system. In: SIGMOD '82, Orlando, Florida, USA. **Proceedings of the 1982 ACM SIGMOD International Conference on Management of Data**. New York, NY, USA: ACM, 1982. p. 207-212.

HEALTH LEVEL SEVEN INTERNATIONAL. Disponível em: <<http://www.hl7.org/>>. Acesso em: 11/07/2012.

HECKMAN, K.; SCHULTZ, T. J. PACS Architecture. In: DREYER, K. J. *et al.* **PACS – A Guide to the Digital Revolution**. New York, NY: Springer Science+Business Media, Inc, 2006. p. 249-267.

HILDEBRAND, D.; HONEYMAN, P. Exporting Storage Systems in a Scalable Manner with pNFS. In: MSST '05, Monterey, CA, USA. **Proceedings of the 22nd IEEE / 13th NASA Goddard Conference on Mass Storage Systems and Technologies**, 2005. p. 18-27.

HILL, M. D. What is Scalability? **ACM SIGARCH Computer Architecture News**, v. 18, n. 4, p. 18-21, 1990.

HONEYMAN-BUCK, J. PACS Adoption. **Seminars in Roentgenology**, v. 38, n. 3, p. 256-269, 2003.

HU, M. *et al.* Informatics in Radiology: Efficiency Metrics for Imaging Device Productivity. **Radiographics**, v. 31, n. 2, p. 603-616, 2011.

HUANG, H. K. *et al.* Data Grid for Large-scale Medical Image Archive and Analysis. In: MULTIMEDIA '05, Hilton, Singapore. **Proceedings of the 13th annual ACM international conference on Multimedia**. New York, NY, USA: ACM, 2005. p. 1005-1013.

- HUANG, H. K.; LIU, B. J.; ZHOU, Z. A Continuous Available (CA) Server for Medical Imaging Applications. **Academic Radiology**, v. 11, n. 7, p. 767-778, 2004.
- HUMBETOV, S. Data-Intensive Computing with Map-Reduce and Hadoop. In: AICT 2012, Tbilisi, Georgia. **6th International Conference on Application of Information and Communication Technologies**. Washington, DC, USA: IEEE Computer Society, 2012. p. 1-5.
- IGWE, S. O.; ELMAGHRABY, A. S. Medical Image Storage System for Content-based Retrieval. In: ISSPIT, Luxor, Egypt. **2010 IEEE International Symposium on Signal Processing and Information Technology**. Washington, DC, USA: IEEE Computer Society, 2010. p. 520-527.
- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO). **ISO/IEC 9075-9:2003**: Management of External Data (SQL/MED), 2003.
- IVANOVA, M.; KERSTEN, M.; MANEGOLD, S. Data Vaults: A Symbiosis between Database Technology and Scientific File Repositories. In: SSDBM '12, Chania, Crete, Greece. **Proceedings of the 24th International Conference on Scientific and Statistical Database Management**. Berlin Heidelberg, Germany: Springer, 2012. p. 485-494.
- JERONIMO, J. *et al.* A Tool for Collection of Region Based Data From Uterine Cervix Images for Correlation of Visual and Clinical Variables Related to Cervical Neoplasia. In: CBMS 2004, Bethesda, MD, USA. **Proceedings of the 17th IEEE Symposium on Computer-based Medical Systems**. Washington, DC, USA: IEEE Computer Society, 2004. p. 558-562.
- JOSIFOVSKI, V. *et al.* Garlic: a New Flavor of Federated Query Processing for DB2. In: SIGMOD '02, Madison, Wisconsin, USA. **Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data**. New York, NY, USA: ACM, 2002. p. 524-532.
- KITCHENHAM, B. **Procedures for Performing Systematic Reviews**. Keele University Technical Report TR/SE-0401, Department of Computer Science, Software Engineering Group, 2004. NICTA Technical Report 0400011T.1, Empirical Software Engineering, 2004. 33 p. Relatório técnico conjunto.
- KOBLITZ, B.; SANTOS, N.; POSE, V. The AMGA Metadata Service. **Journal of Grid Computing**, v. 6, n. 1, p. 61-76, 2008.
- KORENBLUM, D. *et al.* Managing Biomedical Image Metadata for Search and Retrieval of Similar Images. **Journal of Digital Imaging**, v. 24, n. 4, p. 739-748, 2011.
- LAN, Z.; LI, Y. Adaptive Fault Management of Parallel Applications for High-Performance Computing. **IEEE Transactions on Computers**, v. 57, n. 12, p. 1647-1660, 2008.

LANGER, S. G. Challenges for Data Storage in Medical Imaging Research. **Journal of Digital Imaging**, v. 24, n. 2, p. 203-207, 2011.

LAPRIE, J.-C. Dependable Computing and Fault Tolerance: Concepts and Terminology. In: FTCS-15, Ann Arbor, Michigan, USA. **Proceedings of the 15th IEEE International Symposium on Fault Tolerant Computing**. Washington, DC, USA: IEEE Computer Society, 1985. p. 2-11.

LAWRENCE, R. Integration and Virtualization of Relational SQL and NoSQL Systems including MySQL and MongoDB. In: CSCI '14, Las Vegas, NV, USA. **Proceedings of the 2014 International Conference on Computational Science and Computational Intelligence**. Los Alamitos, CA, USA: IEEE Computer Society Press, 2014. p. 285-290.

LI, J. *et al.* Design and Development of an International Clinical Data Exchange System: the International Layer Function of the Dolphin Project. **Journal of the American Medical Informatics Association**, v. 18, n. 5, p. 683-689, 2011.

LI, Y. *et al.* Fault-Aware Runtime Strategies for High-Performance Computing. **IEEE Transactions on Parallel and Distributed Systems**, v. 20, n. 4, p. 460-473, 2009.

LINNEMANN, V. *et al.* Design and Implementation of an Extensible Database Management System Supporting User Defined Data Types and Functions. In: VLDB '88, Los Angeles, California, USA. **Proceedings of the 14th International Conference on Very Large Data Bases**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1988. p. 294-305.

LITH, M.; MATTSSON, J. **Investigating storage solutions for large data**. 70 p. Thesis (Master of Science) – Department of Computer Science and Engineering, Chalmers University of Technology, Göteborg, Sweden, 2010.

LIU, B. J.; ZHOU, M. Z.; DOCUMENT, J. Utilizing Data Grid Architecture for the Backup and Recovery of Clinical Image Data. **Computerized Medical Imaging and Graphics**, v. 29, n. 2, p. 95-102, 2005.

LOMBILLO-BIOSCA, P. *et al.* SAGIMA: An Easy-to-use and Low Cost WEB-PACS System for an Optimal Access and Management of a Digital Angiography Database. In: 32nd Annual Conference on Computers in Cardiology, Lyon, France. **Computers in Cardiology**. 2005. p. 913-916.

MACEDO, D. D. J. *et al.* An architecture for DICOM medical images storage and retrieval adopting distributed file systems. **International Journal of High Performance Systems Architecture**, v. 2, n. 2, p. 99-106, 2009.

MACEDO, D. D. J. *et al.* An Improvement of a Different Approach for Medical Image Storage. In: WETICE, Paris, France. **2011 20th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises**. Washington, DC, USA: IEEE Computer Society, 2011. p. 140-142.

MAGNUS, M. *et al.* A Study of NetCDF as an Approach for High Performance Medical Image Storage. **Journal of Physics: Conference Series**, 341, 2012.

MANSOORI, B.; ERHARD, K. K.; SUNSHINE, J. L. Picture Archiving and Communication System (PACS) Implementation, Integration & Benefits in an Integrated Health System. **Academic Radiology**, v. 19, n. 2, p. 229-235, 2012.

MARIADB. Disponível em: <<https://mariadb.org/pt-br/>>. Acesso em: 04/08/2014.

MARIADB – Cassandra storage engine overview. Disponível em: <<https://mariadb.com/kb/en/mariadb/mariadb-documentation/mariadb-storage-engines/storage-engines-cassandra-storage-engine/cassandra-storage-engine-overview/>>. Acesso em: 04/08/2014.

MELTON, J. *et al.* SQL and management of external data. **SIGMOD Record**, v. 30, n. 1, p. 70-77, 2001.

MELTON, J. *et al.* SQL/MED: a status report. **SIGMOD Record**, v. 31, n. 3, p. 81-89, 2002.

MEO, P. D.; QUATTRONE, G.; URSINO, D. Integration of the HL7 Standard in a Multiagent System to Support Personalized Access to e-Health Services. **IEEE Transactions on Knowledge and Data Engineering**, v. 23, n. 8, p. 1244-1260, 2011.

MILDENBERGER, P.; EICHELBERG, M.; MARTIN, E. Introduction to the DICOM standard. **European Radiology**, v. 12, n. 4, p. 920-927, 2002.

MINATI, L. *et al.* Bio-Image Warehouse System: Concept and Implementation of a Diagnosis-Based Data Warehouse for Advanced Imaging Modalities in Neuroradiology. **Journal of Digital Imaging**, v. 20, n. 1, p. 32-41, 2007.

MITTAL, N.; HSIAO, HUI-L. Database managed external file update. In: **Proceedings of the 17th International Conference on Data Engineering**. 2001. p. 557-564.

MONTAGNAT, J. *et al.* A Secure Grid Medical Data Manager Interfaced to the gLite Middleware. **Journal of Grid Computing**, v. 6, n. 1, p. 45-59, 2008.

MOUGIAKAKOU, S. G. *et al.* DIAGNOSIS: A Telematics-Enabled System for Medical Image Archiving, Management and Diagnosis Assistance. **IEEE Transactions on Instrumentation and Measurement**, v. 58, n. 7, p. 2113-2120, 2009.

MYSQL. Disponível em: <<http://www.mysql.com/>>. Acesso em: 04/08/2014.

MYSQL – Overview of MySQL Storage Engine Architecture. Disponível em: <<http://dev.mysql.com/doc/refman/5.6/en/pluggable-storage-overview.html>>. Acesso em: 12/08/2014.

NAGY, P. G.; SCHULTZ, T. J. Storage and Enterprise Archiving. In: DREYER, K. J. *et al.* **PACS – A Guide to the Digital Revolution**. New York, NY: Springer Science+Business Media, Inc, 2006. p. 319-345.

NATIONAL BIOMEDICAL IMAGING ARCHIVE – National Cancer Institute Web site. Disponível em: <<https://imaging.nci.nih.gov/ncia/>>. Acesso em: 18/07/2013.

NOSQL. Disponível em: <<http://nosql-database.org/>>. Acesso em: 26/07/2012.

OGIELA, M. R.; TADEUSIEWICZ, R. Picture Languages in Intelligent Retrieval of Visual Data Semantic Information. **Lecture Notes in Computer Science**, v. 3336, p. 389-396, 2004.

OOIJEN, P. M. A. v.; BHOMER, P. J. M. t.; OUDKERK, M. PACS Storage Requirements-influence of Changes in Imaging Modalities. **International Congress Series**, v. 1281, p. 888-893, 2005.

ORDONEZ, C.; CHEN, Z.; GARCÍA-GARCÍA, J. Metadata Management for Federated Databases. In: CIMS '07, Lisbon, Portugal. **Proceedings of the ACM First Workshop on CyberInfrastructure: Information Management in eScience**. New York, NY, USA: ACM, 2007. p. 31-38.

ORDONEZ, C. Statistical Model Computation with UDFs. **IEEE Transactions on Knowledge and Data Engineering**, v. 22, n. 12, p. 1752-1765, 2010.

PACHEV, S. The Storage Engine Interface. In: _____. **Understanding MySQL Internals**. Sebastopol, CA, USA: O'Reilly Media, Inc., 2007. p. 119-160.

PAPIANI, M. *et al.* A distributed scientific data archive using the Web, XML and SQL/MES. **SIGMOD Record**, v. 28, n. 3, p. 56-62, 1999.

PAPIANI, M.; WASON, J. L.; NICOLE, D. A. An Architecture for Archiving and Post-Processing Large, Distributed, Scientific Data Using SQL/MED and XML. In: EDBT 2000, Konstanz, Baden-Württemberg, Germany. **Proceedings of the 7th International Conference on Extending Database Technology**. Berlin Heidelberg, Germany: Springer, 2000. p. 447-461.

PARÉ, G.; TRUDEL, M-C. Knowledge barriers to PACS adoption and implementation in hospitals. **International Journal of Medical Informatics**, v. 76, n. 1, p. 22-33, 2007.

PARR, L. F. *et al.* Trials and Tribulations in Deploying Digital Imaging Network and Picture Archiving and Communication System – Film to Filmless in 80 Days. **Journal of Digital Imaging**, v. 14, n. 2, p. 167-170, 2001.

PATANACHAI, N. *et al.* PACS (Picture Archiving Communication System) for Dentistry. In: ECTI-CON 2008, Krabi, Thailand. **5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology**. Washington, DC, USA: IEEE Computer Society, 2008. p. 77-80.

PIANYKH, O. S. **Digital Imaging and Communications in Medicine (DICOM): A Practical Introduction and Survival Guide**. Leipzig, Germany: Springer Berlin Heidelberg, 2008. p. 112-217.

POSTGRES-XC. Disponível em: <<https://wiki.postgresql.org/wiki/Postgres-XC>>. Acesso em: 04/08/2014.

POSTGRESQL. Disponível em: <<http://www.postgresql.org/>>. Acesso em: 07/05/2014.

POSTGRESQL – Chapter 45. Background Worker Processes. Disponível em: <<http://www.postgresql.org/docs/9.3/static/bgworker.html>>. Acesso em: 12/05/2014.

POSTGRESQL – Foreign data wrappers. Disponível em: <http://wiki.postgresql.org/wiki/Foreign_data_wrappers>. Acesso em: 15/05/2014.

POWER, D. *et al.* A relational approach to the capture of DICOM files for Grid-enabled medical imaging databases. In: SAC '04, Nicosia, Cyprus, Greece. **Proceedings of the 2004 ACM Symposium on Applied Computing**. New York, NY, USA: ACM, 2004. p. 272-279.

PRATHIBHA, D. A.; LATHA, B.; SUMATHI, G. Issues in Adapting Cluster, Grid and Cloud Computing for HPC Applications. **International Journal of Conceptions on Computing and Information Technology**, v. 2, n. 1, p. 12-16, 2014.

PS 3.2-2011 – Digital Imaging and Communications in Medicine (DICOM) – Part 2: Conformance. Disponível em: <ftp://medical.nema.org/medical/dicom/2011/11_02pu.pdf>. Acesso em: 12/07/2012.

PS 3.4-2011 – Digital Imaging and Communications in Medicine (DICOM) – Part 4: Service Class Specifications. Disponível em: <ftp://medical.nema.org/medical/dicom/2011/11_04pu.pdf>. Acesso em: 12/07/2012.

PS 3.5-2011 – Digital Imaging and Communications in Medicine (DICOM) – Part 5: Data Structures and Encoding. Disponível em: <ftp://medical.nema.org/medical/dicom/2011/11_05pu.pdf>. Acesso em: 12/07/2012.

PS 3.6-2011 – Digital Imaging and Communications in Medicine (DICOM) – Part 6: Data Dictionary. Disponível em: <ftp://medical.nema.org/medical/dicom/2011/11_06pu.pdf>. Acesso em: 12/07/2012.

PS 3.10-2011 – Digital Imaging and Communications in Medicine (DICOM) – Part 10: Media Storage and File Format for Media Interchange. Disponível em: <ftp://medical.nema.org/medical/dicom/2011/11_10pu.pdf>. Acesso em: 12/07/2012.

PS 3.11-2011 – Digital Imaging and Communications in Medicine (DICOM) – Part 11: Media Storage Application Profiles. Disponível em: <ftp://medical.nema.org/medical/dicom/2011/11_11pu.pdf>. Acesso em: 12/07/2012.

PS 3.12-2011 – Digital Imaging and Communications in Medicine (DICOM) – Part 12: Media Formats and Physical Media for Media Interchange. Disponível em: ftp://medical.nema.org/medical/dicom/2011/11_12pu.pdf. Acesso em: 12/07/2012.

PUTTEN, N. v. D. *et al.* THOPACS : The Multi-modality, Image Review Diagnosis, Archiving and Analysis System. **Computers in Cardiology**, v. 35, p. 291-294, 2008.

REINER, B. New Strategies for Medical Data Mining, Part 3: Automated Workflow Analysis and Optimization. **Journal of the American College of Radiology**, v. 8, n. 2, p. 132-138, 2011.

ROSSET, C.; ROSSET, A.; RATIB, O. General Consumer Communication Tools for Improved Image Management and Communication in Medicine. **Journal of Digital Imaging**, v. 18, n. 4, p. 270-279, 2005.

SADASHIV, N.; KUMAR, S. M. D. Cluster, Grid and Cloud Computing: A Detailed Comparison. In: ICCSE 2011, SuperStar Virgo, Singapore. **The 6th International Conference on Computer Science & Education**. Washington, DC, USA: IEEE Computer Society, 2011. p. 477-482.

SAVARIS, A.; WANGENHEIM, A. v. **Armazenamento de Imagens Médicas no Padrão DICOM em Banco de Dados no Contexto de PACS**. INCoD-11.006.P.01.TELEMED, 2011. 24 p. Relatório técnico.

SAVARIS, A.; HARA, C. S.; WANGENHEIM, A. v. MyGFT - um Módulo de Integração entre MySQL e Google Fusion Tables. In: SBBD 2012, São Paulo, SP, Brazil. **Proceedings of the 27th Brazilian Symposium on Databases – Demos and Applications Session**, 2012. p. 31-36.

SAVARIS, A.; HÄRDER, T.; WANGENHEIM, A. v. DCMDSM: a DICOM decomposed storage model. **Journal of the American Medical Informatics Association**, v. 21, n. 5, p. 917-924, 2014.

SAVARIS, A.; HÄRDER, T.; WANGENHEIM, A. v. Evaluating a row-store data model for full-content DICOM management. In: CBMS 2014, New York, NY, USA. **Proceedings of the 27th IEEE Symposium on Computer-based Medical Systems**. Washington, DC, USA: IEEE Computer Society, 2014. p. 193-198.

SCHWARZ, P. *et al.* Extensibility in the Starburst database system. In: OODS '86, Pacific Grove, California, USA. **Proceedings on the 1986 International Workshop on Object-oriented Database Systems**. Los Alamitos, CA, USA: IEEE Computer Society Press, 1986. p. 85-92.

SELVER, M. A. *et al.* A Software Tool for Interactive Generation, Representation, and Systematical Storage of Transfer Functions for 3D Medical Images. **Computer Methods and Programs in Biomedicine**, v. 86, n. 3, p. 270-280, 2007.

SLAVICEK, K. *et al.* MEDIMED - Regional Centre for Medicine Image Data Processing. In: WKDD '10, Phuket, Thailand. **Third International Conference on Knowledge Discovery and Data Mining**. Washington, DC, USA: IEEE Computer Society, 2010. p. 310-313.

SMITH, G. Introduction to RIS and PACS. In: DREYER, K. J. *et al.* **PACS – A Guide to the Digital Revolution**. New York, NY: Springer Science+Business Media, Inc, 2006. p. 9-25.

SOARES, T. S. *et al.* An Approach Using Parallel Architecture to Storage DICOM Images in Distributed File System. **Journal of Physics: Conference Series**, 341, 2012.

STONEBRAKER, M. Inclusion of New Types in Relational Data Base Systems. In: ICDE 1986, Los Angeles, California, USA. **Proceedings of the Second International Conference on Data Engineering**. Washington, DC, USA: IEEE Computer Society, 1986. p. 262-269.

SURJANTO, B.; RITTER, N.; LOESER, H. XML content management based on object-relational database technology. In: WISE 2000, Hong Kong, China. **Proceedings of the First International Conference on Web Information Systems Engineering**. Los Alamitos, CA, USA: IEEE Computer Society Press, 2000. p. 70-79.

TENG, C. *et al.* A Medical Image Archive Solution in the Cloud. In: ICSESS, Beijing, China. **2010 IEEE International Conference on Software Engineering and Service Sciences**. Washington, DC, USA: IEEE Computer Society, 2010. p. 431-434.

VOGL, R. *et al.* The Innsbruck Advanced Image Management (AIM) Project-A Central Archive and Distribution System for Radiology and Multimedia Data. **International Congress Series**, v. 1281, p. 871-876, 2005.

WARNOCK, M. J. *et al.* Benefits of Using the DCM4CHE DICOM Archive. **Journal of Digital Imaging**, v. 20, n. 1, p. 125-129, 2007.

WATSON, R. W. High Performance Storage System Scalability: Architecture, Implementation and Experience. In: MSST '05, Monterey, CA, USA. **Proceedings of the 22nd IEEE / 13th NASA Goddard Conference on Mass Storage Systems and Technologies**, 2005. p. 145-159.

WHITE, R. V.; MILES, F. M. Principles of Fault Tolerance. In: APEC '96, San Jose, CA, USA. **Proceedings of the 11th Annual Applied Power Electronics Conference and Exposition**. Los Alamitos, CA, USA: IEEE Computer Society Press, 1996. p. 18-25.

YAKAMI, M. *et al.* Development and evaluation of a low-cost and high-capacity DICOM image data storage system for research. **Journal of Digital Imaging**, v. 24, n. 2, p. 190-195, 2011.

- YANG, C. *et al.* MIFAS: Medical Image File Accessing System in Co-allocation Data Grids. In: APSCC '08, Yilan, Taiwan. **IEEE Asia-Pacific Services Computing Conference**. Washington, DC, USA: IEEE Computer Society, 2008. p. 769-774.
- YU, D. *et al.* A XML-based Remote EMI Sharing System Conformable to DICOM. In: ITAB 2008, Shenzhen, China. **International Conference on Information Technology and Applications in Biomedicine**. Washington, DC, USA: IEEE Computer Society, 2008. p. 556-559.
- YUAN, K. *et al.* Diagnosis System of Computer-aided Brain MRI using Content-based Image Retrieval. In: ITAB 2008, Shenzhen, China. **International Conference on Information Technology and Applications in Biomedicine**. Washington, DC, USA: IEEE Computer Society, 2008. p. 152-156.
- ZHANG, Y.; FENG, D. An Active Storage System for High Performance Computing. In: AINA 2008, Okinawa, Japan. **22nd International Conference on Advanced Information Networking and Applications**. Washington, DC, USA: IEEE Computer Society, 2008. p. 644-651.
- ZHANG, Z. *et al.* Medical Data on Demand with WebMIA. **IEEE Engineering in Medicine and Biology Magazine**, v. 24, n. 3, p. 117-122, 2005.

GLOSSÁRIO DE TERMOS E EXPRESSÕES

AE (*Application Entity* – Entidade de Aplicação) – Um componente de *hardware* (como por exemplo um Tomógrafo Computadorizado, utilizado na aquisição de imagens médicas digitais) ou *software* (como por exemplo um servidor HL7 ou DICOM, executado como um processo), utilizado na construção de uma instância PACS. Entidades de aplicação são identificadas unicamente por um nome (AETITLE, de acordo com a definição do padrão DICOM), e comunicam-se entre si obedecendo aos protocolos de comunicação estabelecidos pelo mesmo padrão. A comunicação ocorre de forma ponto-a-ponto, com cada AE assumindo o papel de usuário de serviço (*Service Class User* – SCU) ou provedor de serviço (*Service Class Provider* – SCP).

Armazenamento *near-line* – Estratégia de armazenamento de dados intermediária entre *online* e *offline*. Nela, dependendo das decisões de projeto, o acesso aos dados de interesse pode ser imediato ou sofrer um atraso decorrente de recuperações feitas a partir de mídias mais lentas (como fitas e DVDs – *Digital Video Disks*).

Armazenamento *offline* – Estratégia de armazenamento de dados que demanda um intervalo de tempo maior para que o conteúdo de interesse seja disponibilizado para consulta e recuperação. Comumente, utiliza mídias mais lentas (porém com maior capacidade e menor custo) para o armazenamento massivo de dados históricos.

Armazenamento *online* – Estratégia de armazenamento de dados onde o conteúdo é pesquisável e recuperável de forma imediata. Comumente, é organizado sobre componentes de *hardware* com um maior desempenho de acesso, com o objetivo de acelerar consultas e recuperação de conteúdo.

DICOM (*Digital Imaging and Communications in Medicine* – Comunicação de Imagens Digitais em Medicina) – Um padrão para a estruturação de imagens médicas digitais, incluindo protocolos de comunicação para a transferência dessas imagens (comumente no contexto de um sistema PACS).

DICOM *Conformance Statement* (Declaração de Conformidade DICOM) – Declaração que acompanha as entidades de aplicação, explicitando quais serviços são suportados pelas mesmas e que papéis essas entidades assumem na comunicação de dados com outras entidades.

DIMSE (*DICOM Message Service Element*) – Conjunto de serviços e protocolos utilizados pelas AEs com o propósito de enviar e receber imagens médicas e informações correlatas.

HIS (*Hospital Information System* – Sistema de Informações Hospitalares) – Um sistema integrado de informações no contexto de uma instituição de saúde (comumente um hospital), englobando dados de pacientes, dados clínicos e dados financeiros (incluindo insumos).

HL7 (Health Level Seven) – Um conjunto de normas e padrões internacionais para a representação e a transferência de dados administrativos e dados clínicos entre instituições de saúde, com base no conceito de troca de mensagens.

IOD (Information Object Definition) – Um modelo de dados abstrato, orientado a objetos, usado na especificação de informações a respeito de objetos existentes no mundo real.

MWL (DICOM Modality Worklist – Lista de Tarefas) – Um serviço que permite a transmissão de dados coletados previamente à execução de exames (armazenados em sistemas HIS/RIS) para AEs em sistemas PACS. Comumente, dados de pacientes (como nome, data de nascimento, sexo) e especificações relacionadas aos exames a serem executados (como parte do corpo e protocolo de aquisição de imagens) são coletados e armazenados em sistemas hospitalares e radiológicos, sendo redigitados nos equipamentos utilizados para a aquisição de imagens nos sistemas PACS. Com MWLs, os dados já adquiridos podem ser transferidos entre os diferentes sistemas de forma a otimizar o tempo de uso dos equipamentos e reduzir erros decorrentes da redigitação de dados já conhecidos.

PACS (Picture Archiving and Communication System – Sistema de Comunicação e Arquivamento de Imagens) – Um sistema utilizado para o gerenciamento de conteúdo médico digital (imagens, vídeos, laudos em áudio ou textuais), construído de forma a prover recursos para a aquisição, o armazenamento e a comunicação de dados dentro dos limites de uma instituição de saúde ou mesmo entre diferentes instituições.

RIS (Radiology Information System – Sistema de Informações Radiológicas) – Um sistema de informações utilizado no contexto de um setor de radiologia, provendo recursos de *hardware* e *software* para o gerenciamento de informações relacionadas a pacientes, procedimentos e fluxo de trabalho envolvendo profissionais e equipamentos.

SCP (Service Class Provider) – Perfil de provedor de serviço assumido por uma AE quando em comunicação com outra AE.

SCU (Service Class User) – Perfil de usuário de serviço assumido por uma AE quando em comunicação com outra AE.

SOP (Service-Object Pair) – União entre um IOD e um DIMSE. Relaciona quais operações podem ser aplicadas e quais serviços podem ser executados sobre dados em uma comunicação estabelecida entre duas AEs.

Tag DICOM – A menor unidade de informação definida pelo padrão. Também conhecida como *elemento de dados*, é identificada unicamente por um número de *grupo* e um número de *elemento* e caracterizada por um VR e um VM.

VM (Value Multiplicity – Multiplicidade de Valor) – Indica o número de valores suportado por uma instância de *tag* DICOM. Quando a multiplicidade de valor de uma *tag* for maior do que 1 (um), o VR da *tag* estabelece um separador de valores a ser utilizado em tempo de *parsing*. Exemplos: *tag* (0010,0010) (*PatientName*) – possui

VM = 1, indicando que um paciente pode ter apenas 1 (um) nome; *tag* (0010,2110) (*Allergies*) – possui VM = 1-n, indicando de um paciente pode ter alergia a um ou mais agentes.

VR (*Value Representation – Valor de Representação*) – Também conhecido como *tipo de dado DICOM*, um VR caracteriza o conteúdo de uma *tag* no que se refere à tipologia de dados adotada e o domínio de dados válido. O padrão define 27 VRs diferentes abrangendo tipos textuais, numéricos (representados nos formatos texto e binário) e binários. Exemplo: CS (*Code String*) = *string* de caracteres de tamanho máximo 16, aceitando letras maiúsculas, dígitos de 0 a 9, espaços em branco e *underscore* (“_”).

Workstation (*Estação de Trabalho*) – Equipamento utilizado para a visualização ou manipulação de imagens médias. Dependendo da finalidade de uso, pode ser um computador pessoal (comumente utilizado na área clínica) ou um computador com periféricos específicos a determinadas funções (como por exemplo monitores radiológicos).

APÊNDICE 1 – EXPRESSÕES DE BUSCA GERADAS PELAS INTERFACES DE ACESSO ÀS BIBLIOTECAS DIGITAIS

As expressões relacionadas a seguir foram geradas automaticamente pelas ferramentas de busca das bibliotecas digitais consultadas, a partir dos termos de busca definidos na Seção 3.2.3.

IEEE Xplore

```
("Document Title":DICOM) AND "Document Title":Storage) OR
((((((((("Document Title":DICOM) OR "Document Title":Digital Imaging and
Communications in Medicine") OR "Document Title":Medical Imaging") OR
"Document Title":Medical Image") OR "Document Title":Medical Images")
OR "Document Title":Clinical Imaging") OR "Document Title":Clinical
Image") OR "Document Title":Clinical Images") AND (((("Document
Title":Storage) OR "Document Title":Archive) OR "Document Title":Archiving)
OR (((("Document Title":Database) OR "Document Title":Databases) OR
"Document Title":DBMS) OR "Document Title":RDBMS)))))) OR
((((((((("Abstract":DICOM) OR "Abstract":Digital Imaging and
Communications in Medicine") OR "Abstract":Medical Imaging") OR
"Abstract":Medical Image") OR "Abstract":Medical Images") OR
"Abstract":Clinical Imaging") OR "Abstract":Clinical Image") OR
"Abstract":Clinical Images") AND (((("Abstract":Storage) OR
"Abstract":Archive) OR "Abstract":Archiving) OR (((("Abstract":Database) OR
"Abstract":Databases) OR "Abstract":DBMS) OR "Abstract":RDBMS)))))) AND
(((("Abstract":PACS) OR "Abstract":Picture Archiving and Communication
System")))) OR (((((((("Search Index Terms":DICOM) OR "Search Index
Terms":Digital Imaging and Communications in Medicine") OR "Search
Index Terms":Medical Imaging") OR "Search Index Terms":Medical Image")
OR "Search Index Terms":Medical Images") OR "Search Index
Terms":Clinical Imaging") OR "Search Index Terms":Clinical Image") OR
"Search Index Terms":Clinical Images") AND (((("Search Index
Terms":Storage) OR "Search Index Terms":Archive) OR "Search Index
Terms":Archiving) OR (((("Search Index Terms":Database) OR "Search
Index Terms":Databases) OR "Search Index Terms":DBMS) OR "Search
Index Terms":RDBMS)))))) AND (((("Search Index Terms":PACS) OR "Search
Index Terms":Picture Archiving and Communication System"))))
```

SpringerLink

```
(DICOM AND Storage) OR ((DICOM OR "Medical Image") AND (Storage OR
Database) AND (PACS))
```

ACM Digital Library

(Title:DICOM AND Title:Storage) OR ((Title:DICOM OR Title:"Medical Imaging" OR Title:"Medical Image" OR Title:"Medical Images" OR Title:"Clinical Imaging" OR Title:"Clinical Images") AND (((Title:Storage OR Title:Archive OR Title:Archiving) OR ((Title:Database OR Title:Databases OR Title:DBMS OR Title:RDBMS)))) OR ((Abstract:DICOM OR Abstract:"Medical Imaging" OR Abstract:"Medical Image" OR Abstract:"Medical Images" OR Abstract:"Clinical Imaging" OR Abstract:"Clinical Images") AND (((Abstract:Storage OR Abstract:Archive OR Abstract:Archiving) OR ((Abstract:Database OR Abstract:Databases OR Abstract:DBMS OR Abstract:RDBMS)))) AND ((Abstract:PACS))) OR ((ReviewText:DICOM OR ReviewText:"Medical Imaging" OR ReviewText:"Medical Image" OR ReviewText:"Medical Images" OR ReviewText:"Clinical Imaging" OR ReviewText:"Clinical Images") AND (((ReviewText:Storage OR ReviewText:Archive OR ReviewText:Archiving) OR ((ReviewText:Database OR ReviewText:Databases OR ReviewText:DBMS OR ReviewText:RDBMS)))) AND ((ReviewText:PACS))) OR ((Keywords:DICOM OR Keywords:"Medical Imaging" OR Keywords:"Medical Image" OR Keywords:"Medical Images" OR Keywords:"Clinical Imaging" OR Keywords:"Clinical Images") AND (((Keywords:Storage OR Keywords:Archive OR Keywords:Archiving) OR ((Keywords:Database OR Keywords:Databases OR Keywords:DBMS OR Keywords:RDBMS)))) AND ((Keywords:PACS)))

ScienceDirect

TITLE(DICOM AND Storage) OR TITLE-ABSTR-KEY((DICOM OR "Digital Imaging and Communications in Medicine" OR "Medical Imaging" OR "Medical Image" OR "Medical Images" OR "Clinical Imaging" OR "Clinical Image" OR "Clinical Images") AND ((Storage OR Archive OR Archiving) OR (Database OR Databases OR DBMS OR RDBMS)) AND (PACS OR "Picture Archiving and Communication System"))

Wiley

(DICOM AND Storage) in Publication Titles OR (DICOM AND Storage) in Article Titles OR ((DICOM OR "Digital Imaging and Communications in Medicine" OR "Medical Imaging" OR "Medical Image" OR "Medical Images" OR "Clinical Imaging" OR "Clinical Image" OR "Clinical Images") AND ((Storage OR Archive OR Archiving) OR (Database OR Databases OR DBMS OR RDBMS))) in Publication Titles OR ((DICOM OR "Digital Imaging and Communications in Medicine" OR "Medical Imaging" OR "Medical

Image" OR "Medical Images" OR "Clinical Imaging" OR "Clinical Image" OR "Clinical Images") AND ((Storage OR Archive OR Archiving) OR (Database OR Databases OR DBMS OR RDBMS)) AND (PACS OR "Picture Archiving and Communication System")) in Article Titles OR ((DICOM OR "Digital Imaging and Communications in Medicine" OR "Medical Imaging" OR "Medical Image" OR "Medical Images" OR "Clinical Imaging" OR "Clinical Image" OR "Clinical Images") AND ((Storage OR Archive OR Archiving) OR (Database OR Databases OR DBMS OR RDBMS)) AND (PACS OR "Picture Archiving and Communication System")) in Abstract OR ((DICOM OR "Digital Imaging and Communications in Medicine" OR "Medical Imaging" OR "Medical Image" OR "Medical Images" OR "Clinical Imaging" OR "Clinical Image" OR "Clinical Images") AND ((Storage OR Archive OR Archiving) OR (Database OR Databases OR DBMS OR RDBMS)) AND (PACS OR "Picture Archiving and Communication System")) in Keywords

Web of Knowledge

Topic=((DICOM AND Storage)) OR Title=((DICOM AND Storage)) OR Topic=(((DICOM OR "Digital Imaging and Communications in Medicine" OR "Medical Imaging" OR "Medical Image" OR "Medical Images" OR "Clinical Imaging" OR "Clinical Image" OR "Clinical Images") AND ((Storage OR Archive OR Archiving) OR (Database OR Databases OR DBMS OR RDBMS)) AND (PACS OR "Picture Archiving and Communication System")) OR Title=(((DICOM OR "Digital Imaging and Communications in Medicine" OR "Medical Imaging" OR "Medical Image" OR "Medical Images" OR "Clinical Imaging" OR "Clinical Image" OR "Clinical Images") AND ((Storage OR Archive OR Archiving) OR (Database OR Databases OR DBMS OR RDBMS)) AND (PACS OR "Picture Archiving and Communication System"))

APÊNDICE 2 – DICIONÁRIO DE DADOS DO ESQUEMA DE BD PARA ARMAZENAMENTO DICOM

Entidade: chave_hierarquica			
Descrição: Relação entre os quatro atributos (<i>tags</i>) identificadores dos níveis hierárquicos do padrão DICOM.			
Atributo	Domínio	Tamanho	Descrição
id	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária substituta).
patientid	Texto	64	Identificador de um paciente – corresponde à <i>tag</i> (0010,0020) do dicionário de dados DICOM.
studyinstanceuid	Texto	64	Identificador de um estudo – corresponde à <i>tag</i> (0020,000D) do dicionário de dados DICOM.
seriesinstanceuid	Texto	64	Identificador de uma série – corresponde à <i>tag</i> (0020,000E) do dicionário de dados DICOM.
sopinstanceuid	Texto	64	Identificador de uma imagem – corresponde à <i>tag</i> (0008,0018) do dicionário de dados DICOM.

Entidade: conteudo_original			
Descrição: Imagens DICOM armazenadas na íntegra.			
Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária; chave estrangeira para a entidade 'chave_hierarquica').
conteudo	Binário		Conteúdo original da imagem DICOM.

Entidade: tag_ae			
Descrição: <i>Tags</i> DICOM com VR AE (<i>Application Entity</i>).			
Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da <i>tag</i> na imagem DICOM original (chave primária

			parcial).
tipo	Texto	1	Identifica se a <i>tag</i> é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a <i>tag</i> pertence.
elemento	Numérico		Número de elemento para a <i>tag</i> dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da <i>tag</i> (para os casos em que VM > 1) (chave primária parcial).
conteudo	Texto	16	Conteúdo da <i>tag</i> .
tamanho	Numérico		Tamanho (em <i>bytes</i>) do conteúdo da <i>tag</i> .

Entidade: tag_as			
Descrição: Tags DICOM com VR AS (<i>Age String</i>).			
Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da <i>tag</i> na imagem DICOM original (chave primária parcial).
tipo	Texto	1	Identifica se a <i>tag</i> é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a <i>tag</i> pertence.
elemento	Numérico		Número de elemento para a <i>tag</i> dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da <i>tag</i> (para os casos em que VM > 1) (chave primária parcial).
conteudo	Texto	4	Conteúdo da <i>tag</i> .

Entidade: tag_at			
Descrição: Tags DICOM com VR AT (<i>Attribute Tag</i>).			
Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da <i>tag</i> na imagem DICOM original (chave primária

			parcial).
tipo	Texto	1	Identifica se a <i>tag</i> é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a <i>tag</i> pertence.
elemento	Numérico		Número de elemento para a <i>tag</i> dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da <i>tag</i> (para os casos em que VM > 1) (chave primária parcial).
conteudo	Binário	4	Conteúdo da <i>tag</i> .

Entidade: tag_cs**Descrição:** Tags DICOM com VR CS (Code String).

Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da <i>tag</i> na imagem DICOM original (chave primária parcial).
tipo	Texto	1	Identifica se a <i>tag</i> é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a <i>tag</i> pertence.
elemento	Numérico		Número de elemento para a <i>tag</i> dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da <i>tag</i> (para os casos em que VM > 1) (chave primária parcial).
conteudo	Texto	16	Conteúdo da <i>tag</i> .
tamanho	Numérico		Tamanho (em <i>bytes</i>) do conteúdo da <i>tag</i> .

Entidade: tag_da**Descrição:** Tags DICOM com VR DA (Date).

Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da <i>tag</i> na imagem DICOM original (chave primária

			parcial).
tipo	Texto	1	Identifica se a <i>tag</i> é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a <i>tag</i> pertence.
elemento	Numérico		Número de elemento para a <i>tag</i> dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da <i>tag</i> (para os casos em que VM > 1) (chave primária parcial).
conteudo	Texto	8	Conteúdo da <i>tag</i> .

Entidade: tag_ds**Descrição:** Tags DICOM com VR DS (*Decimal String*).

Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da <i>tag</i> na imagem DICOM original (chave primária parcial).
tipo	Texto	1	Identifica se a <i>tag</i> é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a <i>tag</i> pertence.
elemento	Numérico		Número de elemento para a <i>tag</i> dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da <i>tag</i> (para os casos em que VM > 1) (chave primária parcial).
conteudo	Texto	16	Conteúdo da <i>tag</i> .
tamanho	Numérico		Tamanho (em <i>bytes</i>) do conteúdo da <i>tag</i> .

Entidade: tag_dt**Descrição:** Tags DICOM com VR DT (*Date Time*).

Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da <i>tag</i> na imagem DICOM original (chave primária

			parcial).
tipo	Texto	1	Identifica se a <i>tag</i> é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a <i>tag</i> pertence.
elemento	Numérico		Número de elemento para a <i>tag</i> dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da <i>tag</i> (para os casos em que VM > 1) (chave primária parcial).
conteudo	Texto	26	Conteúdo da <i>tag</i> .
tamanho	Numérico		Tamanho (em <i>bytes</i>) do conteúdo da <i>tag</i> .

Entidade: tag_fd			
Descrição: Tags DICOM com VR FD (<i>Floating Point Double</i>).			
Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da <i>tag</i> na imagem DICOM original (chave primária parcial).
tipo	Texto	1	Identifica se a <i>tag</i> é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a <i>tag</i> pertence.
elemento	Numérico		Número de elemento para a <i>tag</i> dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da <i>tag</i> (para os casos em que VM > 1) (chave primária parcial).
conteudo	Numérico		Conteúdo da <i>tag</i> .

Entidade: tag_fl			
Descrição: Tags DICOM com VR FL (<i>Floating Point Single</i>).			
Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da <i>tag</i> na imagem DICOM original (chave primária

			parcial).
tipo	Texto	1	Identifica se a <i>tag</i> é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a <i>tag</i> pertence.
elemento	Numérico		Número de elemento para a <i>tag</i> dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da <i>tag</i> (para os casos em que VM > 1) (chave primária parcial).
conteudo	Numérico		Conteúdo da <i>tag</i> .

Entidade: tag_is**Descrição:** Tags DICOM com VR IS (*Integer String*).

Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da <i>tag</i> na imagem DICOM original (chave primária parcial).
tipo	Texto	1	Identifica se a <i>tag</i> é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a <i>tag</i> pertence.
elemento	Numérico		Número de elemento para a <i>tag</i> dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da <i>tag</i> (para os casos em que VM > 1) (chave primária parcial).
conteudo	Texto	12	Conteúdo da <i>tag</i> .
tamanho	Numérico		Tamanho (em <i>bytes</i>) do conteúdo da <i>tag</i> .

Entidade: tag_lo**Descrição:** Tags DICOM com VR LO (*Long String*).

Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da <i>tag</i> na imagem DICOM original (chave primária

			parcial).
tipo	Texto	1	Identifica se a <i>tag</i> é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a <i>tag</i> pertence.
elemento	Numérico		Número de elemento para a <i>tag</i> dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da <i>tag</i> (para os casos em que VM > 1) (chave primária parcial).
conteudo	Texto	64	Conteúdo da <i>tag</i> .
tamanho	Numérico		Tamanho (em <i>bytes</i>) do conteúdo da <i>tag</i> .

Entidade: tag_lt			
Descrição: Tags DICOM com VR LT (<i>Long Text</i>).			
Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da <i>tag</i> na imagem DICOM original (chave primária parcial).
tipo	Texto	1	Identifica se a <i>tag</i> é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a <i>tag</i> pertence.
elemento	Numérico		Número de elemento para a <i>tag</i> dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da <i>tag</i> (para os casos em que VM > 1) (chave primária parcial).
conteudo	Texto	10240	Conteúdo da <i>tag</i> .
tamanho	Numérico		Tamanho (em <i>bytes</i>) do conteúdo da <i>tag</i> .

Entidade: tag_ob			
Descrição: Tags DICOM com VR OB (<i>Other Byte String</i>).			
Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').

ordem_tag	Numérico		Ordem (posição) da <i>tag</i> na imagem DICOM original (chave primária parcial).
tipo	Texto	1	Identifica se a <i>tag</i> é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a <i>tag</i> pertence.
elemento	Numérico		Número de elemento para a <i>tag</i> dentro do grupo.
sequencial_1	Numérico		Primeiro sequencial (posição) do conteúdo dentro da <i>tag</i> (para os casos em que VM > 1) (chave primária parcial).
sequencial_2	Numérico		Segundo sequencial (posição) do conteúdo dentro da <i>tag</i> (para os casos em que VM > 1) (chave primária parcial).
conteudo	Binário		Conteúdo da <i>tag</i> .
tamanho	Numérico		Tamanho (em <i>bytes</i>) do conteúdo da <i>tag</i> .

Entidade: tag_of			
Descrição: Tags DICOM com VR OF (<i>Other Float String</i>).			
Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da <i>tag</i> na imagem DICOM original (chave primária parcial).
tipo	Texto	1	Identifica se a <i>tag</i> é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a <i>tag</i> pertence.
elemento	Numérico		Número de elemento para a <i>tag</i> dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da <i>tag</i> (para os casos em que VM > 1) (chave primária parcial).
conteudo	Binário		Conteúdo da <i>tag</i> .
tamanho	Numérico		Tamanho (em <i>bytes</i>) do conteúdo da <i>tag</i> .

Entidade: tag_ow			
Descrição: Tags DICOM com VR OW (<i>Other Word String</i>).			
Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da <i>tag</i> na imagem DICOM original (chave primária parcial).
tipo	Texto	1	Identifica se a <i>tag</i> é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a <i>tag</i> pertence.
elemento	Numérico		Número de elemento para a <i>tag</i> dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da <i>tag</i> (para os casos em que VM > 1) (chave primária parcial).
conteudo	Binário		Conteúdo da <i>tag</i> .
tamanho	Numérico		Tamanho (em <i>bytes</i>) do conteúdo da <i>tag</i> .

Entidade: tag_pn			
Descrição: Tags DICOM com VR PN (<i>Person Name</i>).			
Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da <i>tag</i> na imagem DICOM original (chave primária parcial).
tipo	Texto	1	Identifica se a <i>tag</i> é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a <i>tag</i> pertence.
elemento	Numérico		Número de elemento para a <i>tag</i> dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da <i>tag</i> (para os casos em que VM > 1) (chave primária parcial).
conteudo	Texto	64	Conteúdo da <i>tag</i> .
tamanho	Numérico		Tamanho (em <i>bytes</i>) do conteúdo da <i>tag</i> .

Entidade: tag_sh			
Descrição: Tags DICOM com VR SH (<i>Short String</i>).			
Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da <i>tag</i> na imagem DICOM original (chave primária parcial).
tipo	Texto	1	Identifica se a <i>tag</i> é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a <i>tag</i> pertence.
elemento	Numérico		Número de elemento para a <i>tag</i> dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da <i>tag</i> (para os casos em que VM > 1) (chave primária parcial).
conteudo	Texto	16	Conteúdo da <i>tag</i> .
tamanho	Numérico		Tamanho (em <i>bytes</i>) do conteúdo da <i>tag</i> .

Entidade: tag_sl			
Descrição: Tags DICOM com VR SL (<i>Signed Long</i>).			
Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da <i>tag</i> na imagem DICOM original (chave primária parcial).
tipo	Texto	1	Identifica se a <i>tag</i> é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a <i>tag</i> pertence.
elemento	Numérico		Número de elemento para a <i>tag</i> dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da <i>tag</i> (para os casos em que VM > 1) (chave primária parcial).
conteudo	Numérico		Conteúdo da <i>tag</i> .

Entidade: tag_sq			
Descrição: Tags DICOM com VR SQ (Sequence of Items).			
Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da <i>tag</i> na imagem DICOM original (chave primária parcial).
tipo	Texto	1	Identifica se a <i>tag</i> é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a <i>tag</i> pertence.
elemento	Numérico		Número de elemento para a <i>tag</i> dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da <i>tag</i> (para os casos em que VM > 1) (chave primária parcial).
cardinalidade	Numérico		Cardinalidade (número) de itens aninhados à <i>tag</i> .

Entidade: tag_ss			
Descrição: Tags DICOM com VR SS (Signed Short).			
Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da <i>tag</i> na imagem DICOM original (chave primária parcial).
tipo	Texto	1	Identifica se a <i>tag</i> é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a <i>tag</i> pertence.
elemento	Numérico		Número de elemento para a <i>tag</i> dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da <i>tag</i> (para os casos em que VM > 1) (chave primária parcial).
conteudo	Numérico		Conteúdo da <i>tag</i> .

Entidade: tag_st			
Descrição: Tags DICOM com VR ST (Short Text).			
Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da <i>tag</i> na imagem DICOM original (chave primária parcial).
tipo	Texto	1	Identifica se a <i>tag</i> é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a <i>tag</i> pertence.
elemento	Numérico		Número de elemento para a <i>tag</i> dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da <i>tag</i> (para os casos em que VM > 1) (chave primária parcial).
conteudo	Texto	1024	Conteúdo da <i>tag</i> .
tamanho	Numérico		Tamanho (em <i>bytes</i>) do conteúdo da <i>tag</i> .

Entidade: tag_tm			
Descrição: Tags DICOM com VR TM (Time).			
Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da <i>tag</i> na imagem DICOM original (chave primária parcial).
tipo	Texto	1	Identifica se a <i>tag</i> é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a <i>tag</i> pertence.
elemento	Numérico		Número de elemento para a <i>tag</i> dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da <i>tag</i> (para os casos em que VM > 1) (chave primária parcial).
conteudo	Texto	16	Conteúdo da <i>tag</i> .
tamanho	Numérico		Tamanho (em <i>bytes</i>) do conteúdo da <i>tag</i> .

Entidade: tag_ui			
Descrição: Tags DICOM com VR UI (<i>Unique Identifier (UID)</i>).			
Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da tag na imagem DICOM original (chave primária parcial).
tipo	Texto	1	Identifica se a tag é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a tag pertence.
elemento	Numérico		Número de elemento para a tag dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da tag (para os casos em que VM > 1) (chave primária parcial).
conteudo	Texto	64	Conteúdo da tag.
tamanho	Numérico		Tamanho (em bytes) do conteúdo da tag.

Entidade: tag_ul			
Descrição: Tags DICOM com VR UL (<i>Unsigned Long</i>).			
Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da tag na imagem DICOM original (chave primária parcial).
tipo	Texto	1	Identifica se a tag é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a tag pertence.
elemento	Numérico		Número de elemento para a tag dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da tag (para os casos em que VM > 1) (chave primária parcial).
conteudo	Numérico		Conteúdo da tag.

Entidade: tag_un			
Descrição: Tags DICOM com VR UN (<i>Unknown</i>).			
Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da tag na imagem DICOM original (chave primária parcial).
tipo	Texto	1	Identifica se a tag é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a tag pertence.
elemento	Numérico		Número de elemento para a tag dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da tag (para os casos em que VM > 1) (chave primária parcial).
cardinalidade	Numérico		Cardinalidade (número) de itens aninhados à tag.
conteudo	Binário		Conteúdo da tag.
tamanho	Numérico		Tamanho (em bytes) do conteúdo da tag.

Entidade: tag_us			
Descrição: Tags DICOM com VR US (<i>Unsigned Short</i>).			
Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da tag na imagem DICOM original (chave primária parcial).
tipo	Texto	1	Identifica se a tag é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a tag pertence.
elemento	Numérico		Número de elemento para a tag dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da tag (para os casos em que VM > 1) (chave

			primária parcial).
conteudo	Numérico		Conteúdo da <i>tag</i> .

Entidade: tag_ut			
Descrição: Tags DICOM com VR UT (<i>Unlimited Text</i>).			
Atributo	Domínio	Tamanho	Descrição
id_chave_hierarquica	Numérico		Identificador único de uma relação entre os quatro níveis hierárquicos do padrão DICOM (chave primária parcial; chave estrangeira para a entidade 'chave_hierarquica').
ordem_tag	Numérico		Ordem (posição) da <i>tag</i> na imagem DICOM original (chave primária parcial).
tipo	Texto	1	Identifica se a <i>tag</i> é de [C]abeçalho ou de [D]ados.
grupo	Numérico		Número do grupo ao qual a <i>tag</i> pertence.
elemento	Numérico		Número de elemento para a <i>tag</i> dentro do grupo.
sequencial	Numérico		Número sequencial (posição) do conteúdo dentro da <i>tag</i> (para os casos em que VM > 1) (chave primária parcial).
conteudo	Texto		Conteúdo da <i>tag</i> .
tamanho	Numérico		Tamanho (em <i>bytes</i>) do conteúdo da <i>tag</i> .

APÊNDICE 3 – EXEMPLOS DE INSTRUÇÕES SQL USADAS NOS EXPERIMENTOS DE PESQUISA DE METADADOS

As instruções em SQL relacionadas são construídas de acordo com o esquema de BD baseado no modelo de armazenamento decomposto, e permitem recuperar metadados com base em predicados construídos a partir de testes de igualdade com os identificadores únicos dos quatro níveis hierárquicos do padrão DICOM, testes de igualdade com atributos comuns, testes envolvendo intervalos de valores com atributos comuns e testes envolvendo correspondências de padrão com atributos comuns.

Alternativa	Instrução SQL
Identificador único	<p>Recuperar conteúdo da <i>tag</i> (0010,0010) (nome do paciente) para um determinado valor do identificador único <i>patientid</i></p> <pre>SELECT ch.patientid, pn.conteudo FROM tag_pn pn INNER JOIN chave_hierarquica ch ON pn.id_chave_hierarquica = ch.id WHERE ch.patientid = '1.3.6.1.4.1.9328.50.14.0012' AND pn.grupo = 16 AND pn.elemento = 16 GROUP BY ch.patientid, pn.conteudo;</pre>
Igualdade de valor	<p>Recuperar conteúdo da <i>tag</i> (0010,0010) (nome do paciente) para um determinado valor da <i>tag</i> (0008,103E) (descrição da série do exame)</p> <pre>SELECT ch.patientid, pn.conteudo FROM tag_pn pn INNER JOIN chave_hierarquica ch ON pn.id_chave_hierarquica = ch.id INNER JOIN tag_lo lo ON ch.id = lo.id_chave_hierarquica WHERE pn.grupo = 16 AND pn.elemento = 16 AND lo.grupo = 8 AND lo.elemento = 4158 AND lo.conteudo = 'CT CHEST 5 X 5' GROUP BY ch.patientid, pn.conteudo;</pre>
Intervalo de valores	<p>Recuperar conteúdo da <i>tag</i> (0010,0010) (nome do paciente) para um determinado intervalo de valores da <i>tag</i> (0008,0022) (data de aquisição das imagens)</p> <pre>SELECT ch.patientid, pn.conteudo FROM tag_pn pn INNER JOIN chave_hierarquica ch</pre>

	<pre> ON pn.id_chave_hierarquica = ch.id INNER JOIN tag_da da ON ch.id = da.id_chave_hierarquica WHERE pn.grupo = 16 AND pn.elemento = 16 AND da.grupo = 8 AND da.elemento = 34 AND da.conteudo BETWEEN '19960308' AND '19960815' GROUP BY ch.patientid, pn.conteudo; </pre>
Correspondência de padrão	<p>Recuperar conteúdo da <i>tag</i> (0010,0010) (nome do paciente) dos registros que possuam relação com a <i>tag</i> (0008,1030) (descrição do estudo DICOM) que armazene a <i>string</i> 'ABDO'</p> <pre> SELECT ch.patientid, pn.conteudo FROM tag_pn pn INNER JOIN chave_hierarquica ch ON pn.id_chave_hierarquica = ch.id INNER JOIN tag_lo lo ON ch.id = lo.id_chave_hierarquica WHERE pn.grupo = 16 AND pn.elemento = 16 AND lo.grupo = 8 AND lo.elemento = 4144 AND lo.conteudo LIKE '%ABDO%' GROUP BY ch.patientid, pn.conteudo; </pre>

APÊNDICE 4 – EXEMPLOS DE INSTRUÇÕES SQL USADAS NOS EXPERIMENTOS DE RECUPERAÇÃO DE CONTEÚDO

As instruções em SQL relacionadas são construídas de acordo com o esquema de BD baseado no modelo de armazenamento decomposto, e permitem recuperar imagens completas para os níveis estudo, série e imagem do padrão DICOM.

Nível hierárquico	Instrução SQL
Estudo	<pre>SELECT ch.patientid, ch.studyinstanceuid, co.conteudo FROM conteudo_original co INNER JOIN chave_hierarquica ch ON co.id_chave_hierarquica = ch.id WHERE ch.patientid = '1.3.6.1.4.1.9328.50.14.0012' AND ch.studyinstanceuid = '1.3.6.1.4.1.9328.50.14.4966';</pre>
Série	<pre>SELECT ch.patientid, ch.studyinstanceuid, ch.seriesinstanceuid, co.conteudo FROM conteudo_original co INNER JOIN chave_hierarquica ch ON co.id_chave_hierarquica = ch.id WHERE ch.patientid = '1.3.6.1.4.1.9328.50.14.0012' AND ch.studyinstanceuid = '1.3.6.1.4.1.9328.50.14.4966' AND ch.seriesinstanceuid = '1.3.6.1.4.1.9328.50.14.4974';</pre>
Imagem	<pre>SELECT ch.patientid, ch.studyinstanceuid, ch.seriesinstanceuid, ch.sopinstanceuid, co.conteudo FROM conteudo_original co INNER JOIN chave_hierarquica ch ON co.id_chave_hierarquica = ch.id WHERE ch.patientid = '1.3.6.1.4.1.9328.50.14.0012' AND ch.studyinstanceuid = '1.3.6.1.4.1.9328.50.14.4966' AND ch.seriesinstanceuid = '1.3.6.1.4.1.9328.50.14.4974' AND ch.sopinstanceuid = '1.3.6.1.4.1.9328.50.14.4989';</pre>

ANEXO 1 – TAGS USADAS NA EXECUÇÃO DE EXPERIMENTOS DE PESQUISA

As tags relacionadas podem ser utilizadas como atributos únicos, requeridos ou opcionais durante a execução de pesquisas por metadados.

Nível hierárquico	Descrição	Tag	[Ú]nica [R]equerida [O]pcional	VR	VM
Paciente	Patient's Name	(0010,0010)	R	PN	1
	Patient ID	(0010,0020)	U	LO	1
	Issuer of Patient ID	(0010,0021)	O	LO	1
	Referenced Patient Sequence	(0008,1120)	O	SQ	1
	>Referenced SOP Class UID	(0008,1150)	O	UI	1
	>Referenced SOP Instance UID	(0008,1155)	O	UI	1
	Patient's Birth Date	(0010,0030)	O	DA	1
	Patient's Birth Time	(0010,0032)	O	TM	1
	Patient's Sex	(0010,0040)	O	CS	1
	Other Patient Ids	(0010,1000)	O	LO	1-n
	Other Patient Names	(0010,1001)	O	PN	1-n
	Ethnic Group	(0010,2160)	O	SH	1
	Patient Comments	(0010,4000)	O	LT	1
	Number of Patient Related Studies	(0020,1200)	O	IS	1
	Number of Patient Related Series	(0020,1202)	O	IS	1
	Number of Patient Related Instances	(0020,1204)	O	IS	1
Estudo	Study Date	(0008,0020)	R	DA	1
	Study Time	(0008,0030)	R	TM	1
	Accession Number	(0008,0050)	R	SH	1
	Study ID	(0020,0010)	R	SH	1
	Study Instance UID	(0020,000D)	U	UI	1
	Modalities in Study	(0008,0061)	O	CS	1-n
	SOP Classes in Study	(0008,0062)	O	UI	1-n
	Referring Physician's Name	(0008,0090)	O	PN	1
	Study Description	(0008,1030)	O	LO	1
	Procedure Code Sequence	(0008,1032)	O	SQ	1
	>Code Value	(0008,0100)	O	SH	1
	>Coding Scheme Designator	(0008,0102)	O	SH	1
	>Coding Scheme Version	(0008,0103)	O	SH	1
	>Code Meaning	(0008,0104)	O	LO	1
	Name of Physician(s) Reading Study	(0008,1060)	O	PN	1-n
	Admitting Diagnoses Description	(0008,1080)	O	LO	1-n
	Referenced Study Sequence	(0008,1110)	O	SQ	1
	>Referenced SOP Class UID	(0008,1150)	O	UI	1
	>Referenced SOP Instance UID	(0008,1155)	O	UI	1
	Patient's Age	(0010,1010)	O	AS	1
	Patient's Size	(0010,1020)	O	DS	1
	Patient's Weight	(0010,1030)	O	DS	1
	Occupation	(0010,2180)	O	SH	1
	Additional Patient History	(0010,21B0)	O	LT	1
	Other Study Numbers	(0020,1070)	O	IS	1-n
	Number of Study Related Series	(0020,1206)	O	IS	1
	Number of Study Related Instances	(0020,1208)	O	IS	1
Série	Modality	(0008,0060)	R	CS	1
	Series Number	(0020,0011)	R	IS	1
	Series Instance UID	(0020,000E)	U	UI	1
	Number of Series Related Instances	(0020,1209)	O	IS	1

Imagem	Instance Number	(0020,0013)	R	IS	1
	SOP Instance UID	(0008,0018)	U	UI	1
	SOP Class UID	(0008,0016)	O	UI	1
	Alternate Representation Sequence	(0008,3001)	O	SQ	1
	>Series Instance UID	(0020,000E)	O	UI	1
	>SOP Class UID	(0008,1150)	O	UI	1
	>SOP Instance UID	(0008,1155)	O	UI	1
	>Purpose of Reference Code Sequence	(0040,A170)	O	SQ	1
	>>Code Value	(0008,0100)	O	SH	1
	>>Coding Scheme Designator	(0008,0102)	O	SH	1
	>>Coding Scheme Version	(0008,0103)	O	SH	1
	>>Code Meaning	(0008,0104)	O	LO	1
	Related General SOP Class UID	(0008,001A)	O	UI	1-n
	Concept Name Code Sequence	(0040,A043)	O	SQ	1
	>Code Value	(0008,0100)	O	SH	1
	>Coding Scheme Designator	(0008,0102)	O	SH	1
	>Coding Scheme Version	(0008,0103)	O	SH	1
	>Code Meaning	(0008,0104)	O	LO	1
	Content Template Sequence	(0040,A504)	O	SQ	1
	>Template Identifier	(0040,DB00)	O	CS	1
	>Mapping Resource	(0008,0105)	O	CS	1
	Container Identifier	(0040,0512)	O	LO	1
	Specimen Description Sequence	(0040,0560)	O	SQ	1
	>Specimen Identifier	(0040,0551)	O	LO	1
	>Specimen UID	(0040,0554)	O	UI	1