

Fernando Antônio de Araújo Chacon de Albuquerque

**Arcabouço de arquitetura da informação para
ciclo de vida de projeto de vocabulário
controlado: uma aplicação em Engenharia de
Software**

Brasília

2017

Fernando Antônio de Araújo Chacon de Albuquerque

Arcabouço de arquitetura da informação para ciclo de vida de projeto de vocabulário controlado: uma aplicação em Engenharia de Software

Tese apresentada à Faculdade de Ciência da Informação da Universidade de Brasília como requisito parcial para a obtenção do título de doutor em Ciência da Informação.

Universidade de Brasília – UnB
Faculdade de Ciência da Informação
Programa de Pós-Graduação

Orientador: Prof. Dr. Mamede Lima-Marques

Brasília
2017

AAL345a Albuquerque, Fernando Antônio de Araújo Chacon de
Arcabouço de arquitetura da informação para ciclo de vida
de projeto de vocabulário controlado: uma aplicação em
Engenharia de Software / Fernando Antônio de Araújo Chacon
de Albuquerque; orientador Mamede Lima-Marques. -- Brasília,
2017.
280 p.

Tese (Doutorado - Doutorado em Ciência da Informação) --
Universidade de Brasília, 2017.

1. Arquitetura da informação. 2. Recuperação da informação.
3. Vocabulário controlado. 4. Engenharia de software. I.
Lima-Marques, Mamede, orient. II. Título.



FOLHA DE APROVAÇÃO

Título: “Arcabouço de arquitetura da informação para ciclo de vida de projeto de vocabulário controlado: uma aplicação em Engenharia de Software”

Autor (a): Fernando Antônio de Araújo Chacon de Albuquerque
Área de concentração: Gestão da Informação
Linha de pesquisa: Organização da Informação

Tese submetida à Comissão Examinadora designada pelo Colegiado do Programa de Pós-graduação em Ciência da Informação da Faculdade de Ciência da Informação da Universidade de Brasília como requisito parcial para obtenção do título de **Doutor** em Ciência da Informação.

Tese aprovada em: 24 de outubro de 2017.



Profº Drº Mamedé Lima Marques
Presidente (UnB/PPGCINF)



Profº Drº André Henrique Siqueira
Membro Externo (BACEN)



Profº Drº Edison Fereda
Membro Externo (UCB)



Profº Drº André Porto Ancona Lopez
Membro Interno (UnB/PPGCINF)

Profª Drª Ivette Kafure Munoz
Suplente (UnB/PPGCINF)

Aos meus familiares e aos leitores desta tese.

Agradecimentos

Agradeço ao Prof. Dr. Mamede Lima-Marques, principalmente, pela orientação, ensino e incentivo. Por vários motivos, agradeço aos professores Dr. André Porto Ancona Lopez, Dr. André Henrique de Siqueira e Dr. Edilson Ferneda, à Dra. Cláudia Augusto Dias, aos professores e funcionários da Faculdade de Ciência da Informação (FCI) da Universidade de Brasília (UnB), aos professores e funcionários do Departamento de Ciência da Computação (CIC) da Universidade de Brasília (UnB), aos meus familiares e aos leitores desta tese.

Resumo

A pesquisa que resultou nesta tese investigou processos de desenvolvimento e de avaliação de vocabulários controlados. Esta tese inclui os seguintes elementos: resultado de pesquisa bibliográfica sobre arquitetura da informação, recuperação da informação, organização da informação e representação da informação; proposta de arcabouço para ciclo de vida de projeto de vocabulário controlado; e exemplo de uso de elementos desse arcabouço na construção de um protótipo de vocabulário controlado no domínio da Engenharia de Software. O arcabouço proposto é composto por arquitetura de referência, modelo de domínio, modelo de qualidade e lista de atividades. Entre os elementos do modelo de qualidade proposto, existe uma lista de características de qualidade de vocabulários controlados. Os modelos propostos estão parcialmente alinhados a ferramentas semânticas existentes.

Palavras-chaves: Arquitetura da informação. Recuperação da informação. Vocabulário controlado. Engenharia de software.

Abstract

The research that resulted in this thesis has investigated development and evaluation processes of controlled vocabularies. This thesis includes the following elements: results of a bibliographic research on information architecture, information recovery, information organization and information representation; proposal of a framework for controlled vocabulary project life cycle; and example of use of this framework during the construction of a prototype of a controlled vocabulary on the Software Engineering domain. The proposed framework is composed of reference architecture, domain model, quality model and list of activities. Among the elements of the proposed quality model, there is a list of controlled vocabulary quality characteristics. The proposed models are partially aligned to existing semantic tools.

Key-words: Information architecture. Information retrieval. Controlled vocabulary. Software engineering.

Lista de ilustrações

Figura 1 – Elementos da linguagem de modelagem UML	35
Figura 2 – Elementos em processo de alinhamento de conteúdos	41
Figura 3 – Classes de recursos de informação	47
Figura 4 – Coleção de recursos de informação	48
Figura 5 – Processo de recuperação da informação	51
Figura 6 – Medidas de avaliação de sistema de recuperação da informação	56
Figura 7 – Elementos no cálculo de precisão e de revocação	57
Figura 8 – Triângulo do conceito	61
Figura 9 – Processo de construção de conceito	61
Figura 10 – Triângulo de Ogden e Richards	64
Figura 11 – Exemplo de nuvem de etiquetas	73
Figura 12 – Modelo de descrição de arquitetura	77
Figura 13 – Elementos de arcabouço teórico	80
Figura 14 – Estruturas hierárquicas	86
Figura 15 – Hierarquia de classes de relações	87
Figura 16 – Hierarquia de classes de vocabulários controlados	89
Figura 17 – Estrutura de anel de sinônimos	91
Figura 18 – Exemplo de estrutura de taxonomia	92
Figura 19 – Resposta de um tesauro a uma consulta	94
Figura 20 – Exemplo de estrutura de tesauro	95
Figura 21 – Trecho da <i>Basic Formal Ontology</i>	98
Figura 22 – Trecho de consulta à ontologia PRO	99
Figura 23 – Elementos em tripla RDF	113
Figura 24 – Exemplo de grafo RDF	114
Figura 25 – Exemplo de código RDF com termos do FOAF	115
Figura 26 – Metadado e conceitos relacionados	119
Figura 27 – Exemplo de definição de propriedade	131
Figura 28 – Fases no ciclo <i>Plan-Do-Check-Act</i>	138
Figura 29 – Elementos do GQM	143
Figura 30 – Mapeamento entre componente e impacto	148
Figura 31 – Hierarquia de características de qualidade	148
Figura 32 – Modelo de medidas (métricas) de qualidade	149
Figura 33 – Modelo de ciclo de vida no Processo Unificado	160
Figura 34 – Elementos de arcabouço de arquitetura	176
Figura 35 – Elementos de arcabouço de arquitetura da informação	177
Figura 36 – Definição de processo	179

Figura 37 – Modelo de domínio	180
Figura 38 – Modelo de qualidade – Parte 1	181
Figura 39 – Modelo de qualidade – Parte 2	182
Figura 40 – Modelo de qualidade – Parte 3	184
Figura 41 – Elementos no exemplo de uso do arcabouço	194
Figura 42 – Tela da ferramenta <i>VocBench</i>	195
Figura 43 – Sequência de uso de ferramentas	196
Figura 44 – Trecho de uma lista de verificação	205
Figura 45 – Trechos de uma das páginas do protótipo de vocabulário controlado . .	206

Lista de tabelas

Tabela 1 – Percentual de taxonomias por área	201
Tabela 2 – Percentual de ontologias por área	201

Lista de quadros

Quadro 1 – Princípios para análise de facetas	70
Quadro 2 – Categorias aplicáveis a diversos domínios	71
Quadro 3 – Relações entre categorias	72
Quadro 4 – Categorias aplicáveis a domínios específicos	72
Quadro 5 – Relação de tesouros em domínios	93
Quadro 6 – Nomes de ontologias de topo	98
Quadro 7 – Elementos do vocabulário SKOS	116
Quadro 8 – Exemplo de formulário	144
Quadro 9 – Características de qualidade da informação	144
Quadro 10 – Características de qualidade de vocabulários controlados	152
Quadro 11 – Traduções de termos	153
Quadro 12 – Exemplos de classes de recursos de informação	163
Quadro 13 – Relações entre termos no modelo de domínio e na <i>WordNet</i>	179
Quadro 14 – Relações entre termos no modelo de qualidade e na <i>WordNet</i>	181
Quadro 15 – Evidências - Parte 1	185
Quadro 16 – Evidências - Parte 2	186
Quadro 17 – Evidências - Parte 3	187
Quadro 18 – Medidas	188
Quadro 19 – Fontes de informação na análise de domínio	197
Quadro 20 – Normas selecionadas	198
Quadro 21 – Dados sobre projetos selecionados	199
Quadro 22 – Trecho de lista de termos	199
Quadro 23 – Elementos iniciais em <i>Objects</i>	202
Quadro 24 – Elementos iniciais em <i>Agents</i>	203
Quadro 25 – Elementos iniciais em <i>Attributes</i>	203
Quadro 26 – Elementos iniciais em <i>Activities</i>	204
Quadro 27 – Elementos iniciais em <i>Space</i>	204
Quadro 28 – Elementos iniciais em <i>Time</i>	204

Lista de abreviaturas e siglas

AAT	<i>Art & Architecture Thesaurus</i>
ABNT	Associação Brasileira de Normas Técnicas
AC	Antes de Cristo
ACM	<i>Association for Computing Machinery</i>
AHP	<i>Analytic Hierarchy Process</i>
AI	Arquitetura da Informação
ANSI	<i>American National Standards Institute</i>
ASQ	<i>American Society for Quality</i>
ASTIA	<i>Armed Services Technical Information Agency</i>
BFO	<i>Basic Formal Ontology</i>
BSI	<i>British Standards Institution</i>
CASE	<i>Computer-Aided Software Engineering</i>
CC	<i>Creative Commons</i>
CIESIN	<i>Consortium for International Earth Science Information Network</i>
CMMI	<i>Capability Maturity Model Integration</i>
CMMI-DEV	<i>Capability Maturity Model Integration for Development</i>
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
CRG	<i>Classification Research Group</i>
CWA	<i>Cognitive Work Analysis</i>
DAMA	<i>Data Management Association</i>
DCAP	<i>Dublin Core Application Profile</i>
DCC	<i>Dewey Decimal Classification</i>
DCMI	<i>Dublin Core Metadata Initiative</i>

DLESE	<i>Digital Library for Earth System Education</i>
DMBOK	<i>The Data Management Body of Knowledge</i>
DOAP	<i>Description of a Project</i>
DSP	<i>Description Set Profile</i>
EUA	Estados Unidos da América
EUP	<i>Enterprise Unified Process</i>
FAO	<i>Food and Agriculture Organization of the United Nations</i>
FCI	Faculdade de Ciência da Informação
FOAF	<i>Friend-of-a-Friend</i>
FODA	<i>Feature Oriented Domain Analysis Method</i>
GQM	<i>Goal Question Metric</i>
HTML	<i>Hypertext Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
IBM	<i>International Business Machines</i>
IEC	<i>International Electrotechnical Commission</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IHC	Interação Humano-Computador
IHMC	<i>Florida Institute for Human & Machine Cognition</i>
IRI	<i>International Resource Identifier</i>
ISACA	<i>Information Systems Audit and Control Association</i>
ISO	<i>International Organization for Standardization</i>
ISTQB	<i>International Software Testing Qualifications Board</i>
JODA	<i>Joint Object Oriented Domain Analysis</i>
KOS	<i>Knowledge Organization System</i>
LCC	<i>Library of Congress Classification</i>

LOD	<i>Linked Open Data</i>
MAAEN	<i>Multi-agent Application Engineering Methodology</i>
MADDEM	<i>Multi-agent Domain Engineering Methodology</i>
MAIA	Método de Arquitetura da Informação Aplicada
MER	Modelo de Entidade Relacionamento
MLM	<i>Metadata Lifecycle Model</i>
MODAL	<i>Metadata Objectives and principles, Domains and Architectural Layout</i>
NISO	<i>National Information Standards Organization</i>
OGC	<i>Office of Government Commerce</i>
OPA	<i>Organizational Process Asset</i>
ORG	<i>Ontology Research Group</i>
OSD	<i>Open Software Description Format</i>
OTKM	<i>On-To-Knowledge Methodology</i>
OWL	<i>Web Ontology Language</i>
PDCA	<i>Plan-Do-Check-Act</i>
PERT	<i>Program Evaluation Review Technique</i>
PMBOK	<i>Project Management Body of Knowledge</i>
PMI	<i>Project Management Institute</i>
PMEST	<i>Personality, Matter, Energy, Space, Time</i>
PPGCINF	Programa de Pós-Graduação em Ciência da Informação
PRO	<i>Protein Ontology</i>
RAS	<i>Reusable Asset Specification</i>
RDF	<i>Resource Description Framework</i>
RT	<i>Related term</i>
RUP	<i>Rational Unified Process</i>
SAA	<i>Society of American Archivists</i>

SABiO	<i>Systematic Approach for Building Ontologies</i>
SIOC	<i>Semantically-Interlinked Online Communities</i>
SKOS	<i>Simple Knowledge Organization System</i>
SOFTEX	Associação para Promoção da Excelência do Software Brasileiro
SP-OPL	<i>Software Process Ontology Pattern Language</i>
SPLC	<i>Software Project Life Cycle</i>
SPLCM	<i>Software Project Life Cycle Model</i>
SPLCP	<i>Software Project Life Cycle Process</i>
SWEBOK	<i>Software Engineering Body of Knowledge</i>
TGAI	Teoria Geral da Arquitetura da Informação
TOGAF	<i>The Open Group Architecture Framework</i>
UDC	<i>Universal Decimal Classification</i>
UF	<i>Used for</i>
UFO	<i>Unified Foundational Ontology</i>
UML	<i>Unified Modeling Language</i>
UnB	Universidade de Brasília
UP	<i>Unified Process</i>
URI	<i>Uniform Resource Identifier</i>
URL	<i>Uniform Resource Locator</i>
USDP	<i>Unified Software Development Process</i>
W3C	<i>World Web Consortium</i>
XML	<i>Extensible Markup Language</i>
XP	<i>Extreme Programming</i>

Sumário

	Introdução	23
I	PREPARAÇÃO DA PESQUISA	26
	Prólogo	27
1	ELEMENTOS DE PREPARAÇÃO DA PESQUISA	28
1.1	Processo de definição de tema e problema de pesquisa	28
1.2	Problema de pesquisa	30
1.3	Pergunta de pesquisa	30
1.4	Objetivos de pesquisa	30
1.4.1	Objetivo geral de pesquisa	30
1.4.2	Objetivos específicos de pesquisa	30
1.5	Potenciais usuários e usos dos resultados da pesquisa	31
1.6	Motivações e justificativas	31
1.7	Delimitação da pesquisa	34
1.8	Aspectos relativos ao texto e aos diagramas	34
1.9	Experiência prévia	35
2	METODOLOGIA DE PESQUISA	38
2.1	Classificação da pesquisa	38
2.2	Procedimentos técnicos	38
2.3	Percurso metodológico	39
2.4	Ferramentas usadas	40
2.5	Fontes de informação	42
II	REVISÃO DE LITERATURA	43
	Prólogo	44
3	CONCEITOS DE BASE	46
3.1	Informação	46
3.2	Recurso de informação	47
3.3	Coleção de recursos de informação	48
3.3.1	Espaço de informação	49
3.4	Recuperação da informação	49

3.4.1	Modos de recuperação da informação	50
3.4.2	Recuperação semântica da informação	52
3.4.3	Sistema de recuperação da informação	54
3.4.3.1	Avaliação de sistema de recuperação da informação	55
3.5	Organização da informação	57
3.5.1	Sistema de organização da informação	58
3.5.1.1	Conceito	60
3.6	Representação da informação	62
3.6.1	Análise de assunto	62
3.6.2	Linguagem para representação da informação	63
3.7	Classificação da informação	64
3.7.1	Classificação enumerativa da informação	66
3.7.2	Classificação facetada da informação	66
3.7.2.1	Teoria da Classificação Facetada da Informação	68
3.7.2.2	Análise de facetas	69
3.7.3	Classificação social da informação	72
3.8	Indexação da informação	74
3.9	Gestão da informação	75
4	ARQUITETURA DA INFORMAÇÃO	76
4.1	Definições de arquitetura	76
4.2	Definições de arquitetura da informação	76
4.3	Elementos da prática da arquitetura da informação	78
4.4	Elementos da disciplina Arquitetura da Informação	78
4.5	Elementos da Teoria Geral da Arquitetura da Informação	79
5	VOCABULÁRIOS CONTROLADOS	81
5.1	Definições de vocabulário controlado	81
5.2	Propósitos e usos de vocabulários controlados	81
5.3	Controle de vocabulário na recuperação da informação	82
5.4	Termos em vocabulários controlados	83
5.5	Relações em vocabulários controlados	84
5.5.1	Relação de equivalência	85
5.5.2	Relação hierárquica	86
5.5.3	Relação de associação	87
5.5.4	Relação semântica	87
5.6	Estruturas de vocabulários controlados	88
5.6.1	Arquivo de autoridade	88
5.6.2	Lista de termos	89
5.6.3	Esquema de classificação	89

5.6.4	Anel de sinônimos	91
5.6.5	Taxonomia	92
5.6.6	Tesauro	92
5.6.7	Ontologia	95
5.6.7.1	Elementos de ontologias	96
5.6.7.2	Propósitos e usos de ontologias	96
5.6.7.3	Classes de ontologias	97
5.7	Desenvolvimento de vocabulário controlado	101
5.7.1	Desenvolvimento de taxonomia	103
5.7.2	Desenvolvimento de tesauro	103
5.7.3	Desenvolvimento de ontologia	104
5.7.3.1	<i>Ontology Development 101</i>	107
5.7.3.2	<i>METHONTOLOGY</i>	107
5.7.3.3	<i>On-To-Knowledge Methodology</i>	108
5.7.3.4	<i>NeOn Methodology</i>	109
5.7.4	Ferramentas de desenvolvimento	111
5.8	Reuso de vocabulário controlado	112
5.9	Tecnologias de representação	112
5.9.1	<i>Resource Description Framework</i>	113
5.9.2	Dados ligados	114
5.9.3	<i>Simple Knowledge Organization System</i>	116
5.9.4	<i>Web Ontology Language</i>	117
5.10	Normas e padrões sobre vocabulário controlado	117
6	METADADOS	119
6.1	Definições de metadado	119
6.2	Propósitos e usos de metadado	120
6.3	Registro de metadado	122
6.4	Atributos de metadado	122
6.5	Qualidade de metadado	123
6.6	Classes de metadados	123
6.7	Esquema de metadados	124
6.8	Perfil de aplicação de metadados	125
6.9	Gestão de metadados	125
6.10	Desenvolvimento e uso de metadados	126
6.11	Anotação	128
6.12	Normas e padrões sobre metadados	129
6.12.1	<i>Dublin Core</i>	129
6.12.2	<i>IEEE Learning Object Metadata</i>	130

7	ELEMENTOS DE MODELAGEM	132
7.1	Definições de modelo	132
7.2	Propósitos e usos de modelos	133
7.3	Modelagem	133
7.3.1	Princípios de modelagem	133
7.4	Análise de domínio	133
7.4.1	Métodos de análise de domínio	135
7.4.1.1	<i>Feature Oriented Domain Analysis Method</i>	135
7.4.1.2	<i>Joint Object Oriented Domain Analysis</i>	135
8	ELEMENTOS DE QUALIDADE	137
8.1	Definições de qualidade e de conceitos correlatos	137
8.1.1	<i>Plan-Do-Check-Act</i>	138
8.2	Qualidade da informação	138
8.3	Qualidade de modelo conceitual	139
8.4	Qualidade de vocabulário controlado	140
8.4.1	Avaliação da qualidade de vocabulário controlado	141
8.5	Características e medidas de qualidade	142
8.5.1	<i>Goal Question Metrics</i>	143
8.5.2	Características e medidas de qualidade da informação	144
8.5.3	Características e medidas de qualidade de modelo conceitual	145
8.5.4	Características e medidas de qualidade de vocabulário controlado	146
8.6	Modelo de qualidade	147
8.6.1	Modelo de qualidade da informação	149
8.6.2	Modelo de qualidade de modelo conceitual	150
8.6.3	Modelo de qualidade de vocabulário controlado	151
9	ELEMENTOS DE ENGENHARIA DE SOFTWARE	154
9.1	Definições de engenharia de software	154
9.2	Necessidades de informação de engenheiros de software	155
9.3	Ciclo de vida de software	156
9.4	Processo de software	157
9.4.1	Processos de desenvolvimento de software	158
9.4.1.1	Processo Unificado de Desenvolvimento de Software	159
9.4.1.2	Processo ágil de desenvolvimento de software	160
9.4.1.3	Processo distribuído de desenvolvimento de software	161
9.5	Recursos de informação em ciclo de vida de software	162
9.6	Recuperação da informação em ciclo de vida de software	163
9.7	Representação da informação em ciclo de vida de software	165
9.8	Vocabulários controlados em Engenharia de Software	166

9.8.1	Esquemas de classificação da informação em Engenharia de Software	166
9.8.2	Taxonomias e ontologias em Engenharia de Software	167
9.9	Metadados em Engenharia de Software	169
9.10	Melhoria da qualidade de software	170
9.10.1	Modelos de qualidade de processo de software	170
9.10.1.1	Modelos de maturidade de capacidade	171
III	RESULTADOS	173
	Prólogo	174
10	PROPOSTA DE ELEMENTOS DE ARCABOUÇO	175
10.1	Considerações sobre a pesquisa bibliográfica	175
10.2	Definição de arcabouço de arquitetura da informação	175
10.3	Elementos de arcabouço	176
10.4	Contexto de aplicação e limites	177
10.5	Proposta de arquitetura de referência	178
10.6	Proposta de modelo de domínio	178
10.7	Proposta de modelo de qualidade	180
10.7.1	Pontos de vista	182
10.7.2	Características de qualidade	183
10.7.3	Evidências	184
10.7.4	Medidas	185
10.8	Proposta de lista de atividades	187
10.8.1	Grupo de atividades de requisitos	189
10.8.2	Grupo de atividades de desenho (<i>design</i>)	190
10.8.3	Grupo de atividades de implementação	191
10.8.4	Grupo de atividades de avaliação	191
10.8.5	Grupo de atividades de desenvolvimento de documentação	191
10.8.6	Grupo de atividades de importação de vocabulário controlado	191
10.8.7	Grupo de atividades de instalação	192
10.8.8	Grupo de atividades de manutenção	192
11	EXEMPLO DE USO DE ELEMENTOS PROPOSTOS	193
11.1	Contexto de aplicação e limites	193
11.2	Definição de processos	193
11.3	Resultados de atividades	194
11.3.1	Definição de domínio	195
11.3.2	Definição de idioma e garantia semântica	195
11.3.3	Definição de necessidades de informação	196

11.3.4	Seleção de fontes de informação	197
11.3.5	Construção de lista de termos	198
11.3.6	Definição de facetas	200
11.3.7	Atribuição de termos	200
11.3.8	Definição de relações	200
11.3.9	Definição de conceitos	202
11.3.10	Controle de qualidade	205
11.3.11	Publicação do vocabulário controlado	205
11.4	Sugestões para avaliação e desenvolvimento futuro	206

IV CONSIDERAÇÕES FINAIS 208

	Prólogo	209
--	--------------------------	------------

12	CONSIDERAÇÕES FINAIS	210
12.1	Alcance dos objetivos	210
12.2	Contribuições da pesquisa	210
12.3	Limitações da pesquisa	211
12.4	Sugestões para pesquisas futuras	211

	REFERÊNCIAS	213
--	------------------------------	------------

	Glossário	258
--	----------------------------	------------

APÊNDICES 270

	APÊNDICE A – LISTAS DE VOCABULÁRIOS CONTROLADOS	271
--	--	------------

A.1	Taxonomias em Engenharia de Software	271
A.1.1	Requisitos de Software	271
A.1.2	Desenho (<i>design</i>) de Software	272
A.1.3	Construção de Software	272
A.1.4	Manutenção de Software	273
A.1.5	Gestão de Engenharia de Software	273
A.1.6	Processo de Engenharia de Software	274
A.1.7	Qualidade de Software	274
A.1.8	Diversas áreas de conhecimento em Engenharia de Software	275
A.2	Ontologias em Engenharia de Software	275
A.2.1	Requisitos de Software	276

A.2.2	Desenho (<i>design</i>) de Software	276
A.2.3	Construção de Software	277
A.2.4	Teste de Software	277
A.2.5	Manutenção de Software	278
A.2.6	Gestão de Engenharia de Software	278
A.2.7	Processo de Engenharia de Software	278
A.2.8	Modelos e Métodos de Engenharia de Software	279
A.2.9	Qualidade de Software	280
A.2.10	Diversas áreas de conhecimento em Engenharia de Software	280

Introdução

Ao longo dos últimos séculos, a quantidade de fontes de informação disponíveis aumentou significativamente. A título de ilustração, cinquenta anos após a invenção da prensa de tipos móveis por Johannes Gutenberg, ocorrida por volta de 1440, cerca de vinte milhões de cópias haviam sido impressas. Cem anos após essa invenção, cerca de duzentos milhões de cópias haviam sido impressas (WRIGHT, 2014). Em 1481, o acervo da Biblioteca do Vaticano, uma das maiores da Europa na época, continha 3.498 manuscritos (VATICAN LIBRARY, 2017). Em 2017, existiam bilhões de páginas em sítios na *World Wide Web* (KUNDER, 2017) e o acervo da Biblioteca do Congresso dos EUA, a maior biblioteca do mundo nesse ano, continha milhões de itens (LIBRARY OF CONGRESS, 2017).

Em parte, devido ao número de fontes de informação disponíveis, tempo significativo é frequentemente gasto por pessoas e por organizações (empresas, instituições etc.) na recuperação da informação. A recuperação da informação engloba diversos elementos, muitas vezes falha ou resulta em informação que não é integralmente usada, por exemplo, por não ser informação relevante às necessidades (HAWKING, 2011; MOOERS, 1951). Parte significativa dos esforços e recursos em Ciência da Informação é voltada à solução de problemas relacionados à recuperação da informação (SARACEVIC, 1995). Na recuperação da informação, organização e representação da informação são processos importantes, e identificar conceitos relevantes em domínios de interesse é um desafio. Nesse contexto, o uso de informação semântica pode ter impacto positivo. Informação semântica é informação sobre significados de símbolos e de relações entre conceitos, e pode ser provida por ferramentas semânticas (HJØRLAND, 2007c; SOERGEL et al., 2004). Ferramenta é algo tangível usado na realização de atividade e voltado a produzir produto ou resultado (PMI, 2013).

Informação semântica pode ser provida por vocabulário controlado. Existem várias definições para o termo “vocabulário controlado”. Nesta tese, esse termo designa classe de sistema de organização de conhecimento ou classe de estrutura para organização de conhecimento (ABBAS, 2010; HODGE, 2000; NISO, 2005). Além de prover suporte à recuperação da informação, vocabulários controlados podem ser usados com várias outras finalidades, por exemplo, podem ser usados para prover suporte ao entendimento de domínio, e para promover a uniformização de termos e conceitos. Considerando a importância de vocabulários controlados, nesta tese são propostos elementos de arcabouço para ciclo de vida de projeto de vocabulário controlado. Esses elementos têm o objetivo de prover suporte a processo de desenvolvimento e a processo de avaliação em ciclo de vida de projeto de vocabulário controlado. Nesse contexto, o termo arcabouço designa estrutura

que estabelece conceitos e vocabulário para conjunto de teorias explicativas. Um arcabouço provê estrutura para o entendimento de algo sendo apresentado ou explorado e pode englobar elementos relevantes e questões em área de conhecimento, indústria, empresa etc. É importante destacar que, em certos domínios, o desenvolvimento de arcabouço é uma abordagem de pesquisa para construção de teorias (ABBAS, 2010; GLUSHKO, 2013; PORTER, 1991).

Considerando o modelo descrito em Albuquerque e Lima-Marques (2011), os elementos propostos nesta tese integram um arcabouço de arquitetura da informação para ciclo de vida de projeto de vocabulário controlado. Esta tese está organizada em quatro partes principais: preparação da pesquisa, revisão de literatura, resultados e considerações finais.

- (a) Parte I (Preparação da pesquisa): a primeira parte desta tese, composta pelos capítulos 1 e 2, aborda a preparação da pesquisa. O capítulo 1 aborda processo de definição de tema de pesquisa, processo de definição de problema de pesquisa, tema de pesquisa, problema de pesquisa, pergunta de pesquisa, objetivo geral de pesquisa, objetivos específicos de pesquisa, potenciais usuários e usos dos resultados da pesquisa, motivações da pesquisa, justificativas da pesquisa, delimitação da pesquisa, aspectos relativos ao texto e aos diagramas desta tese, aspectos relativos à experiência prévia do autor desta tese em pesquisas correlatas. O capítulo 2 aborda classificação da pesquisa, procedimentos técnicos, percurso metodológico, ferramentas e fontes de informação usadas durante a realização da pesquisa.
- (b) Parte II (Revisão de literatura): a segunda parte desta tese, composta pelos capítulos 3, 4, 5, 6, 7, 8 e 9, contém o resultado da revisão de literatura. O capítulo 3 descreve conceitos de base relevantes no contexto desta tese, em especial, informação, recurso de informação, coleção de recursos de informação, recuperação da informação, organização da informação, representação da informação, classificação da informação, indexação da informação e gestão da informação. Esse capítulo tem o objetivo de prover fundamentação conceitual e promover consistência no uso de termos e conceitos. O capítulo 4 contém definições do termo “arquitetura da informação” e informação sobre a prática da arquitetura da informação, a disciplina Arquitetura da Informação e a Teoria Geral da Arquitetura da Informação. O capítulo 5 contém definições do termo “vocabulário controlado” e informação sobre propósitos, usos, elementos, estruturas, desenvolvimento e reuso de vocabulários controlados. O capítulo 6 contém definições do termo metadado e aborda propósitos, usos, desenvolvimento e gestão de metadados. O capítulo 7 contém definições dos termos “modelo”, “modelo conceitual” e “modelo de domínio” e informação sobre propósitos e usos de modelos, modelagem e análise de domínio. O capítulo 8 aborda qualidade, garantia de qualidade, controle de qualidade, característica de qualidade, medida de qualidade, qualidade da informação,

qualidade de vocabulário controlado, qualidade de modelo conceitual e modelo de qualidade. Finalmente, o capítulo 9 aborda o domínio para o qual foi desenvolvido o protótipo de vocabulário controlado. Esse capítulo contém definições do termo “engenharia de software” e discute necessidades de informação de engenheiros de software, recursos de informação e processos em ciclo de vida de software, repositório de software, recuperação da informação em repositório de software, vocabulários controlados e metadados em Engenharia de Software, melhoria da qualidade de software.

- (c) Parte III (Resultados): a terceira parte desta tese, composta pelos capítulos 10 e 11, contém proposta de elementos de arcabouço de arquitetura da informação para ciclo de vida de projeto de vocabulário controlado e exemplo de uso desses elementos. O capítulo 10 contém proposta do arcabouço, em especial, proposta de arquitetura de referência, modelo de domínio, modelo de qualidade e lista de atividades. No capítulo 11 é descrito exemplo de uso do arcabouço proposto no processo de desenvolvimento e no processo de avaliação de um protótipo de vocabulário controlado para o domínio da Engenharia de Software.
- (d) Parte IV (Considerações finais): a quarta parte desta tese, composta pelo capítulo 12, contém considerações finais sobre a pesquisa cujos resultados são descritos nesta tese. Esse capítulo contém considerações sobre contribuições da pesquisa, limitações da pesquisa e sugestões de possíveis pesquisas futuras.

Além das partes anteriormente relacionadas, também integram esta tese glossário, apêndice e referências. O glossário é composto por 104 entradas e contém definições de termos relevantes no contexto desta tese. As fontes dessas definições foram identificadas em pesquisa bibliográfica realizada pelo autor desta tese. O apêndice A contém informação sobre vocabulários controlados estruturados como taxonomias e ontologias no domínio da Engenharia de Software. Essa informação resulta de mapeamento sistemático de literatura realizado pelo autor desta tese. Esse apêndice foi construído com a intenção de prover suporte ao desenvolvimento de vocabulários controlados no domínio da Engenharia de Software. Nas referências, são identificadas as fontes de informação que são usadas nesta tese.

Finalmente, vale destacar a característica exploratória da pesquisa que resultou nesta tese e a importância da pesquisa bibliográfica nesse contexto. Também vale destacar que interdisciplinaridade é característica de diversas pesquisas em Ciência da Informação (SARACEVIC, 1995), e que essa foi uma característica da pesquisa que resultou nesta tese. Nesta pesquisa, além das contribuições originadas na disciplina Ciência da Informação, também foram relevantes contribuições originadas na disciplina Ciência da Computação.

Parte I

Preparação da pesquisa

Prólogo

Esta parte da tese é composta por capítulos que têm o objetivo de prover informação sobre a preparação da pesquisa e sobre a metodologia adotada na pesquisa realizada. O capítulo 1 é dedicado aos elementos de preparação da pesquisa e contém informação sobre processos usados na definição do tema e do problema de pesquisa, problema e pergunta de pesquisa, objetivo geral e objetivos específicos de pesquisa, potenciais usuários e usos dos resultados da pesquisa, motivações, justificativas e delimitação da pesquisa, aspectos relativos ao texto e aos diagramas desta tese. Também apresenta aspectos relativos à experiência prévia do autor desta tese em pesquisas correlatas. O capítulo 2 é dedicado à metodologia adotada na pesquisa que resultou nesta tese e contém informação sobre classificação da pesquisa, procedimentos técnicos usados, percurso metodológico seguido, ferramentas e fontes de informação usadas durante a realização da pesquisa que resultou nesta tese.

1 Elementos de preparação da pesquisa

Este capítulo descreve elementos de preparação da pesquisa que resultou nesta tese, tais como processo de definição de tema e problema de pesquisa, problema de pesquisa, pergunta de pesquisa, objetivo geral de pesquisa, objetivos específicos de pesquisa, potenciais usuários e usos dos resultados da pesquisa, motivações e justificativas de pesquisa, delimitação da pesquisa, aspectos relativos ao texto e aos diagramas desta tese. Por fim, são apresentados aspectos relativos à experiência prévia do autor desta tese em pesquisas correlatas.

1.1 Processo de definição de tema e problema de pesquisa

No processo de definição de tema e problema de pesquisa, inicialmente ocorreu a escolha do tema de pesquisa, baseada em conhecimento, experiência prévia e interesse do autor desta tese. Entre as motivações originais, destacou-se o interesse do autor desta tese por pesquisa sobre arquitetura da informação no contexto da Engenharia de Software. Como o tema de pesquisa “arquitetura da informação em engenharia de software”, era abrangente, foi realizado levantamento bibliográfico para refinar o tema e definir o problema de pesquisa. O levantamento bibliográfico foi realizado por meio de estudo exploratório que identificou temas de pesquisa potencialmente relevantes à disciplina Arquitetura da Informação. Esses temas de pesquisa foram identificados por análise de diversas fontes de informação sobre essa disciplina, por exemplo: [The Information Architecture Institute \(2013\)](#), [Morville e Rosenfeld \(2006\)](#), [Dillon \(2002\)](#), [Hagedorn \(2000\)](#), [Ding e Lin \(2009\)](#), [Halvorson e Rach \(2012\)](#), [Toms \(2002\)](#), [Spencer e Featherstone \(2010\)](#), [Lima-Marques \(2011\)](#), [Taylor e Joudrey \(2009\)](#), [Albuquerque e Lima-Marques \(2011\)](#) e [Duarte \(2011\)](#). A análise das diversas fontes de informação possibilitou a identificação dos seguintes potenciais temas de pesquisa: acessibilidade; classificação da informação; comportamento de pesquisa; avaliação, desenho (*design*), implementação, integração e representação de espaço de informação; *findability*; usabilidade; gestão da informação; metadado; necessidade de informação; organização da informação; representação da informação; sistema de busca; sistema de navegação; sistema de organização da informação; taxonomia; tesouro; e ontologia.

Após a identificação de temas de pesquisa potencialmente relevantes à disciplina Arquitetura da Informação, foi realizado mapeamento sistemático de literatura para identificar quais desses temas se destacavam em pesquisas no domínio da Engenharia de Software e quais fontes de informação deveriam ser acessadas em estudo exploratório. Esse mapeamento foi realizado por processo similar ao descrito em [Petersen et al. \(2008\)](#). As

atividades nesse processo foram as seguintes: definir consultas, selecionar bases de dados, construir enunciados das consultas às bases de dados, definir critérios para seleção de fontes de informação, consultar bases de dados, selecionar fontes de informação dentre as localizadas nas bases de dados, armazenar dados sobre as fontes de informação selecionadas, realizar medidas, e construir taxonomia composta por termos identificados nas fontes de informação acessadas. Nesse processo, foram consultadas diversas bases de dados, por exemplo: ABNT Coleção, *ACM Digital Library*, *IEEE Xplore Digital Library*, *ScienceDirect*, *Scopus Database*, *Web of Science*, *Wiley Online Library*, e Portal de Periódicos CAPES. Foram também usados sistemas de busca como *Yahoo!* e *Google*. As fontes acessadas foram principalmente das seguintes classes: artigo em periódico científico, trabalho publicado em conferência científica, dissertação de mestrado, tese de doutorado, livro, norma técnica, guia de melhores práticas. Para construir os enunciados das consultas, foram selecionadas palavras-chave identificadoras dos temas de pesquisa. Os enunciados foram construídos e executados usando-se essas palavras-chave e operadores booleanos. Os resultados foram armazenados por meio de uma ferramenta para gerenciamento de referências bibliográficas.

No processo de mapeamento sistemático de literatura anteriormente descrito, foi armazenada informação sobre 2.298 fontes. A partir dos resultados armazenados, foram realizadas medidas com o objetivo de identificar temas frequentemente abordados em pesquisas relatadas nas fontes de informação localizadas. A seguir, são listados percentuais dessas fontes por tema de pesquisa (estão listados apenas aqueles temas de pesquisa que estavam presentes em mais de 5% das fontes de informação localizadas no mapeamento): 33,2% ontologia, 22,1% usabilidade, 15% gestão da informação, 9,6% metadado e 9,4% taxonomia. Considerando esses resultados e os interesses do autor desta tese, foram selecionadas, acessadas e analisadas fontes de informação sobre os temas ontologia, metadado e taxonomia.

A partir da análise das fontes de informação selecionadas, foram possíveis as seguintes constatações preliminares: ocorrem dificuldades na recuperação da informação em diversos contextos, o uso de informação semântica pode melhorar a recuperação da informação, informação semântica pode ser provida por modelo de domínio, modelo de domínio pode ser representado por vocabulário controlado, existem diversas classes de vocabulários controlados. Também se constatou preliminarmente que podem ocorrer dificuldades em desenvolvimento e avaliação de vocabulário controlado, em decorrência de fatores como os seguintes: falta de diretrizes para seleção de métodos, integração de métodos, incorporação de métodos a processos, seleção de técnicas de aquisição de conhecimento, avaliação da qualidade; processos ou métodos com falta de embasamento teórico ou níveis inadequados de formalização; processos ou métodos compostos por atividades não detalhadas ou descritas de modo impreciso. A partir dessas constatações, o problema de pesquisa foi então descrito. Seguindo recomendação em Gil (2010), além de descrito, o problema de pesquisa foi definido sob a forma de pergunta. Para esclarecer os

resultados esperados da pesquisa, foram definidos objetivo geral e objetivos específicos da pesquisa.

1.2 Problema de pesquisa

No ciclo de vida de projeto de vocabulário controlado, diversidade de métodos, carência de fundamentação teórica, diversidade de enfoques, falta de descrições de atividades, descrições imprecisas de atividades, descrições não detalhadas de atividades, carência de diretrizes sobre como garantir qualidade e carência de diretrizes sobre como escolher entre métodos, podem dificultar a execução de processos de desenvolvimento e de avaliação de vocabulário controlado.

1.3 Pergunta de pesquisa

Qual arcabouço conceitual usar em ciclo de vida de projeto de vocabulário controlado com o objetivo de prover suporte a processo de desenvolvimento e a processo de avaliação de vocabulário controlado?

1.4 Objetivos de pesquisa

A seguir, são descritos objetivo geral e objetivos específicos de pesquisa.

1.4.1 Objetivo geral de pesquisa

Propor elementos de um arcabouço conceitual para suporte a processo de desenvolvimento e a processo de avaliação em ciclo de vida de projeto de vocabulário controlado.

1.4.2 Objetivos específicos de pesquisa

- (a) Identificar e descrever conceitos, métodos, atividades, processos, teorias, normas, guias de melhores práticas, ferramentas e modelos de qualidade sobre vocabulário controlado.
- (b) Propor elementos de um arcabouço conceitual para suporte a processo de desenvolvimento e a processo de avaliação em ciclo de vida de projeto de vocabulário controlado.
- (c) Escolher domínio de aplicação onde ocorrerá exemplo de uso de elementos do arcabouço conceitual proposto e descrever conceitos relevantes ao entendimento do exemplo.

- (d) Exemplificar o uso de elementos do arcabouço conceitual proposto por meio da construção de um protótipo de vocabulário controlado sobre o domínio de aplicação escolhido.

1.5 Potenciais usuários e usos dos resultados da pesquisa

Educadores, estudantes, engenheiros de software, arquitetos de processo e profissionais da informação são potenciais usuários dos resultados da pesquisa que resultou nesta tese. Os resultados desta pesquisa podem ser usados em pesquisa e ensino em Ciência da Informação e em Engenharia de Software; ciclo de vida de projeto de vocabulário controlado; definição de conteúdo, semântica, formato e meio para representar, reter, transmitir e recuperar informação. Em Engenharia de Software, os resultados desta pesquisa podem ser usados na melhoria da qualidade de modelos; na implementação de práticas necessárias a certos níveis de maturidade de processos em ciclo de vida de software; na normalização de termos (conversão de termos em um termo) usados em projetos de software; no desenvolvimento de processos usados em ciclo de vida de software; em tarefa como “definir conteúdo, semântica, formatos e meios para representar, reter, transmitir e recuperar informação”, tarefa essa definida em ISO/IEC 12207:2008: *Systems and software engineering - Software life cycle processes* e integrante do “Processo de Gestão da Informação” (*Information Management Process*); e em sistema de recuperação da informação em ciclo de vida de software.

1.6 Motivações e justificativas

Pessoas e organizações (empresas, instituições etc.) gastam tempo significativo na procura por informação. Em espaços de informação digitais, a necessidade de melhorar o processo de navegação e as experiências dos usuários torna relevante pesquisar sobre ferramentas semânticas que possam ser usadas, por exemplo, na construção de interfaces com usuários, uma vez que interfaces compostas apenas por caixas de busca e por listas de resultados podem ser inadequadas à exploração de espaços de informação complexos. Na recuperação da informação na rede Internet, onde a baixa qualidade da indexação pode resultar na recuperação de grande quantidade de informação e de informação irrelevante, disponibilizar informação semântica pode melhorar o processo de recuperação da informação (HJØRLAND; PEDERSEN, 2005; LASSILA, 1997; MARCONDES; SAYÃO, 2001; NISO, 2007; SHIRI, 2012; STOICA; HEARST, 2007). Em projetos de engenharia, também é um desafio prover informação relevante. Nesses projetos, engenheiros frequentemente precisam acessar diversos recursos de informação. Muitas vezes, esses recursos de informação são desenvolvidos por equipes cujos membros estão distribuídos geograficamente e/ou temporalmente. Por sua vez, os recursos de informação podem ter representações diferentes

e estar armazenados em sistemas variados. Nesses recursos de informação, a terminologia usada pode não ser completamente compartilhada, o que pode dificultar a interpretação da informação e a sua recuperação por meio de comparação de palavras-chave. Para minimizar essas dificuldades, uma alternativa é organizar os recursos por meio de esquemas de classificação capazes de prover suporte a equipes compostas por engenheiros com diferentes pontos de vista e com diferentes necessidades de informação (GIESS; WILD; MCMAHON, 2008).

A importância da definição de conteúdo, semântica, formatos e meios para representar, reter, transmitir e recuperar informação em ciclo de vida de software, é reconhecida em diversas fontes. Por exemplo, na fonte ISO (2008a) são previstos processos com essas responsabilidades. Facilitar a recuperação da informação contida em repositórios de software pode ter impacto positivo no desenvolvimento de software, contribuir na gestão e no reuso de software, promover a realização de estudos empíricos sobre desenvolvimento de software, reduzir barreiras à participação em projeto de software, promover modo de aprendizado onde conhecimentos e habilidades são usados e ensinados em contextos realistas, facilitar a educação e o treinamento de engenheiros de software, e contribuir em processos de inovação e aperfeiçoamento de produtos e serviços (SHAW, 2000). Particularmente no reuso de recursos em desenvolvimento de software, pode melhorar a produtividade em projetos de software e a qualidade de produtos. Em projetos de software, é importante recuperar informação em modelos de software, pois neles pode haver informação relevante a outros projetos. Reusar apenas código de software pode não ser suficiente. Apesar da importância do reuso em projetos de software, as ferramentas usadas muitas vezes realizam apenas procura por palavra-chave e são desenvolvidas para uso genérico ou para procurar apenas por códigos ou por componentes de software. Ferramentas para recuperar informação em outras classes de recursos são menos frequentes. Em parte, isso decorre de dificuldades na representação da informação em certas classes de recursos de informação (BISLIMOVSKA et al., 2012; ROBLES et al., 2012).

Ao longo do ciclo de vida de software, engenheiros de software executam diversas atividades. Em processo de desenvolvimento de software, é gerada informação variada, por exemplo, informação sobre requisitos, informação sobre projeto, código fonte, documentação de código, informação sobre histórico de decisões de projeto, plano de teste e manuais destinados a usuários. A natureza global de vários projetos de software, assim como a complexidade e o tamanho de muitos desses projetos, resultam em grande quantidade de informação. Nos processos em ciclo de vida de software, acesso à informação é, portanto, relevante aos engenheiros de software. Nesses processos, a procura por informação é atividade frequente e que consome tempo significativo de trabalho desses engenheiros (GRZYWACZEWSKI; IQBAL, 2012; MARCUS; MENZIES, 2010; PIETO-DIAZ, 1990). Pesquisa com engenheiros de software indica que eles gastam tempo significativo na procura por informação e no acesso à informação. Engenheiros de software frequentemente

encontram dificuldades na formulação de consulta que melhore a precisão da recuperação da informação, assim como na avaliação da qualidade de fontes de informação acessadas. Nesse contexto, ocorrem dificuldades na recuperação da informação devido a fatores variados, tais como grande quantidade e variedade de fontes de informação, dispersão de informação em várias fontes e instabilidade de fontes (FREUND; TOMS; WATERHOUSE, 2005).

Em ciclo de vida de software, ocorre desenvolvimento, operação e manutenção de software por meio de processos complexos e intensivos na aquisição, produção e uso da informação. Nesse ciclo, processos de gestão, recuperação e reuso da informação podem ser dificultados pelo tratamento não sistemático dado à informação, pela pouca estruturação da informação, pela adoção de práticas limitadas na representação e organização da informação, pela não consideração de semântica na recuperação da informação, e pela não adoção de um conjunto comum de conceitos e termos. Dificuldades em gerir, recuperar e reusar informação podem impactar negativamente a produtividade de projetos de software e a qualidade de produtos de software (ANKOLEKAR; HERBSLEB; SYCARA, 2003; BANI-SALAMEH; JEFFERY; AL-GHARAIBEH, 2010; BERGAMASCHI; MARTOGLIA; SORRENTINO, 2014; BINKLEY; LAWRIE, 2010; CALERO; RUIZ; PIATTIN, 2006; DILLON; SIMMONS, 2008; FALBO; BERTOLLO, 2005; FUGGETTA, 2000; GALIN, 2003; GONZÁLEZ-PÉREZ; HENDERSON-SELLERS, 2006; THE STANDISH GROUP, 2013; JOHSON; BLAIS, 2009; PERRY; STAUDENMEYER; VOTTA, 1994; PRIETO-DIAZ, 1990; PRESSMAN, 2011; SARMA, 2005; SCHACH, 2008; SIMMONS; DILLON, 2006; SOMMERVILLE, 2010; CMMI PRODUCT TEAM, 2010; ZHAO; DONG; PENG, 2009).

Garantir produtividade e qualidade em projetos de software não é tarefa simples. Nesse contexto, o termo produtividade designa relação entre produto do trabalho e esforço do trabalho (ISO, 2010). Por sua vez, o termo qualidade designa grau com o qual um produto ou processo atende a requisitos estabelecidos (IEEE, 2014b). Projetos de software frequentemente não atingem plenamente as suas metas de produtividade e qualidade. Em decorrência disso, comportamentos indesejáveis, que podem resultar em diversos problemas e danos, podem ser apresentados por produtos desses projetos (BERGAMASCHI; MARTOGLIA; SORRENTINO, 2014; FUGGETTA, 2000; THE STANDISH GROUP, 2013). Produtividade e qualidade em projetos de software são influenciadas, entre outros fatores, por processos em ciclo de vida de software e adoção de conjunto comum de conceitos e de termos para designar conceitos (FALBO; BERTOLLO, 2005; GALIN, 2003; GONZÁLEZ-PÉREZ; HENDERSON-SELLERS, 2006; CMMI PRODUCT TEAM, 2010). Também vale destacar que a informação em repositórios de software registra conhecimento acumulado em projetos de software. Facilitar a descoberta dessa informação tende a aumentar a produtividade e a qualidade em projetos de software. Além disso, o acesso a essa informação pode contribuir, por exemplo, na educação e na realização de pesquisas em

Engenharia de Software. Finalmente, o desenvolvimento de recursos terminológicos sobre o corpo de conhecimento da disciplina Engenharia de Software, embasados em especificações endossadas por organizações renomadas e tidas como referência, pode contribuir na melhoria do aprendizado em Engenharia de Software, reduzir problemas no desenvolvimento de ferramentas e na interoperabilidade entre elas (HILERA; FERNÁNDEZ-SANZ, 2010; JIMENO-YEPES et al., 2009).

1.7 Delimitação da pesquisa

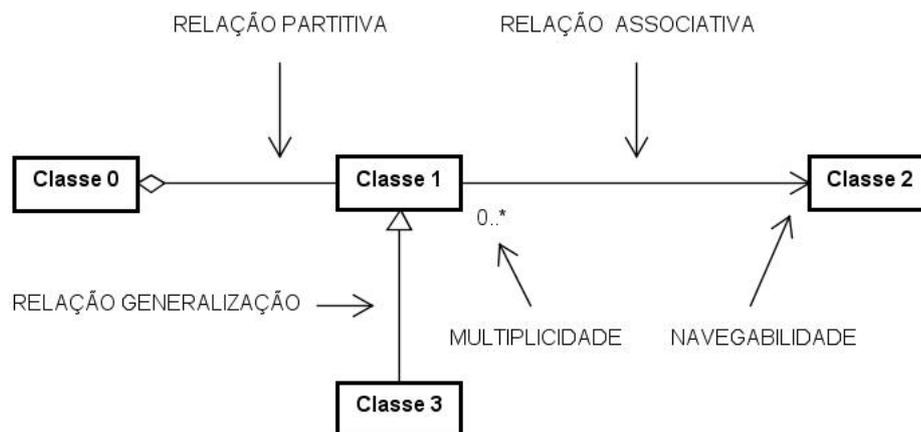
Considerando a abrangência e a característica interdisciplinar da pesquisa que resultou nesta tese, é importante destacar que essa pesquisa não teve os seguintes escopos e/ou objetivos: (a) avaliar ou comparar teorias propostas por distintas correntes filosóficas; (b) desenvolver, avaliar ou comparar produtos de software; (c) desenvolver, avaliar ou comparar produtos de hardware; (d) desenvolver, avaliar ou comparar algoritmos para recuperação da informação; (e) abranger todas as classes de vocabulários controlados; (f) abranger todas as classes de recursos de informação usados em ciclo de vida de software; (g) abranger todas as áreas de conhecimento integrantes da disciplina Engenharia de Software.

1.8 Aspectos relativos ao texto e aos diagramas

Com o objetivo de minimizar mal-entendidos, muitos termos em português estão acompanhados por termos nos idiomas adotados nas fontes referenciadas; na versão digital desta tese, a cor azul identifica hiperligações; com o objetivo de facilitar a navegação na versão digital desta tese, nas referências estão incluídas hiperligações para citações presentes no texto desta tese; considerando que o tema abordado nesta tese é interdisciplinar e que pode ser de interesse de leitores com formação em diversas disciplinas, nesta tese são descritos conceitos de base com o objetivo de promover consistência no uso dos mesmos e prover fundamentação conceitual, sendo também apresentados diversos exemplos, apesar disso poder causar didatismo excessivo em trechos desta tese; parte desta tese segue abordagem prescritiva (texto contendo definição de atividades e de recursos para uso por outros); com o objetivo de reduzir ambiguidades, sempre que conveniente, definições de conceitos originadas em normas técnicas publicadas são apresentadas; diversas normas de organizações reconhecidas foram usadas nas definições de conceitos; com o objetivo de reduzir ambiguidades, foram construídos diversos diagramas de classe representando conceitos e relações entre eles; termos que constam de dicionário da língua portuguesa foram utilizados sem grifo, e foram grafados em itálico os termos que não constam. Em parte devido ao objetivo da pesquisa, o texto desta tese, em certos trechos, privilegia amplitude e não detalhamento. Finalmente, nesta tese foram construídos diagramas usando-se a

linguagem de modelagem *Unified Modeling Language* (UML). A figura 1 contém elementos da UML usados nesses diagramas.

Figura 1 – Elementos da linguagem de modelagem UML



Fonte: Elaborado pelo autor

Quando figura, tabela ou quadro foi elaborado pelo autor desta tese, esse fato foi explicitado pelo texto “Elaborado pelo autor”. Quando figura, tabela ou quadro foi elaborado pelo autor desta tese, por adaptação de figura, tabela ou quadro de alguma fonte de informação, esse fato foi explicitado pelo texto “Adaptado de”, seguido por referência para a fonte de informação. Quando figura, tabela ou quadro foi reusado de alguma fonte de informação, esse fato foi explicitado por referência à fonte de informação. Em decorrência da abrangência dos assuntos abordados, da pesquisa ter adotado uma abordagem exploratória e da quantidade de fontes de informação consultadas, há diversas referências nesta tese. O autor desta tese acredita que essas referências poderão contribuir para melhorar o entendimento de seu conteúdo, assim como contribuir para futuras pesquisas correlatas.

1.9 Experiência prévia

O autor desta tese atuou por anos em engenharia de software, particularmente em atividades de modelagem e garantia da qualidade de software. Quanto ao envolvimento do autor desta tese em pesquisas correlatas, destaca-se a pesquisa sobre arcabouços de arquitetura da informação (*information architecture framework*), cujos resultados são descritos em [Albuquerque \(2014\)](#). Entre esses resultados, há um arcabouço de arquitetura da informação composto por processos e roteiros. Nesse contexto, roteiro é um conjunto de passos para a execução de processo ([JACKA; KELLER, 2009](#); [PMI, 2013](#)). Nessa pesquisa,

após o desenvolvimento do arcabouço de arquitetura da informação, ele foi usado, pelo autor desta tese, na descrição da arquitetura da informação em projeto de software aberto (*open source software*) realizado por comunidade em linha (*online community*) na rede Internet.

O arcabouço de arquitetura da informação resultante da pesquisa descrita em [Albuquerque \(2014\)](#) foi desenvolvido a partir de consulta a diversas fontes, muitas delas acerca do *The Open Group Architecture Framework* (TOGAF), do *Project Management Body of Knowledge Guide* (PMBOK) ou do Método de Arquitetura da Informação Aplicada (MAIA). TOGAF é um arcabouço de arquitetura composto por métodos e ferramentas para produção, uso e manutenção de arquiteturas nos domínios de negócios, dados, aplicação e tecnologia ([The Open Group, 2011](#)). PMBOK é um arcabouço composto por descrições de princípios para gerenciamento de projetos, definições de conceitos relacionados ao gerenciamento de projetos, descrições de ciclos de vida de projeto e de gerenciamento de projetos, descrições de processos ([PMI, 2013](#)). MAIA é um método de arquitetura da informação no qual as atividades são distribuídas em momentos denominados Escutar, Pensar, Construir e Habitar. Esses momentos são compostos por atos e ocorrem de modo sequencial e cíclico. Em cada momento, o sujeito executa ações sobre o espaço de informação até chegar ao fim do ciclo. Esses ciclos se sucedem, resultando na evolução da arquitetura da informação ([COSTA, 2009](#)). Na pesquisa descrita em [Albuquerque \(2014\)](#), a escolha do TOGAF como arcabouço a ser configurado decorreu, principalmente, das seguintes características desse arcabouço: adotado em diversas organizações, padrão de consórcio composto por várias organizações, genérico e passível de configuração, suporte a vários domínios. O PMBOK foi escolhido como arcabouço a ser configurado, principalmente, por ser um guia de melhores práticas em gerenciamento de projetos internacionalmente reconhecido e usado por diversas organizações. Outro importante fator que influenciou a escolha desses arcabouços foi a facilidade de acesso a fontes de informação sobre eles. Finalmente, MAIA foi escolhido por ser um método de arquitetura da informação adaptável, desenvolvido com base em pressupostos científicos, que enfatiza a representação de espaços de informação por taxonomias e ontologias, e que pode ser usado em contexto organizacional.

Entre as conclusões resultantes da pesquisa descrita em [Albuquerque \(2014\)](#), algumas são relevantes no contexto desta tese. Por exemplo, as seguintes: (a) existem arcabouços e métodos de arquitetura da informação desenvolvidos para diversos domínios; (b) fontes de informação sobre arcabouços de arquitetura da informação apresentam níveis de detalhamento e de coerência variáveis; (c) em fontes de informação sobre alguns arcabouços de arquitetura da informação, são encontrados conceitos com definições imprecisas ou designados por mais de um termo; (d) alguns arcabouços de arquitetura da informação são coleções de boas práticas com poucas referências a trabalhos científicos; (e) o desenvolvimento de artefatos prescritos em arcabouços de arquitetura da informação pode requerer

diversas ferramentas; (f) o desenvolvimento de artefatos prescritos em arcabouços de arquitetura da informação pode requerer esforço significativo; (g) arcabouços de arquitetura da informação podem contribuir na organização da informação em ciclo de vida de software.

2 Metodologia de pesquisa

A adoção de metodologias facilita a repetição de processos e torna o sucesso de projetos menos dependente de experiências prévias em projetos similares (FERNÁNDEZ-LÓPEZ; GÓMEZ-PÉREZ, 2002). Nesse contexto, metodologia é sistema de métodos a ser usado em classe de trabalho intelectualmente intensivo (IEEE, 1996; PMI, 2013); sistema é coleção de componentes organizada para função ou para conjunto de funções, combinação de componentes inter-relacionados organizados para alcançar um ou mais propósitos definidos (ISO, 2010); e método é abordagem definida e reproduzível para tratar certo tipo de problema (The Open Group, 2013). Este capítulo descreve a metodologia adotada na pesquisa que resultou nesta tese, e contém informação sobre classificação dessa pesquisa, procedimentos técnicos adotados, percurso metodológico seguido, ferramentas usadas na pesquisa e fontes de informação acessadas no processo de revisão de literatura.

2.1 Classificação da pesquisa

Esta pesquisa ocorreu no domínio da Ciência da Informação, na área de conhecimento denominada Ciências Sociais Aplicadas pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). Considerando informação em Gil (2010), esta pesquisa pode ser classificada, quanto à finalidade, como aplicada, pois visa à aquisição de conhecimento para aplicação em situação específica, para a solução de problema específico. Quanto aos objetivos mais gerais, esta pesquisa pode ser classificada como exploratória e explicativa. Segundo Gil (2010), pesquisa exploratória tem o propósito de proporcionar maior familiaridade com o problema com vistas a torná-lo mais explícito ou a construir hipótese e, geralmente, envolve levantamento bibliográfico, entrevistas e análise de exemplos. Portanto, esta pesquisa pode ser classificada como exploratória, pois teve o objetivo inicial de proporcionar maior familiaridade com o problema de modo a torná-lo mais explícito. Considerando que, segundo Gil (2010), pesquisa explicativa tem o propósito de identificar fatores que determinam ou contribuem para a ocorrência de fenômenos, esta pesquisa também é explicativa.

2.2 Procedimentos técnicos

Os procedimentos técnicos adotados na pesquisa foram pesquisa bibliográfica, construção de modelo e construção de protótipo. Na parte II (Revisão de literatura) desta tese, se destacou a realização de pesquisa bibliográfica. Esse procedimento foi executado, principalmente, por meio do acesso a fontes de informação tais como: artigo em periódico

científico, trabalho publicado em conferência científica, dissertação de mestrado, tese de doutorado, livro, norma técnica e guia de melhores práticas. Na Parte III (Resultados), os procedimentos técnicos usados foram a construção de modelo e a construção de protótipo.

2.3 Percurso metodológico

Para alcançar os objetivos da pesquisa, foi realizado projeto decomposto em fases em que foram agrupadas atividades logicamente relacionadas. Cada fase culminou com a conclusão de entregas, enquanto as atividades em cada fase, prescritas em métodos selecionados, integraram processos. A estruturação do projeto em fases teve o objetivo de facilitar o gerenciamento, o planejamento e o controle do projeto. O modelo de ciclo de vida do projeto foi definido a partir do modelo de ciclo de vida genérico descrito na fonte [PMI \(2013\)](#), guia internacionalmente reconhecido de melhores práticas em gerenciamento de projetos. O modelo de ciclo de vida genérico foi escolhido por ser possível mapear qualquer projeto para esse modelo ([PMI, 2013](#)). Os nomes das fases presentes nesse modelo de ciclo de vida de projeto são: Iniciação, Organização e preparação, Execução e Encerramento.

Para alcançar os objetivos dessas fases, foram executadas atividades de processos gerenciais, descritos em [PMI \(2013\)](#), e de processos orientados a produto, isto é, processos que especificam e criam produtos do projeto ([PMI, 2013](#)). Os processos gerenciais executados nesta pesquisa tiveram as seguintes responsabilidades: desenvolvimento de plano de gerenciamento do projeto, coleta de requisitos, definição de escopo, definição de atividades, sequenciamento de atividades, estimativa de duração das atividades, desenvolvimento de cronograma, controle de cronograma, e encerramento de projeto ou fase. Finalmente, os processos orientados a produto executados nesta pesquisa tiveram as seguintes responsabilidades: revisão de literatura, desenvolvimento de elementos de arcabouço, e desenvolvimento de exemplo de uso de elementos de arcabouço.

No processo responsável pela revisão de literatura, seguindo fases sugeridas em [Gil \(2010\)](#) para realização de pesquisa bibliográfica, inicialmente foi elaborado plano provisório com o objetivo de definir a estrutura lógica da parte da tese resultante de pesquisa bibliográfica. Em seguida, foram localizadas e lidas fontes de informação relevantes ao tema de pesquisa. Para localizar essas fontes de informação, inicialmente foram acessados resultados do processo de mapeamento sistemático de literatura realizado. Em seguida, foram acessados e analisados os conteúdos dessas fontes. Finalmente, foi redigido texto com os resultados desse processo de revisão de literatura. Considerando o problema de pesquisa e o objetivo de pesquisa, a pesquisa bibliográfica enfocou conceitos de base, arquitetura da informação, vocabulários controlados, metadados, elementos de modelagem, elementos de qualidade e elementos de Engenharia de Software.

Depois de concluída a pesquisa bibliográfica, foram analisados resultados da mesma

e foram propostos elementos de arcabouço para ciclo de vida de projeto de vocabulário controlado. Esses elementos são descritos no capítulo 10. Entre esses elementos, são encontrados modelos. Esses modelos foram desenvolvidos por meio de processo em que foram usados elementos da metodologia *METHONTOLOGY*. Essa metodologia é resumidamente descrita na seção 5.7.3.2. Na construção dos modelos, foi usada a linguagem de modelagem UML, escolhida em decorrência da existência de ferramentas, popularidade, maturidade e padronização (CRANEFIELD; PURVIS, 1999; OBJECT MANAGEMENT GROUP, 2017). Considerando as abordagens descritas em Moody (2005) para desenvolvimento de modelo de qualidade, o modelo de qualidade proposto no capítulo 10 foi desenvolvido por meio de abordagem embasada em teorias e normas. Nas definições dos elementos integrantes dos modelos, foi seguida recomendação em Moody (2005) e se procurou observar recomendações em Belnap (1993) e Pepper e Driscoll (2015). Cada elemento foi definido por meio de uma sentença concisa.

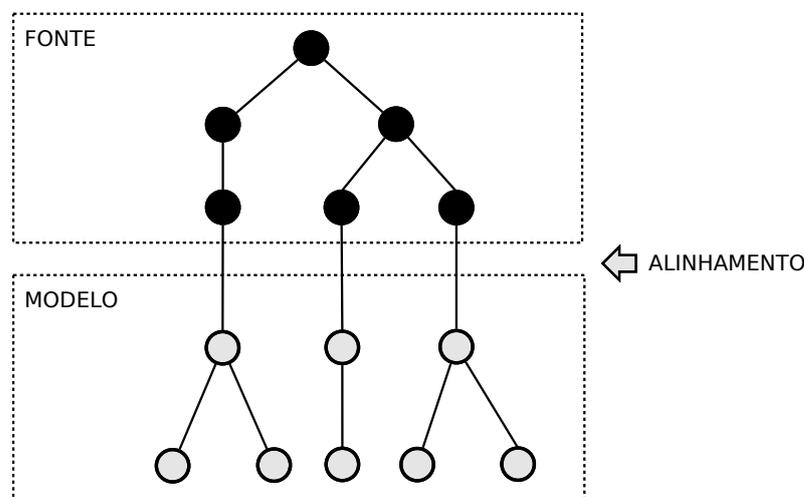
Considerando vantagens de compatibilizar os modelos a fontes de informação correlatas, quando adequado, foram reusados ou adaptados elementos originados em fontes de informação identificadas na pesquisa bibliográfica. No processo de desenvolvimento dos modelos (modelo de domínio e modelo de qualidade), foram executadas atividades com o objetivo de alinhá-los a conteúdos de fontes de informação identificadas na pesquisa bibliográfica, em particular, a ferramentas semânticas como a base de dados léxica *WordNet* (vide Princeton University (2010)). Nessa base de dados, substantivos, verbos, adjetivos e advérbios são agrupados em conjuntos de sinônimos que expressam conceitos (*synset*). Esses conjuntos de sinônimos são interligados por relações semânticas e léxicas. O uso da *WordNet* é justificado pela sua abrangência, compreensibilidade e facilidade de acesso. A figura 2 ilustra o processo de alinhamento entre conteúdos.

Finalmente, o uso de elementos do arcabouço proposto foi exemplificado por meio de desenvolvimento e avaliação da qualidade de um protótipo de vocabulário controlado no domínio da Engenharia de Software. Resultados desse processo são apresentados no capítulo 11. Entre os processos executados no desenvolvimento do protótipo de vocabulário controlado, com o objetivo de identificar vocabulários controlados no domínio da Engenharia de Software, foi executado mapeamento sistemático de literatura por meio de processo similar ao que é descrito em Petersen et al. (2008). Resultados desse processo são apresentados no apêndice A.

2.4 Ferramentas usadas

Ao longo da pesquisa, foram usadas diversas ferramentas. Principalmente, as seguintes: *OpenOffice Writer*, *OpenOffice Calc*, *GanttProject*, *TeXworks*, *Dia*, *CmapTools*, *JabRef*, *Astah*, *Tables Generator*, *AntConc*, *VocBench* e *SKOS Play*. *OpenOffice Writer*

Figura 2 – Elementos em processo de alinhamento de conteúdos



Fonte: Elaborado pelo autor

é um editor de texto ([THE APACHE SOFTWARE FOUNDATION, 2017](#)). *OpenOffice Calc* é um editor de planilhas ([THE APACHE SOFTWARE FOUNDATION, 2017](#)). *GanttProject* é uma ferramenta para gerir projetos que possibilita construir diagramas de Gantt para representação de cronogramas ([GANTTPROJECT TEAM, 2015](#)). *TeXworks* é uma ferramenta para desenvolvimento de documentos no formato *TeX* ([TEXWORKS, 2015](#)). *Dia* é uma ferramenta para desenho e edição de diagramas estruturados ([DIA, 2015](#)). *CmapTools* é uma ferramenta para criar, navegar, compartilhar e criticar modelos representados por mapas de conceitos, que são diagramas para representar conceitos e relações entre conceitos ([IHMC, 2014](#)). *JabRef* é uma ferramenta para gerenciamento de referências bibliográficas. Essa ferramenta adota o formato de arquivos *BibTeX*, formato esse usado na descrição de referências no sistema de preparação de documentos *LaTeX* ([JABREF, 2015](#)). *Astah* é uma ferramenta para desenvolvimento de modelos representados por diagramas construídos usando-se a linguagem de modelagem UML ([VISION, 2016](#)). *Tables Generator* é uma ferramenta para geração de tabelas em diversos formatos. Essa ferramenta foi usada na geração de tabelas *LaTeX* ([TABLESGENERATOR.COM, 2017](#)). *AntConc* é uma ferramenta para análise de concordância e texto ([ANTHONY, 2014](#)). *VocBench* é uma ferramenta para desenvolvimento de vocabulários controlados ([FAO, 2015b](#); [FAO, 2015a](#)). Por fim, *SKOS Play* é uma ferramenta para a publicação de vocabulário controlado expresso como *Simple Knowledge Organization System* (SKOS) ([FRANCART, 2017](#)). Dentre os critérios usados na seleção dessas ferramentas, destacam-se os seguintes: existência de versão gratuita da ferramenta, facilidade de acesso à ferramenta, presença de funcionalidades requeridas e facilidade de uso.

2.5 Fontes de informação

No processo de revisão de literatura executado pelo autor desta tese, foram acessadas bases de dados, sítios e bibliotecas. Por exemplo: ABNT Coleção; *ACM Digital Library*; *ProQuest*; *Web of Science*; Biblioteca Central da Universidade de Brasília (BCE - UnB); *Ebrary*; *IEEE Xplore Digital Library*; Portal da Biblioteca do Congresso dos EUA (*Library of Congress*); Portal de Periódicos CAPES; portais de editoras como *Prentice Hall*, *Addison-Wesley* e *John Wiley & Sons*; portais de livrarias como *Amazon*, *Barnes & Noble*, *Livraria Cultura* e *Livraria Siciliano*; *The Information Architecture Institute*; Repositório Institucional da Universidade de Brasília; *ScienceDirect*; *Scientific Electronic Library Online*; *Scopus Database*; e *Wiley Online Library*.

Parte II

Revisão de literatura

Prólogo

Esta parte da tese contém resultados de pesquisa bibliográfica realizada, organizados em capítulos dedicados aos seguintes assuntos: conceitos de base, arquitetura da informação, vocabulário controlado, metadado, elementos de modelagem, elementos de qualidade e Engenharia de Software. O capítulo 3, sobre conceitos de base, tem como objetivo prover fundamentação conceitual necessária ao entendimento dos capítulos seguintes e promover a normalização de termos usados nesta tese. Esse capítulo aborda conceitos designados pelos seguintes termos: informação, recurso de informação, coleção de recursos de informação, espaço de informação, recuperação da informação, organização da informação, representação da informação, análise de assunto, sumarização da informação, classificação da informação e indexação da informação. O capítulo 4, dedicado ao assunto arquitetura da informação, contém não só definições para arquitetura e para arquitetura da informação, mas também informação sobre a prática da arquitetura da informação, sobre a disciplina Arquitetura da Informação e sobre a Teoria Geral da Arquitetura da Informação. O capítulo 5, dedicado ao assunto vocabulário controlado, contém definições do termo “vocabulário controlado” e informação sobre propósitos e usos de vocabulários controlados, aspectos do controle de vocabulário na recuperação da informação, elementos de vocabulários controlados, relações em vocabulários controlados, estruturas de vocabulários controlados, desenvolvimento de vocabulários controlados, tecnologias para representação de vocabulários controlados, normas e padrões sobre vocabulários controlados. O capítulo 6, dedicado ao assunto metadado, contém definições do termo metadado e informação sobre propósitos, usos, armazenamento, atributos, qualidade, classes, esquemas, perfis de aplicação, gestão, desenvolvimento e normas de metadados. O capítulo 7, dedicado ao assunto modelagem, contém definições do termo modelo e informação sobre propósitos de modelos, usos de modelos, processo de modelagem, princípios de modelagem e análise de domínio. O capítulo 8, dedicado ao assunto qualidade, contém definições do termo qualidade e informação sobre qualidade da informação, qualidade de modelo conceitual, qualidade de vocabulário controlado, avaliação da qualidade de vocabulário controlado, características de qualidade, medidas e modelo de qualidade. O capítulo 9, sobre Engenharia de Software, aborda o domínio sobre o qual foi desenvolvido o protótipo de vocabulário controlado. Esse capítulo contém informação sobre necessidades de informação de engenheiros de software, ciclo de vida de software, recursos de informação, processos, desenvolvimento de processos e melhoria de processos em ciclo de vida de software, modelo de maturidade, processo de desenvolvimento de software, repositório de software, recuperação da informação em repositório de software, vocabulários controlados e metadados propostos para o domínio da Engenharia de Software. Por fim, é importante destacar que esses capítulos não incluem

descrições detalhadas de vários resultados da pesquisa bibliográfica realizada, mas que esses capítulos incluem referências para fontes de informação que podem ser consultadas pelos leitores interessados.

3 Conceitos de base

Este capítulo descreve conceitos de base relevantes no contexto desta tese, por exemplo, conceitos designados pelos termos informação, recurso de informação, coleção de recursos de informação, espaço de informação, recuperação da informação, sistema de recuperação da informação, avaliação de sistema de recuperação da informação, organização da informação, sistema de organização da informação, definição do termo “conceito”, representação da informação, análise de assunto, sumarização da informação, classificação, indexação e gestão da informação. Nesta tese, este capítulo tem, como principais objetivos, prover embasamento e promover uniformização de termos.

3.1 Informação

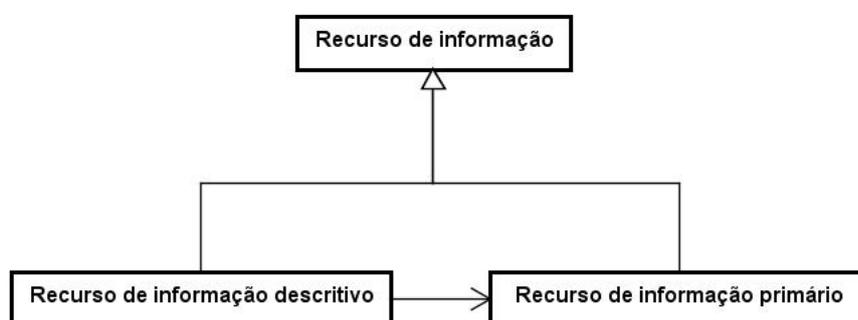
O termo informação designa conceito não singular estudado por várias disciplinas (MATHEUS, 2005; NUNBERG, 1996; PINHEIRO; LOUREIRO, 1995; ROWLEY, 1998). Existem inúmeras definições para esse conceito, que variam entre áreas de conhecimento e contextos em que são usadas (BELKIN, 1978; CAPURRO; HJØRLAND, 2003; BRAMAN, 1989; BUCKLAND, 1991; BATES, 2005). Diversas disciplinas científicas usam o conceito de informação, relacionando-o a fenômenos específicos (CAPURRO; HJØRLAND, 2003). Esta tese enfoca a informação registrada, a informação como registro. Em particular, no contexto tecnológico. O termo “informação tecnológica” é usado em diversas fontes de informação. Aguiar (1991) sugere que informação tecnológica é aquela relacionada com o modo de fazer um produto ou prestar um serviço para colocá-lo no mercado. Por sua vez, Jannuzzi e Montalli (1999) conceituam informação tecnológica como aquela que trata da informação necessária, utilizada e gerada em procedimentos de aquisição, inovação e transferência de tecnologia; em procedimentos de metrologia, de certificação da qualidade e de normalização; em processos de produção. A informação tecnológica pode ter diferentes finalidades, tais como, prover insumo para pesquisas; assegurar direito de propriedade industrial; difundir tecnologias; subsidiar gestão tecnológica; possibilitar acompanhamento e avaliação de tendência tecnológica; e prover suporte à avaliação de impactos econômico, social e ambiental de tecnologias (AGUIAR, 1991). Quanto à informação gerada em engenharia, segundo Goh et al. (2009), ela é frequentemente não estruturada e difícil de estruturar em seu todo. Giess, Wild e McMahon (2008) destacam que parte da documentação em engenharia é informal, e que essa parte da documentação frequentemente provê suporte à documentação mais formal. Também destacam que interpretar essa documentação requer conhecimento tácito e entendimento do contexto. A comunicação da informação tecnológica envolve diversas fontes de informação e canais de comunicação (IBGE, 2010).

No processo de inovação tecnológica, as organizações podem usar diversas fontes de informação tecnológica, por exemplo, patentes, normas, legislações, manuais, catálogos de fabricantes, livros, artigos técnicos, relatórios técnicos, exposições, feiras, seminários, máquinas, equipamentos, produtos, congressos e pessoas (ALLEN, 1984; IBGE, 2010; MONTALLI; CAMPELLO, 1997). As fontes usadas dependem dos setores da economia em que as organizações atuam (IBGE, 2010). Nesta tese, normas técnicas são fontes de informação importantes. Segundo a Associação Brasileira de Normas Técnicas (ABNT), o termo norma é definido como “documento estabelecido por consenso e aprovado por um organismo reconhecido, que fornece regras, diretrizes ou características mínimas para atividades ou para seus resultados, visando à obtenção de um grau ótimo de ordenação em um dado contexto” (ABNT, 2014).

3.2 Recurso de informação

Nesta tese, como em Glushko (2013), o termo “recurso de informação” (*information resource*) designa “unidade descritível de informação registrada”, enquanto artefato designa recurso de informação criado por uma ou mais pessoas. Artefato é uma classe de recurso de informação. O conteúdo de um recurso de informação é a informação no recurso. Podem existir relações entre recursos de informação. Essas relações podem refletir diferentes perspectivas. Um recurso de informação pode ser primário ou descritivo. Um recurso de informação descritivo é um recurso sobre recurso de informação primário (TAYLOR; JOUDREY, 2009; GLUSHKO, 2013; NISO, 2005). A figura 3 ilustra relações entre essas classes de recursos de informação.

Figura 3 – Classes de recursos de informação



Fonte: Elaborado pelo autor

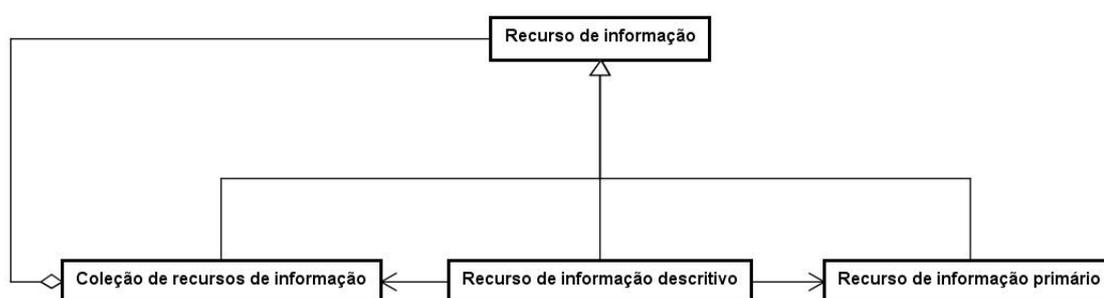
Existem diversos termos que, em certos contextos, são sinônimos de recurso de informação, tais como objeto de conteúdo, item de conteúdo e documento (GLUSHKO, 2013; MILLER, 2011; NISO, 2005). NISO (2005) define objeto de conteúdo (*content*

object) como entidade que contém dado/informação, como qualquer item a ser descrito para inclusão em sistema de recuperação da informação, sítio *Web* ou outra fonte de informação. Em Kowalski e Maybury (2002), item é a menor unidade que pode ser processada e manipulada por sistema de recuperação da informação. Segundo essa fonte, itens podem ter diferentes níveis de granularidade, e um item pode ser composto por diferentes modalidades de informação. A fonte ISO (2017) define documento como unidade unicamente identificada de informação para uso humano, enquanto a fonte NISO (2005) define documento como item, impresso ou não, passível de catalogação e indexação. Por fim, em Buckland (1997), também são propostas definições do conceito designado pelo termo documento.

3.3 Coleção de recursos de informação

Nesta tese, coleção é definida como agregado de recursos de informação organizado de acordo com algum critério. Uma coleção pode conter identificadores de recursos, em vez de conter os próprios recursos. Isso possibilita que um mesmo recurso integre mais de uma coleção. Uma coleção é também um recurso e a ela pode ser associado recurso descritivo (DLESE, 2005; GLUSHKO, 2013; NISO, 2007). Miller (2011) define coleção digital como coleção de recursos digitais, com metadados sobre esses recursos, disponibilizada em linha (*online*) por meio de interface que possibilite aos usuários buscarem (*search*) e navegarem (*browse*) o conteúdo da coleção. A figura 4 ilustra relações entre conceitos anteriormente referenciados.

Figura 4 – Coleção de recursos de informação



Fonte: Elaborado pelo autor

Em coleções, os recursos devem ser selecionados e organizados para promover descoberta, acesso e uso. A seguir, são relacionadas atividades geralmente presentes na construção de uma coleção: identificar necessidades de informação de usuários, analisar estatísticas de uso, formular critérios para seleção de recursos, elaborar planos para compartilhar recursos, desenvolver plano de catalogação, coletar informação sobre recursos, prover

manutenção de recursos e expor coleção aos usuários (DLESE, 2005). Uma coleção pode ser armazenada em repositório, deve ser desenvolvida segundo políticas explícitas e descrita de modo que os seus usuários possam entender as suas características. No desenvolvimento de uma coleção, devem ser observados: disponibilidade, acessibilidade, respeito a direitos de propriedade, registro de medidas de uso e de utilidade, interoperabilidade, integração ao fluxo de trabalho dos usuários e sustentabilidade. Também existem aspectos relevantes sobre os recursos das coleções, tais como os seguintes: formato que suporte uso atual e futuro; ausência de barreiras desnecessárias ao acesso; recurso significativo, coerente e usável fora do contexto em que foi criado; recurso nomeado por identificador persistente, globalmente único e conversível no endereço do recurso; recurso gerenciado; possibilidade de determinar se o recurso está de acordo com a sua origem, a sua estrutura e a sua história; possibilidade de determinar se o recurso foi modificado sem autorização ou corrompido; possibilidade do recurso ser indexado (BAEZA-YATES; RIBEIRO-NETO, 2011; MILLER, 2011; NISO, 2007).

3.3.1 Espaço de informação

O conceito designado pelo termo “espaço de informação” é um conceito relacionado àquele designado pelo termo “coleção de recursos de informação”. Segundo Gödert, Hubrich e Nagelschmidt (2014), espaço de informação é coleção delimitada de recursos de informação. O termo “espaço de informação” pode designar, desde coleção de recursos de informação homogêneos armazenados em um repositório, até coleção de recursos de informação heterogêneos, distribuídos em diversos repositórios e organizados independentemente. Em Siqueira (2008), espaço de informação designa “delimitação de uma coleção de registros”, onde registro é coisa para um sujeito. Por fim, em Siqueira (2012), o termo “espaço de informação” é definido como “conjunto de informações distintas em um espaço distinto”.

3.4 Recuperação da informação

A descoberta de recursos de informação (*resource discovery*) é um processo para identificar recursos de informação relevantes aos usuários. Nesse contexto, são pertinentes os conceitos designados pelos termos “recuperação da informação” e “organização da informação”, que podem designar processos, resultados de processos ou campos de pesquisa (HJØRLAND, 2015; POMERANTZ, 2015). Nesse momento, é importante distinguir entre mineração de dados (*data mining*), mineração de texto (*text mining*) e recuperação da informação. O processo de mineração de dados visa identificar, por exemplo, padrões em conjuntos de dados, enquanto o processo de recuperação da informação visa recuperar recursos de informação relevantes em coleções. Segundo Goh et al. (2009), o processo de mineração de dados é usado para extrair padrões de dados estruturados e pode ser aplicado na construção de modelos para definição de valores desconhecidos de atributos a

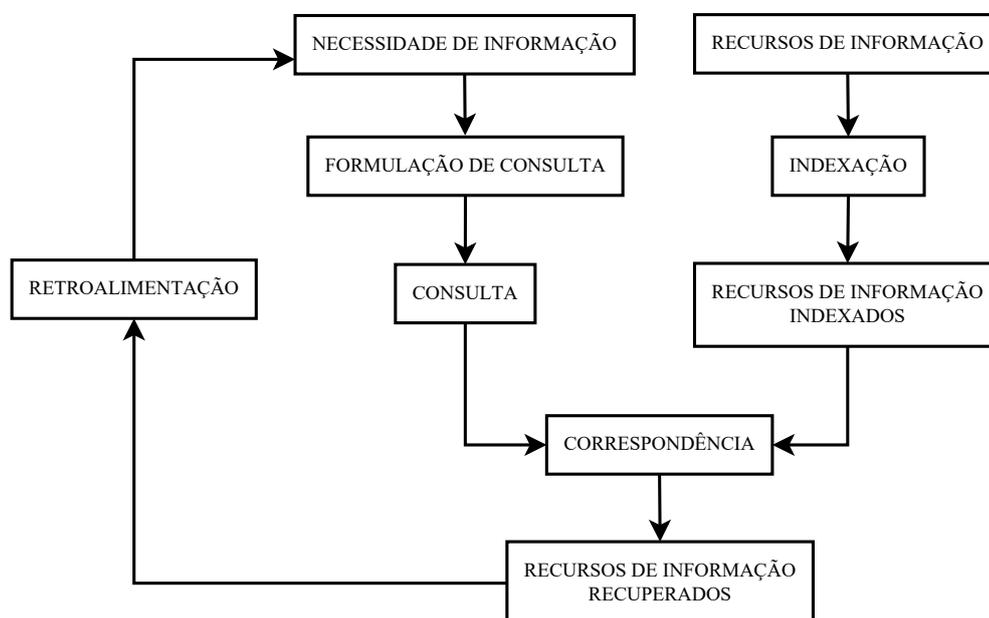
partir de valores conhecidos de outros atributos e na construção de modelos para descrever domínios de interesse. O termo “mineração de texto”, segundo [Goh et al. \(2009\)](#), designa processo de procura por padrões em texto em linguagem natural e pode ser definido como processo de análise de texto para extrair informação com propósito particular. O processo de mineração de texto, útil na identificação de frases chave (*key phrase*), enfoca a extração de informação de texto por meio de algoritmos e a descoberta de relações a partir dessa informação. Por fim, segundo [Goh et al. \(2009\)](#), esse processo pode englobar atividades para extrair características de recursos de informação, classificar e agrupar recursos de informação.

O termo “recuperação da informação” frequentemente designa processo por meio do qual é recuperada informação que melhor corresponda à solicitação do usuário. Esse processo frequentemente engloba atividades para obter informação sobre determinado assunto ([IEEE, 2016](#); [SAA, 2014](#)). A figura 5 ilustra a recuperação da informação segundo [Hiemstra \(2009\)](#). Na recuperação da informação por máquina, definições do termo “recuperação da informação” não são recentes. [Mooers \(1951\)](#) define recuperação da informação como processo ou método pelo qual um usuário de informação é capaz de converter a sua necessidade de informação em lista de citações para documentos armazenados com informação de utilidade para ele. Por fim, vale destacar que a recuperação da informação pode ser dificultada por diversos motivos, por exemplo, por características da linguagem natural, por experiências prévias e motivações dos usuários do sistema de recuperação ([BAEZA-YATES; RIBEIRO-NETO, 2011](#); [BRASETHVIK, 2004](#); [JACOB, 2004](#); [ROWLEY; HARTLEY, 2008](#)).

3.4.1 Modos de recuperação da informação

A recuperação da informação pode ocorrer por busca, navegação ou combinação dessas abordagens. Por exemplo, [Marchionini \(2006\)](#) descreve processo denominado “busca exploratória” (*exploratory search*), que combina abordagens. A combinação de abordagens pode resultar em diferentes níveis de sucesso. Na recuperação da informação por busca, são executadas atividades para representação de consulta, execução de consulta e avaliação de resultados. A informação é recuperada a partir de termos na consulta do usuário. O usuário submete a consulta de acordo com sua necessidade de informação; a consulta é analisada e possivelmente expandida; a consulta é processada; recursos são recuperados, posicionados (*ranked*) e retornados ao usuário ([BAEZA-YATES; RIBEIRO-NETO, 2011](#); [CHU, 2010](#); [HAGEDORN, 2000](#)). A recuperação da informação por busca tem o objetivo de recuperar informação que satisfaça à consulta, sendo adequada quando o usuário sabe pelo que procura e necessita de informação específica. Uma desvantagem da recuperação da informação por busca é depender da capacidade de formulação de consultas por parte do usuário. Buscas de diferentes tipos podem ser realizadas no processo de recuperação da

Figura 5 – Processo de recuperação da informação



Fonte: Adaptado de [Hiemstra \(2009\)](#)

informação, por exemplo, buscas por item específico, por assunto ou por palavra-chave. Existem também diferentes estratégias de busca. Por exemplo, o usuário pode realizar consulta e depois modificá-la incorporando termos a resultado. A busca por informação pode apresentar características particulares em diferentes organizações (empresas, instituições etc.), pois em diferentes organizações podem, por exemplo, existir diferentes classes de recursos de informação (estruturados, não estruturados etc.), diferentes produtores de recursos de informação, diferentes esquemas de metadados, usuários com diferentes direitos de acesso e diferentes repositórios ([CHU, 2010](#); [GOH et al., 2009](#); [HAWKING, 2011](#); [ROWLEY; HARTLEY, 2008](#); [WEI et al., 2013](#)). A recuperação da informação por busca pode ser embasada em correspondência exata (*exact match*) ou parcial (*partial match*). Na correspondência exata, a relevância de um recurso de informação depende de termos, no recurso de informação, corresponderem a termos na consulta. Finalmente, existem diversas alternativas para indicar a relevância de recurso de informação, tais como por meio de operadores booleanos para expressar operações que ampliem ou restrinjam consultas, cálculos para avaliar a similaridade entre consultas e respostas, ou modelos para estimar a probabilidade da resposta ser relevante ([BAEZA-YATES; RIBEIRO-NETO, 2011](#); [CHU, 2010](#); [GÖDERT; HUBRICH; NAGELSCHMIDT, 2014](#)).

Existem situações em que a recuperação da informação por busca de palavra-chave não é a melhor abordagem. Nessas situações, uma alternativa é a recuperação da informação

por navegação. Nessa alternativa, não é necessário usar termos específicos para recuperar informação; o usuário explora o espaço de informação e frequentemente navega por recursos de informação organizados de acordo com estruturas definidas para facilitar a identificação de relações entre conceitos. A recuperação da informação por navegação é importante, por exemplo, quando o usuário tem dificuldade de explicitar a informação desejada; quando necessita de uma visão geral da informação recuperável; quando tem necessidade de informação abrangente e pouco específica; quando não sabe construir consulta; quando deseja explorar coleção; quando não está familiarizado com termos no domínio (CHU, 2010; GOH et al., 2009; HAGEDORN, 2000; MAAREK, 2011). A recuperação da informação por navegação pode ser particularmente relevante em determinados domínios ou contextos. Por exemplo, em projetos de engenharia, segundo Giess, Wild e McMahon (2008), a recuperação de só um recurso de informação por meio de busca por palavra-chave pode dificultar a avaliação da proveniência da informação e sua assimilação, em decorrência da falta de contexto. Nesses projetos, Giess, Wild e McMahon (2008) sugerem que a organização prévia dos recursos de informação provê um meio para navegação e indicação de relações no domínio, provê indicação sobre contexto.

3.4.2 Recuperação semântica da informação

A recuperação da informação por meio da comparação entre termos, sem considerar significados, apresenta potenciais dificuldades como: resultados incorretos devido a informação insuficiente na recuperação da informação; incapacidade de recuperar informação com semântica similar, porém com termos distintos dos informados na consulta; resultados sensíveis a vocabulário; termo que designa o conceito no recurso não necessariamente é o termo informado na consulta; recuperação da informação irrelevante devido ao termo usado na consulta ter diversos significados; relações semânticas são desconhecidas; incapacidade de eliminar ambiguidades por não considerar contexto do usuário; grande quantidade de resultados apresentados ao usuário; pode requerer elevado esforço do usuário para a formulação de consulta que resulte em recuperação com precisão satisfatória; pode haver pouco compartilhamento de terminologia entre os usuários, dificultando a formulação de consultas (ANTONIOU et al., 2012; CHEN; CHU; CHEN, 2010; DAVIES, 2010; GIESS; WILD; MCMAHON, 2008; GÖDERT; HUBRICH; NAGELSCHMIDT, 2014; MUSTAFA; KHAN; LATIF, 2008).

Uma alternativa ao processo de recuperação da informação anteriormente descrito, é considerar semântica na recuperação da informação. O termo “semântica” designa o estudo do significado. Por exemplo, o estudo dos significados de palavras, expressões, sinais, sentenças e textos (HJØRLAND, 2008a). Na Linguística, que é o estudo científico da linguagem, semântica é nível de análise linguística que lida com o significado, nível de análise linguística em que significado é analisado (SYAL, 2010). Existem várias definições

para o termo significado. [Ogden e Richards \(1923\)](#) apresentam uma lista de definições: uma propriedade intrínseca, as outras palavras associadas a uma palavra em um dicionário, a conotação de uma palavra, uma essência, aquilo ao qual o usuário de um símbolo acredita ele próprio estar referenciando, aquilo ao qual o intérprete de um símbolo referencia. Sobre o significado de uma palavra, [Syal \(2010\)](#) sugere as seguintes distinções: significado lógico ou denotativo, consiste do significado literal de uma palavra; significado conotativo, consiste de atributos associados ao conceito no mundo real; significado social, decorre da circunstância na qual a palavra é usada; significado temático, depende da forma como a mensagem é organizada em termos de ordem, foco e ênfase.

Em um processo de recuperação semântica da informação (*semantic information retrieval*), são consideradas relações semânticas para melhorar medidas como precisão e revocação. [Gödert, Hubrich e Nagelschmidt \(2014\)](#) usam o termo “recuperação conceitual” (*conceptual retrieval*) para designar processo em que a recuperação da informação considera conhecimento sobre conceitos. Relacionam expansão de consulta (*query expansion*) e modificação de consulta (*query modification*) como abordagens nesse processo. Na expansão de consulta, a consulta original é suplementada com termos, enquanto que na modificação de consulta, a consulta original é modificada por termos. Por fim, [Stock \(2010\)](#) usa o termo “recuperação da informação embasada em conceitos” (*concept-based information retrieval*) para designar processo de recuperação da informação no nível de conceitos, que vai além do nível de palavras.

Na recuperação semântica da informação, podem ser usadas ferramentas semânticas. Uma ferramenta semântica é uma ferramenta que provê informação semântica. Nesse contexto, o termo “informação semântica” designa informação sobre os significados de símbolos e de relações entre conceitos ([HJØRLAND, 2007c](#)), enquanto o termo ferramenta pode ser definido como algo tangível usado na realização de atividade, voltado a produzir produto ou resultado ([PMI, 2013](#)). Na recuperação semântica da informação, o entendimento de conceitos pode ser realizado, por exemplo, por meio de vocabulário controlado. Nesse contexto, vocabulário controlado pode contribuir na descrição de recursos de informação, geração de metadados, seleção de termos em consultas, expansão de consultas e classificação de recursos. Por fim, vale destacar que a recuperação semântica da informação também apresenta potenciais dificuldades, por exemplo, quando da mudança de significado de termo e quando da identificação de termos e conceitos relevantes a domínios específicos ([MUSTAFA; KHAN; LATIF, 2008](#); [REIMER, 2011a](#); [REN; BRACEWELL, 2009](#); [SOERGEL et al., 2004](#)).

No contexto da recuperação semântica da informação, pode ocorrer busca semântica (*semantic search*), definida por [Giunchiglia, Dutta e Maltese \(2013\)](#) como busca em que termos são expandidos e ambiguidades removidas por meio de sinônimos e de termos mais específicos relacionados em vocabulário controlado. Segundo [Giunchiglia e](#)

Zaihrayeu (2007), a busca semântica consiste em encontrar categorias e/ou recursos em categorias, de modo que os recursos encontrados sejam semanticamente correspondentes aos informados na consulta. Na correspondência semântica, o significado associado ao recurso de informação é mais específico ou equivalente ao atribuído à consulta no senso comum. A resposta a uma consulta é então o conjunto de recursos de informação cujos conceitos são mais específicos ou equivalentes ao conceito designado na consulta. Wei, Barnaghi e Bargiela (2008) e Dong, Hussain e Chang (2008) descrevem e classificam diversos sistemas e tecnologias relacionados à busca semântica da informação, enquanto, em Mangold (2007), são propostos os seguintes critérios para classificar abordagens de busca semântica: arquitetura, acoplamento, transparência, contexto de usuário, modificação de consulta, estrutura de ontologia e tecnologia de ontologia. Finalmente, em Strasunskas e Tomassen (2010), o termo “sistema de busca semântica” (*semantic search system*) designa sistema de recuperação da informação que usa tecnologias semânticas para melhorar a recuperação da informação.

3.4.3 Sistema de recuperação da informação

Um sistema é uma coleção de componentes organizada para função ou conjunto de funções, é composto por componentes inter-relacionados organizados para propósitos definidos (ISO, 2010; IEEE, 1990; ISO, 2008a; IEEE, 2016). Sistemas de informação são sistemas desenvolvidos para identificar e recuperar recursos de informação, representar atributos de recursos, organizar recursos e organizar representações de recursos. Os processos geralmente presentes em sistemas de informação são desenvolvimento de coleção, representação, organização e recuperação da informação (JACOB, 2004). Gilliland (2008) relaciona as seguintes atividades no desenvolvimento de sistemas de informação: identificar esquemas de metadados e aplicá-los de modo a atender necessidades de quem cria e necessidades de quem usa informação, definir granularidade e outros aspectos dos metadados necessários para se alcançar os objetivos definidos, garantir que os vocabulários controlados sejam atuais e garantir que as terminologias sejam as apropriadas.

Um sistema de armazenamento e recuperação da informação (*information storage and retrieval system*) engloba operações, equipamentos e software por meio dos quais recursos de informação são indexados e armazenados de tal modo que possam ser recuperados em resposta a solicitações realizadas por meio de comandos aceitos pelo sistema (NISO, 2005). Os primeiros sistemas de recuperação da informação surgiram da necessidade de organizar repositórios centrais tais como bibliotecas. Na medida em que computadores se tornaram comercialmente disponíveis, eles passaram a integrar sistemas de recuperação da informação (KOWALSKI; MAYBURY, 2002). Existem diversas classes de sistemas de recuperação da informação, por exemplo, sistema de recuperação da informação em linha (*online*) e sistema de recuperação da informação na Internet. Nos sistemas de recuperação

da informação em linha (*online*), frequentemente a informação é representada por profissionais e são usados vocabulários controlados. Por sua vez, os sistemas de recuperação da informação na Internet caracterizam-se pelo uso de algoritmos para automatizar a indexação e ordenar resultados. Esses sistemas são usados em ambiente com as seguintes características: pouco controle de qualidade na produção de informação; presença de recursos de informação em diversos idiomas, tamanhos e formatos; presença de recursos de informação não estruturados ou pouco estruturados; e raro uso de vocabulários controlados (CHU, 2010; LEWANDOWSKI, 2005; ROWLEY; HARTLEY, 2008). Como características da recuperação da informação na *World Wide Web*, Baeza-Yates e Ribeiro-Neto (2011) relacionam coleção com grande quantidade de recursos, recursos distribuídos entre sítios, recursos conectados por hiperligações (*hyperlink*) e grande quantidade de consultas. Para Maarek (2011), certas características frequentemente apresentadas por recursos de informação na *World Wide Web* resultam em desafios ao processo de recuperação da informação. Por exemplo, as seguintes: distribuição, volatilidade, quantidade, falta de estruturação, redundância, qualidade e heterogeneidade. No contexto da *World Wide Web*, Maarek (2011) também menciona dificuldades dos usuários para expressar consultas e interpretar resultados.

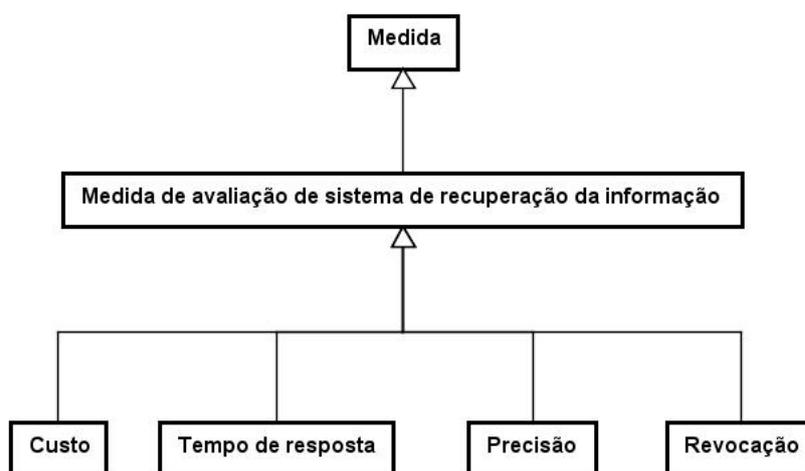
3.4.3.1 Avaliação de sistema de recuperação da informação

Um importante objetivo de um sistema de recuperação da informação é recuperar todos os recursos relevantes e o mínimo de recursos irrelevantes. Para recuperar informação, os usuários dessa classe de sistema geralmente traduzem as suas necessidades de informação em consultas formuladas por meio de linguagens providas por esses sistemas (BAEZA-YATES; RIBEIRO-NETO, 2011). Segundo Kowalski e Maybury (2002), uma medida de sucesso de um sistema de recuperação da informação é quanto ele consegue minimizar o tempo gasto pelo usuário para encontrar a informação necessária, excluído o tempo para leitura de itens relevantes. Um sistema de recuperação da informação deve minimizar o tempo gasto na composição da busca, na execução da busca e na leitura de itens irrelevantes ao usuário. Kowalski e Maybury (2002) destacam que sistemas de recuperação da informação têm papel no suporte aos usuários na geração de consultas e na apresentação de resultados em formatos que facilitem a determinação da informação relevante. Para esses autores, são obstáculos: ambiguidades em linguagens, limitações do usuário, e diferenças entre o vocabulário do usuário e o usado por autor do recurso de informação. Entre as limitações dos usuários de sistemas de recuperação da informação, Kowalski e Maybury (2002) citam a falta de conhecimento do vocabulário no domínio dos recursos consultados e a falta de foco na informação necessária.

Em sistema de recuperação da informação, a informação pode ou não ser recuperada em resposta à consulta do usuário (KOWALSKI; MAYBURY, 2002). Um sistema de recuperação da informação pode ser avaliado segundo diversos critérios. Nesse contexto,

avaliar é determinar sistematicamente o grau com o qual é atendido certo critério (IEEE, 2016). A seguir, são listadas medidas de avaliação de sistema de recuperação da informação: custo, tempo de resposta (tempo entre a submissão da consulta e o retorno dos resultados), precisão (*precision*) e revocação (*recall*). A figura 6 apresenta relações entre alguns dos conceitos anteriormente citados.

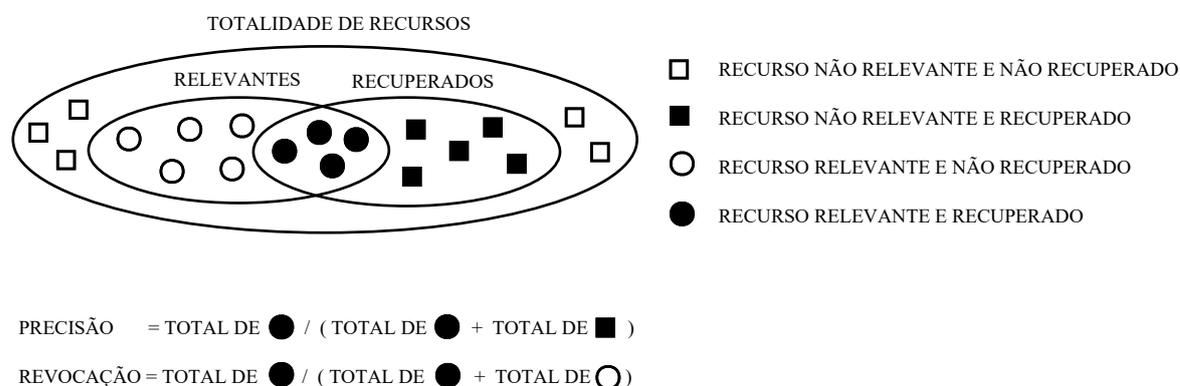
Figura 6 – Medidas de avaliação de sistema de recuperação da informação



Fonte: Elaborado pelo autor

A precisão mede a exatidão dos resultados, a capacidade de um sistema recuperar apenas recursos relevantes, de um sistema suprimir informação irrelevante. É calculada dividindo-se a quantidade de recursos relevantes recuperados pela quantidade total de recursos recuperados. A precisão pode ser interpretada como a probabilidade de um recurso recuperado ser relevante (BINKLEY; LAWRIE, 2010; GLUSHKO, 2013; NISO, 2005; ROWLEY; HARTLEY, 2008). Alta precisão indica quantidade elevada de recursos de informação relevantes no resultado (GÖDERT; HUBRICH; NAGELSCHMIDT, 2014). A revocação mede o quão completo é um conjunto de resultados, quanto dos recursos relevantes em uma coleção foi recuperado. É uma medida da capacidade de um sistema recuperar informação relevante. É calculada dividindo-se a quantidade de recursos relevantes recuperados pelo total de recursos relevantes em uma coleção. Pode ser interpretada como a probabilidade de um artefato relevante ser recuperado (BINKLEY; LAWRIE, 2010; GLUSHKO, 2013; NISO, 2005; ROWLEY; HARTLEY, 2008). Segundo Aitchison, Bawden e Gilchrist (2000), em geral, é inversa a relação entre precisão e revocação, um ganho na revocação tende a ser acompanhado por uma perda na precisão e vice-versa. A figura 7 é uma representação do cálculo dessas medidas de avaliação de sistema de recuperação da informação.

Figura 7 – Elementos no cálculo de precisão e de revocação



Fonte: Elaborado pelo autor

Nesse contexto, [Chu \(2010\)](#) considera difícil definir relevância e determinar os recursos relevantes em um sistema. [Kowalski e Maybury \(2002\)](#) definem como relevante o recurso que contém informação que auxilia o usuário a responder sua questão e, como irrelevante, o recurso que não provê informação útil. Segundo [Baeza-Yates e Ribeiro-Neto \(2011\)](#), relevância é uma avaliação pessoal dependente de tarefa e contexto. Para [Gödert, Hubrich e Nagelschmidt \(2014\)](#), relevância é um conceito fundamental à recuperação da informação, que mede o acordo entre a necessidade de informação representada pela consulta formulada e a informação recuperada. [Strasunskas e Tomassen \(2010\)](#) consideram que o processo de avaliação de sistema de recuperação da informação pode ser centrado no sistema ou no usuário. Ainda segundo [Strasunskas e Tomassen \(2010\)](#), algumas avaliações indicam que sistemas de recuperação semântica da informação apresentam melhoras em relação a sistemas de recuperação da informação tradicionais. [Morato, Sanchez-Cuadrado e Dimou \(2013\)](#) sugerem, como critérios para avaliação de sistemas de recuperação semântica da informação, a capacidade do sistema eliminar ambiguidades e do sistema gerar resultados derivados de relações entre conceitos. Finalmente, em [Strasunskas e Tomassen \(2010\)](#) é proposto um arcabouço para avaliação de aplicações de busca semântica. Segundo esse arcabouço, qualidade de sistema, qualidade de ontologia e qualidade de consulta têm impacto na qualidade dessas aplicações.

3.5 Organização da informação

A recuperação da informação é relacionada à organização da informação. A organização da informação visa facilitar o acesso à informação por auxílio à navegação, localização, identificação e seleção de recursos; a recuperação da informação é um objetivo da organi-

zação da informação (BAEZA-YATES; RIBEIRO-NETO, 2011; ROWLEY; HARTLEY, 2008; SVENONIUS, 2000). Para Glushko (2013), quanto maior o esforço na organização da informação, maior tende a ser a eficácia na recuperação da informação. Segundo esse autor, organizar é criar capacidades por meio da imposição intencional de ordem e estrutura. Para Morville e Rosenfeld (2006), organizar envolve agrupar em classes significativas e distintas. Por fim, segundo Taylor e Joudrey (2009), são atividades integrantes de processo de organização da informação: identificar recursos, agrupar recursos em coleções, criar listas de recursos, prover acessos a recursos e prover meios para localização de recursos.

Brascher e Café (2008) distinguem entre organização do conhecimento e organização da informação. Segundo essa fonte, organização do conhecimento engloba representação do conhecimento, enquanto organização da informação engloba representação da informação. A representação do conhecimento compreende estrutura conceitual que representa modelos de mundo. Por sua vez, a representação da informação compreende conjunto de atributos que representa objeto informacional. Segundo Brascher e Café (2008), a organização da informação se aplica às ocorrências individuais de objetos informacionais, enquanto a organização do conhecimento se aplica a unidades de pensamento (conceitos). A organização da informação compreende a organização de objetos informacionais para arranjá-los sistematicamente em coleções, enquanto a organização do conhecimento visa construir modelos de mundo que são abstrações da realidade. Para Pontes e Lima (2012), a organização do conhecimento é uma condição necessária para a organização da informação, para a organização dos recursos de informação, a organização do conhecimento produz representações sistematizadas que podem ser usadas na organização da informação. Segundo Hjørland (2008b), em sentido estrito, a organização do conhecimento engloba atividades como descrever, indexar e classificar documentos. O termo “organização do conhecimento” pode designar campo de estudo preocupado com natureza e qualidade de processos de organização do conhecimento, como pode designar sistemas de organização do conhecimento (*knowledge organization system*) usados na organização e representação de documentos, trabalhos e conceitos. Ainda segundo Hjørland (2008b), sistemas e processos de organização de conhecimento têm o objetivo de preencher necessidades de informação dos usuários.

3.5.1 Sistema de organização da informação

Glushko (2013) define sistema de organização como “caracterização abstrata de como uma coleção de recursos é descrita e disposta com o objetivo de possibilitar que seres humanos e agentes computacionais interajam com os recursos”. Esse tipo de sistema tem o objetivo de possibilitar que recursos sejam localizados e acessados, estabelece relações entre classes, agrupa a informação similar e diferencia a não similar (JACOB, 2004; SVENONIUS, 2000). Nesse contexto, Glushko (2013) descreve os conceitos “arranjo

intencional” e “princípio de organização”. Um arranjo intencional resulta de atos de organização por pessoas ou processos computacionais; resulta de decisões de desenho (*design*) sobre o que, quando, porque, quanto, como e por quem organiza. Um princípio de organização consiste de diretivas de desenho (*design*) ou arranjo de coleções de recursos. Esses princípios usam propriedades dos recursos e idealmente são expressos de modo independente de como são implementados (GLUSHKO, 2013). Em Morville e Rosenfeld (2006) são encontrados os termos “esquema de organização” e “estrutura de organização”. Entre os esquemas, relacionam alfabético, cronológico, geográfico, tópico, tarefa, audiência e metáfora e, entre as estruturas, relacionam hierárquica, modelo de banco de dados e hipertextual.

Segundo Glushko (2013), as atividades realizadas pelos responsáveis por um sistema de organização são: selecionar recursos a incluir no sistema de organização; especificar princípios e regras a serem usados para organização dos recursos; desenhar (*design*); implementar ações, funções ou serviços que usam os recursos; e prover a manutenção dos recursos e da organização para suportar as interações. Glushko (2013) relaciona as seguintes atividades no ciclo de vida de um sistema de organização: definir domínio e escopo do sistema, identificar requisitos do sistema, desenhar (*design*) e implementar sistema, realizar operação e manutenção do sistema. Os princípios de desenho (*design*) para alcançar os propósitos de um sistema de organização dependem das classes de recurso, domínio e contexto. Svenonius (2000) destaca a importância de uma ideologia que declare o que se espera alcançar e relaciona os seguintes princípios como relevantes em sistemas de organização da informação: ideologia formulada em termos de objetivos a serem alcançados e princípios de desenho (*design*); formalização dos processos usados para organização da informação; conhecimento adquirido por meio de pesquisa, particularmente sobre desenho (*design*) e do uso de sistemas de organização; e problemas a resolver. Segundo Glushko (2013), sistemas de organização podem ser estruturados em camadas que englobam identificação de interações com os recursos, determinação dos princípios de organização que viabilizam as interações, e decisões sobre como armazenar e gerir os recursos de acordo com os princípios de organização.

Além do conceito designado pelo termo “sistema de organização da informação”, também se tem o conceito designado pelo termo “sistema de organização do conhecimento” (*knowledge organization system*). Segundo Hodge (2000), sistema de organização do conhecimento é todo tipo de esquema para organização da informação e promoção da gestão do conhecimento. Para Hjørland (2009), sistema de organização de conhecimento é sistema que organiza conceitos e suas relações semânticas. Em Stock (2010), sistema de organização do conhecimento é composto por conceitos e relações semânticas que representam terminologicamente um domínio de conhecimento com o objetivo primário de prover suporte a processo de recuperação. Segundo Stock (2010), sistema de organização do conhecimento é sistema de conceitos em um domínio, onde domínio de conhecimento (*knowledge domain*)

é uma área temática que pode ser delimitada. A seguir, são listadas classes de sistema de organização de conhecimento segundo [Stock \(2010\)](#): nomenclatura, sistema de classificação, tesouro, ontologia e folksonomia. Em [Gödert, Hubrich e Nagelschmidt \(2014\)](#), o termo “sistema de organização do conhecimento” designa “estrutura de conhecimento”, estrutura composta por conceitos relacionados, construída para suporte à indexação e recuperação da informação.

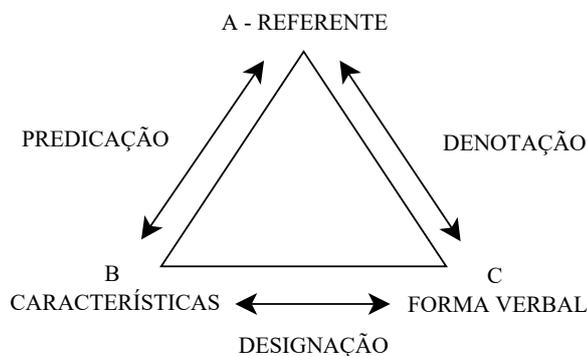
3.5.1.1 Conceito

Nesse contexto, é importante definir o que é conceito. Existem diversas definições do que é conceito e diversas teorias sobre conceito ([HJØRLAND, 2009](#); [MARGOLIS; LAURENCE, 2014](#)). Frequentemente, o termo conceito designa unidade de pensamento, entidade abstrata, construto mental que representa objeto material ou imaterial. A definição e a classificação de conceito dependem de fatores como entendimento prévio, antecedentes sociais, antecedentes culturais, conhecimento de domínio e perspectiva teórica ([CRUSE, 2011](#); [HJØRLAND; PEDERSEN, 2005](#); [ISAAC; SUMMERS, 2009](#); [STOCK, 2010](#)). Segundo a norma ISO 1087-1 2000 (E/F) *Terminology work - Vocabulary - Part 1: Theory and application* (vide [ISO \(2000\)](#)), conceito é “unidade de conhecimento criada por combinação única de características”. Característica é abstração de propriedade de objeto ou de conjunto de objetos, em que objeto é qualquer coisa perceptível ou concebível, material, imaterial ou imaginado. Por meio de enunciados sobre item de referência, se obtém características. A extensão de um conceito é a totalidade de objetos ao qual o conceito corresponde. A intensão de um conceito é o conjunto de características que constitui o conceito, é a soma total das suas características, a soma das características específicas do conceito às características dos conceitos a ele mais abrangentes. Por fim, [ISO \(2000\)](#) classifica conceito como: individual, geral, superordenado, subordinado, genérico, abrangente, específico, partitivo ou coordenado. Conceito individual corresponde a um objeto. Conceito geral corresponde a dois ou mais objetos que formam grupo por razão de propriedades em comum.

Uma teoria do conceito no contexto de sistemas para organização do conhecimento é a *Referent-oriented, Analytical Concept Theory*, desenvolvida por Ingetratur Dahlberg. Os seguintes elementos integram essa teoria do conceito: item de referência (referente), característica e forma verbal. Segundo a *Referent-oriented, Analytical Concept Theory*, a formação de conceito ocorre por reunião e compilação de enunciados verdadeiros sobre item de referência. Cada enunciado verdadeiro sobre item de referência é elemento do conceito. Conceito é síntese das características essenciais de item de referência, em que característica é propriedade enunciada de item de referência. Conceito é representado por designações e fixado por símbolo linguístico. Por fim, a soma total dos enunciados necessários e verdadeiros sobre item de referência consiste no conceito ([DAHLBERG, 1978c](#); [DAHLBERG, 1978a](#); [DAHLBERG, 1978b](#); [DAHLBERG, 1993](#); [DAHLBERG, 2009](#);

DAHLBERG, 2011). O Triângulo do conceito, representado na figura 8, ilustra elementos anteriormente citados.

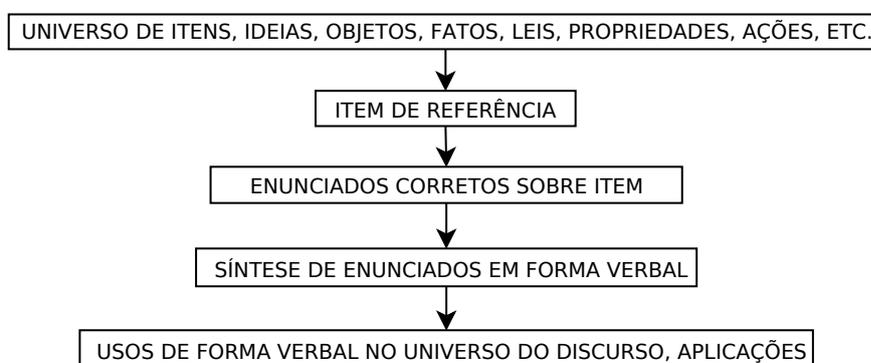
Figura 8 – Triângulo do conceito



Fonte: Adaptado de Dahlberg (1993)

A definição de um conceito pode ser nominal ou real. A definição é nominal quando tem o objetivo de fixar sentido de uma palavra, e real quando tem o objetivo de delimitar intensão do conceito, apresentar conhecimento sobre um objeto. Para estruturar uma definição real, pode ser definida uma “equação de sentido”, em que, à esquerda se encontra o que deve ser definido (*definiendum*) e à direita se encontra aquilo pelo qual algo é definido (*definiens*). No *definiens* se encontra conceito mais amplo do que aquele no *definiendum* seguido de característica especificadora (DAHLBERG, 1978c; DAHLBERG, 1978a). Por fim, a figura 9 é uma ilustração do processo de construção de conceito (DAHLBERG, 1993).

Figura 9 – Processo de construção de conceito



Fonte: Adaptado de Dahlberg (1993)

3.6 Representação da informação

A representação da informação permeia a organização da informação e influencia a recuperação da informação (GLUSHKO, 2013; JACOB, 2004). A informação pode ser representada antes ou depois de ser publicada. No primeiro caso, a representação pode ser acessada na recuperação da informação; no segundo caso, a representação é gerada a partir de informação publicada, como no caso de motores de busca (MAAREK, 2011; QIN, 2000). Segundo Dodebei (2002), objeto (aquilo que se deseja representar) e propriedade (característica daquilo que se deseja representar) são elementos na representação da informação. A representação da informação trata de relações entre objetos e propriedades e de como manipular representações em suportes. Segundo Novellino (1996), na representação da informação, ocorre substituição de entidade linguística longa e complexa por descrição abreviada.

No contexto da representação da informação, a descrição de recurso de informação procura identificar e caracterizar a essência do recurso com o objetivo de prover suporte a processos de seleção, organização, interação e manutenção (GLUSHKO, 2013). Segundo Brasethvik (2004), a descrição de recurso de informação procura capturar e comunicar significados do recurso para melhorar a recuperação da informação. A seguir, são listadas atividades que geralmente integram processo de descrição da informação: determinar escopo, determinar foco, determinar propósitos da descrição, identificar recursos, identificar propriedades de recursos, definir vocabulário controlado, definir modo de descrição, definir modo de implementação, descrever recursos, avaliar descrição, escolher termos adequados aos usuários e refletir como os recursos se descrevem (GLUSHKO, 2013; SVENONIUS, 2000).

3.6.1 Análise de assunto

O processo de representação da informação geralmente engloba análise de assunto. A análise de assunto tem como objetivos possibilitar que usuários acessem recursos de informação a partir dos assuntos desses recursos, posicionar próximos os recursos de informação que tenham assuntos em comum, prover localização lógica para recursos de informação similares, e melhorar a recuperação da informação por meio de termos e símbolos controlados (TAYLOR; JOUDREY, 2009). A análise de assunto geralmente engloba análise conceitual para determinar os assuntos tratados no recurso de informação, elaboração de enunciado que descreva o assunto, seleção de termos e símbolos com o objetivo de representar os conceitos identificados, tradução do enunciado elaborado usando os termos e símbolos selecionados. Para determinar o assunto de um recurso de informação, geralmente é necessário determinar o propósito do seu autor, identificar usuários e usos do recurso, e considerar questões que o recurso pode responder (TAYLOR; JOUDREY, 2009).

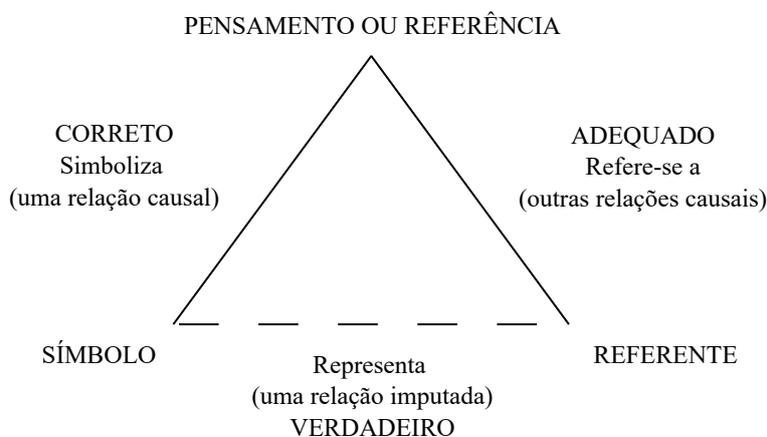
3.6.2 Linguagem para representação da informação

Existem várias definições do termo linguagem. Por exemplo, “meio sistemático para se comunicar ideias por sinais, sons, gestos ou marcas convencionadas e regras para formação de expressões aceitáveis”, “um meio de comunicação, com sintaxe e semântica, consistindo de um conjunto de representações, convenções e regras associadas, usadas para transmitir informação” (IEEE, 1990). Segundo Syal (2010), ocorrem dificuldades na definição desse termo e sugere, como alternativa, a seguinte relação de características: linguagem é meio de comunicação entre humanos; linguagem é arbitrária, pois não há relação inerente entre palavras de uma linguagem e significados ou ideias transmitidas; linguagem é sistema de sistemas; linguagem é primariamente constituída por sons vocais; linguagem difere da comunicação animal; linguagem é forma de comportamento social; linguagem é sistema de símbolos; linguagem é criativa e produtiva, na medida em que o usuário de uma linguagem nem sempre produz apenas sentenças anteriormente aprendidas; permutabilidade, na medida em que o emissor de um sinal linguístico pode também ser o receptor.

Linguagens usadas pelo ser humano nas necessidades da vida diária são frequentemente denominadas linguagens naturais (DAHLBERG, 1978c). A seguir, são listadas características dessas linguagens: regras baseadas no uso em vez de estabelecidas antes do uso, usadas por seres humanos para comunicação verbal, está ou esteve em uso em comunidade humana (IEEE, 1990; ISO, 2001; NISO, 2005). Em linguagens naturais, são usados símbolos para expressar pensamento (referência) sobre objetos (referente). O triângulo de Ogden e Richards (figura 10) expressa relações entre símbolo, referência e referente. Nos vértices desse triângulo, há fatores quando enunciado é feito ou entendido. Entre símbolo e referente, existe relação indireta mediada por pensamento (OGDEN; RICHARDS, 1923).

Linguagens são relevantes na recuperação e organização da informação. Em sistemas de recuperação e organização da informação que usam linguagem natural, os recursos de informação e as necessidades de informação dos usuários são representados por meio de vocabulários abertos e compostos por palavras e frases usadas nos domínios aos quais esses sistemas se destinam. Nesses sistemas, não ocorre controle de vocabulário, e o uso de linguagem natural possibilita especificidade na recuperação da informação. Nesses sistemas, os índices usados na indexação geralmente são termos extraídos dos recursos de informação (LANCASTER, 1986). Segundo Svenonius (2000), uma alternativa ao uso de linguagem natural é usar linguagem de representação que sistematize e imponha método à linguagem natural. Uma linguagem de representação pode englobar termos, semântica, regras de sintaxe e pragmática (uso ou aplicação da linguagem). Svenonius (2000) destaca o trabalho para construção de uma linguagem não ambígua que “sistematize e imponha método à linguagem natural e ao mesmo tempo permita aos usuários encontrar o que desejam por nomes que conhecem”.

Figura 10 – Triângulo de Ogden e Richards



Fonte: Adaptado de [Ogden e Richards \(1923\)](#)

Linguagens de representação podem ser usadas para organizar campo conceitual e, controlar dispersões léxicas, sintáticas e simbólicas. Uma linguagem de representação usada em indexação pode ser denominada “linguagem de indexação”. Essas linguagens, são usadas para representar conceitos em recursos de informação e são compostas por termos usados como pontos de acesso. A definição de uma linguagem de indexação geralmente engloba o exame de conceitos em domínios de conhecimento. A especificidade da linguagem de indexação impacta a precisão das descrições dos recursos de informação, quanto maior é a quantidade e o detalhamento dos termos, potencialmente maior é a precisão na descrição. Quando uma linguagem de representação é usada na codificação de metadados, são usados termos da linguagem como valores de elementos de metadados. Linguagem de representação usada como linguagem de busca, por sua vez, contém termos usados como requisitos de busca ([AITCHISON; BAWDEN; GILCHRIST, 2000; CAPLAN, 2003; DODEBEI, 2002; GÖDERT; HUBRICH; NAGELSCHMIDT, 2014; NISO, 2005; ROWLEY; HARTLEY, 2008; SICILIA, 2006; TAYLOR; JOUDREY, 2009](#)). Em linguagens de representação, são relevantes os princípios da monoreferencialidade e da contextualização. Segundo o princípio da monoreferencialidade, um termo só representa um conceito, enquanto que, segundo o princípio da contextualização, contexto especifica significado de termo ([DODEBEI, 2002](#)).

3.7 Classificação da informação

O termo classificação pode designar diversos conceitos em Ciência da Informação, tais como sistema de classes ordenado de acordo com conjunto de princípios para organizar entidades; processo pelo qual entidades são atribuídas a classes por meio da análise de suas características; conjunto de princípios, por meio dos quais, recursos são organizados

de acordo com seus assuntos (FEATHER; STURGES, 2003; JACOB, 2004). Hjørland e Pedersen (2005) definem classificação como triagem (*sorting*) de objetos embasada em algum critério selecionado entre propriedades dos objetos. Esses autores também destacam a importância, na qualidade da classificação, da base na qual o critério foi escolhido, motivado e substanciado. Por sua vez, Glushko (2013) define classificação, como atribuição sistemática de recursos a categorias intencionais, e categoria, como conjunto de coisas ou entidades abstratas tratadas de modo equivalente. Segundo esse autor, a classificação de recursos provê suporte à descoberta, seleção, combinação, integração, análise de recursos em sistemas de organização e facilita o entendimento de domínios. A classificação é sistemática quando segue princípios que governam a estrutura de classes e relações. Para Jones (2005), a classificação é um processo em que se perde informação detalhada sobre os objetos classificados, mas se ganha informação ao se explicitar que alguns objetos são semelhantes, que qualquer membro de uma classe pode ser tratado como possuindo propriedades que caracterizam a classe. Para Reitz (2012), classificação é processo em que ocorre divisão em hierarquias lógicas de classes, com base em características que as classes têm em comum e em características que as distinguem. Segundo Goh et al. (2009), na classificação, objetos são arrumados em classes que podem ser divididas em subclasses segundo algum princípio. Em uma classe, os objetos compartilham um grupo particular de atributos. Para Jacob (2004), categorização é processo que divide o mundo em grupos, cujos membros apresentam similaridades, e em que ocorre divisão sistemática em estrutura formalizada e potencialmente hierárquica de categorias, sendo cada categoria definida por um conjunto único de características.

Para Kwasnik (1992), assim como uma teoria explica um fato relacionando-o a outros fatos, uma classificação representa conhecimento sobre uma entidade relacionando-a a outras entidades. Uma classificação não é só um modo de representar entidades, mas também um modo de lhes impor ordem. Hjørland e Pedersen (2005) destacam que a classificação não deve se basear em descrições triviais e ingênuas, mas em conhecimento sobre contextos mais amplos de significado e produção. Também destacam a importância dos usuários da classificação terem acesso a diferentes pontos de vista, à informação a partir de diferentes pontos de vista. Segundo Jones (2005), a classificação pode ser caracterizada considerando-se a relação entre propriedades e classes, entre objetos e classes, entre classes e classes. Quanto à relação entre propriedades e classes, há classe na qual todos seus membros possuem as mesmas propriedades em comum e há classe sem essa exigência. Quanto à relação entre objetos e classes, há classe cujos objetos podem ser membros de outras classes e há classe cujos objetos são exclusivamente membros dessa classe. Quanto à relação entre classes, há classificações em que as classes são sistematicamente relacionadas umas às outras e há classificações em que não ocorre ordenamento entre as classes integrantes da classificação.

O processo de classificação da informação geralmente engloba diversas atividades.

Particularmente, atividades com os seguintes objetivos: definição de propósito; definição de critérios para descrever, diferenciar e classificar membros; definição de método para atribuir membros a classes de modo completo e não ambíguo (GLASS; VESSEY, 1995). Hjørland (2012) relaciona as seguintes atividades em processo de classificação: definir classes, determinar relações entre classes, e atribuir elementos a classes. Também destaca a interdependência com processo que englobe atividades para definir conceitos, determinar relações semânticas entre conceitos, e determinar elementos atribuídos aos conceitos. No domínio bibliográfico, por meio de classificação, mantêm-se próximos recursos sobre assuntos relacionados visando facilitar a recuperação da informação. Na classificação bibliográfica, são geralmente usados códigos para representar assuntos (FEATHER; STURGES, 2003; ROWLEY; HARTLEY, 2008).

3.7.1 Classificação enumerativa da informação

Em um esquema de classificação enumerativo, são listadas classes de assuntos específicos (HJØRLAND, 2007a). A classificação enumerativa da informação procura listar exaustivamente as classes de interesse. Nesse processo dedutivo e descendente (*top-down*), o domínio é subdividido segundo determinados princípios e pontos de vista. São realizadas subdivisões sucessivas de classes e classes são organizadas em estrutura hierárquica (CAPLAN, 2003; CHU, 2010; GIESS; WILD; MCMAHON, 2008; GOH et al., 2009; ROWLEY; HARTLEY, 2008). São possíveis desvantagens da classificação enumerativa: dependência de ponto de vista; imposição de princípios de divisão; o ponto de vista de determinado grupo de usuários é privilegiado em detrimento de outros; quantidade elevada de classes necessárias no esquema de classificação em certos campos de conhecimento; ambiguidades na classificação de assuntos compostos; necessidade de revisão regular para incluir assunto emergente não considerado na construção do esquema de classificação; esforço significativo de manutenção quando o campo de conhecimento classificado é dinâmico e muda frequentemente; quando surge um novo conceito, é necessário aguardar que um termo para o conceito seja provido pelo esquema de classificação (GIESS; WILD; MCMAHON, 2008; GOH et al., 2009). Por fim, para Giess, Wild e McMahon (2008), a imposição de pontos de vista na classificação enumerativa pode ser uma desvantagem em engenharia, pois os recursos de informação nessa área de conhecimento frequentemente abordam assuntos compostos e o resultado da classificação enumerativa dependerá do ponto de vista adotado.

3.7.2 Classificação facetada da informação

A noção de faceta é embasada na crença de que há mais de um modo de se ver o mundo, que classificações são provisórias e dinâmicas, que construir classificações capazes de acomodar novos fenômenos é um desafio (KWASNIK, 1999). Para Albrechtsen (1992), facetadas podem ser categorias genéricas independentes de domínio, que podem ser definidas

a priori; categorias específicas dependentes de domínio, primariamente definidas a *posteriori* considerando-se conhecimento e investigação sobre categorias em domínio; ou categorias específicas, articuladas por usuários ao expressarem necessidades de informação. Segundo [Albrechtsen \(1992\)](#), um sistema de classificação pode englobar facetas genéricas e facetas específicas a domínios. O termo “classificação facetada”, é definido por [Pearce-Moses \(2005\)](#) como “sistema para organizar materiais em categorias, baseado em combinação sistemática de características mutuamente exclusivas e coletivamente exaustivas dos materiais, e apresentar características em um modo que mostre as suas relações”. Um modo de descrever a classificação facetada é descrever a metodologia analítico-sintética. Nessa metodologia, análise designa divisão de assunto em seus conceitos básicos e síntese designa combinação de unidades relevantes e conceitos para descrever assunto ([HJØRLAND, 2008b](#)).

Na classificação facetada, ocorre o agrupamento em facetas usando uma característica ou princípio de divisão por vez. Relações entre termos podem ser intra-faceta ou inter-facetadas. Termos em uma faceta podem ser sinônimos ou hierarquicamente relacionados. Relações entre termos em diferentes facetas são relações de associação. Termos de diferentes facetas podem ser usados na descrição de recurso de informação segundo diferentes perspectivas. Cada recurso de informação pode ser descrito segundo propriedades de diferentes facetas ([AITCHISON; BAWDEN; GILCHRIST, 2000](#); [BROUGHTON, 2006b](#); [GARSHOL, 2004](#); [GLUSHKO, 2013](#); [BARRE, 2010](#); [SVENONIUS, 2000](#)). Segundo [Goh et al. \(2009\)](#), o desenvolvimento de esquema de classificação facetada pode seguir abordagem descendente ou ascendente, mas consideram que a ascendente tem maior probabilidade de garantir independência de conceitos entre facetas, ausência de duplicações e descrições adequadas. Finalmente, para [Giess, Wild e McMahon \(2008\)](#), o desenvolvimento de esquema de classificação facetada por processo apenas ascendente pode não ser viável, pois geralmente é necessário atualizar o esquema quando conceitos são introduzidos em decorrência, por exemplo, de mudança de entendimento ou do uso de novas fontes de informação.

[Broughton \(2006b\)](#) considera que a classificação facetada é parte de diversos métodos de recuperação da informação e destaca os seguintes contextos em que ela é importante: construção de vocabulários controlados, organização e apresentação da informação em sítios na *World Wide Web*, recuperação automatizada da informação, construção de ferramentas de navegação, e construção de ontologias. Segundo [Broughton \(2006b\)](#), na construção de vocabulário controlado, um esquema de classificação facetada pode servir como ponto de partida, uma vez que relações hierárquicas e associativas geralmente já foram definidas no esquema. Segundo a fonte [NISO \(2005\)](#), a classificação facetada é útil em campos novos ou emergentes nos quais conhecimento do domínio é incompleto ou relacionamentos entre recursos de informação são desconhecidos ou definidos pobremente; áreas interdisciplinares em que há diferentes perspectivas dos recursos de

informação ou são necessárias combinações de conceitos; vocabulários em que são requeridas diversas hierarquias, mas que podem ser inadequadas por ser difícil definir suas fronteiras; classificação de recursos de informação digitais; recursos de informação em que localização e co-localização não são requisitos importantes; vocabulário controlado composto por grande quantidade de termos. Barre (2010) considera que a classificação facetada é importante em ambientes digitais, particularmente no suporte à recuperação da informação e relaciona indexação, metadados, sistemas de busca e navegação como aplicações da classificação facetada. Giess, Wild e McMahan (2008) destacam a redução de dependências quanto a pontos de vista específicos como vantagem da classificação facetada da informação. Por sua vez, Goh et al. (2009) destacam que classificar recursos sobre novos assuntos é mais fácil na classificação facetada da informação do que na classificação enumerativa da informação, pois, na classificação facetada da informação, isso pode ser feito combinando-se termos de esquema de classificação. Também destacam a possibilidade de diversos pontos de vista serem representados por meio de diferentes esquemas de classificação. Broughton (2006b) ressalta a estrutura lógica rigorosa e a compatibilidade com interface gráfica como vantagens da classificação facetada em ambiente digital.

É importante destacar que existem críticas à teoria da classificação facetada da informação. Por exemplo, em Wild, Giess e McMahan (2009) são apresentadas as seguintes: divergentes interpretações sobre a noção de faceta, falta de orientação pragmática de como gerar conjunto de facetas, diferença entre descrição facetada e análise de facetas, construção de facetas ascendente versus descendente, diferença entre abordagem geradora e critérios de avaliação. Em Giess, Wild e McMahan (2008), são descritas como dificuldades no desenvolvimento de sistema de classificação facetada para recursos de informação em engenharia: a falta de metodologia para a construção de sistema de classificação facetada que apresente as propriedades consideradas requisitos desse tipo de sistema, e a escassez de literatura sobre aplicações e sobre exemplos de análise de facetas. Por fim, para Goh et al. (2009), são desvantagens da classificação facetada da informação: dificuldade em posicionar recursos de informação recuperados em ordem de importância devido ao uso de termos em diferentes facetas, e a possibilidade de termos serem selecionados em diferentes facetas sem que exista sentido prático.

3.7.2.1 Teoria da Classificação Facetada da Informação

A Teoria da Classificação Facetada da Informação, também denominada Teoria das Facetas, engloba a “classificação facetada da informação” e a “análise de facetas” (BARRE, 2010). Um pioneiro da teoria da classificação facetada foi Shiyali Ramamrita Ranganathan, matemático e bibliotecário nascido na Índia em 1892, que desenvolveu a Classificação de *Colon* e publicou sobre a teoria da classificação facetada da informação, principalmente nos anos 1930. Outro marco na história da teoria da classificação facetada ocorreu em 1956, no Reino Unido, quando membros do *Classification Research Group* (CRG) passaram

a usar a “análise de facetas” (*faceted analysis*) em diversas classificações. Trabalhos de membros do CGR contribuíram para avanços na teoria da classificação facetada e para o desenvolvimento de sistemas de indexação. Para S. R. Ranganathan, a construção de sistema de classificação pode ser dividida em planos: Plano das Ideias, Plano Verbal e Plano Notacional. O Plano das Ideias engloba análise do campo do assunto em suas partes componentes, enquanto o Plano Verbal engloba escolha de terminologia para expressar as partes componentes do campo do assunto. O Plano Notacional engloba processo para expressar as partes componentes do campo do assunto por meio de notação. Os planos propostos por S. R. Ranganathan organizam a análise de facetas, associando cânones que regulam o trabalho aos planos. Ao longo dos anos, a teoria da classificação facetada da informação tem sido usada, por exemplo, no desenvolvimento de sistemas de classificação, de tesouros para recuperação da informação e no desenvolvimento de sistemas de indexação (BCA, 2015; BROUGHTON, 2006b; RANGANATHAN, 1967; SPITERI, 1998).

3.7.2.2 Análise de facetas

O termo “análise de facetas” pode designar processo para construção de classificações facetadas. Processo esse que resulta em conjunto de facetas para criar uma estrutura de classificação facetada (GIESS; WILD; MCMAHON, 2008). Segundo Aitchison, Bawden e Gilchrist (2000), a análise de facetas é um processo de classificação sistemática da informação que usa técnicas analíticas e sintéticas (*analytico-synthetic*). Esse processo usa técnicas sintéticas, pois é possível expressar conceitos não explicitamente enumerados, os termos que representam conceitos podem ser combinados para representar conceitos não explicitamente enumerados. Usa técnicas analíticas, pois é estruturado de modo que números de classes representam conceitos simples que são organizados em categorias definidas por meio de processo rigoroso. Ainda segundo Aitchison, Bawden e Gilchrist (2000), a construção de esquemas de classificação facetada, em áreas de assunto especiais, resulta em fontes de terminologia e em estruturas úteis na construção de vocabulários controlados.

Denton (2009) sugere as seguintes fases em desenvolvimento de esquema de classificação facetada da informação: coleta de domínio, fase responsável por coletar amostra representativa de entidades; listagem de entidade, fase responsável por listar entidades, decompor descrições e reordenar palavras, separar sentenças e frases em conceitos, isolar conceitos; criação de facetas, fase responsável por examinar termos resultantes das fases anteriores e identificar categorias que englobem entidades, estudar categorias e refiná-las em um conjunto de facetas mutuamente exclusivas e exaustivas; organização das facetas, fase responsável por estabelecer organização levando em consideração princípios listados e definir controle de vocabulário; ordem de citação, fase responsável por definir ordem de citação das facetas; classificação, fase responsável por analisar entidades no domínio e classificá-las por meio do uso do esquema de classificação desenvolvido; revisão, teste e

manutenção.

Com base em Vickery (1966) e Vickery (1960), Barre (2010) relaciona as seguintes atividades no desenvolvimento de esquema de classificação facetada da informação: definir campo do assunto, formular facetas, amplificar e estruturar facetas, construir notas de escopo, organizar facetas, construir e adequar notação. Definir campo do assunto engloba identificar entidades e aspectos delas que sejam de interesse. Formular facetas engloba elaborar lista de termos candidatos a partir de material representativo e organizar esses termos em grupos. Amplificar e estruturar facetas engloba estruturar hierarquicamente termos. Construir notas de escopo engloba definir termos e facetas. Organizar facetas engloba estabelecer organização entre facetas. Construir e adequar notação engloba definir notação a ser usada quando o material for classificado.

Em Kwasnik (1999), são relacionadas as seguintes atividades como atividades típicas de processo de análise de facetas: escolher facetas, desenvolver facetas, analisar entidades usando facetas e desenvolver ordem de citação. Aitchison, Bawden e Gilchrist (2000) sugerem que, na análise de um campo de assunto, os conceitos sejam inicialmente divididos em facetas abrangentes que representem categorias fundamentais, que essas facetas abrangentes sejam analisadas, que termos sejam alocadas às mesmas, e que esses termos sejam organizados em subfacetas. Aitchison, Bawden e Gilchrist (2000) também sugerem que a ordem das facetas reflita especificidade crescente, complexidade e correção. A ordenação dos conceitos em subfacetas pode também seguir esse mesmo padrão. Finalmente, em Spiteri (1998) é proposto um modelo simplificado para análise de facetas, o qual incorpora princípios propostos por S. R. Ranganathan e pelo CRG. O quadro 1 relaciona princípios que integram esse modelo.

Quadro 1 – Princípios para análise de facetas

Princípios do Plano das Idéias	Princípios do Plano Verbal	Princípios do Plano Notacional
Princípios para escolha de facetas	Contexto Atualidade (<i>currency</i>)	Sinonímia Homonímia Hospitalidade Ordem de preenchimento
Diferenciação Relevância Determinabilidade (<i>ascertainability</i>) Permanência Homogeneidade Excusividade mútua Categorias fundamentais		
Princípios para ordem de citação e foco		
Sucessão relevante Sucessão consistente		

Fonte: Adaptado de Spiteri (1998)

Na análise de facetas, uma lista de categorias fundamentais, ou de categorias que

ocorrem frequentemente, reduz a chance de omissões e promove análise mais completa do domínio. Há diversas listas de categorias fundamentais em diferentes domínios, que geralmente refletem interesses de usuários, domínios e características de entidades analisadas. A aplicação de cada categoria produz uma faceta (BROUGHTON, 2006b; KWASNIK, 1999; BARRE, 2010; SVENONIUS, 2000). Existem categorias fundamentais aplicáveis à classificação da informação em diversos domínios, assim como existem categorias fundamentais aplicáveis à classificação da informação em domínios específicos. Categorias fundamentais aplicáveis a diversos domínios são propostas, por exemplo, em Ranganathan (1967), Vickery (1966), Barre (2010), Broughton e Slavic (2007) e Aitchison, Bawden e Gilchrist (2000). O quadro 2 contém nomes de categorias fundamentais identificadas nessas fontes de informação, os nomes das categorias nesse quadro estão mantidos no idioma inglês.

Quadro 2 – Categorias aplicáveis a diversos domínios

FONTES	Ranganathan	CRG	Aitchison
CATEGORIAS	<i>Personality</i>	<i>Thing</i>	<i>Entities (things, objects)</i>
	<i>Matter</i>	<i>Kind</i>	<i>Actions</i>
	<i>Energy</i>	<i>Part</i>	<i>Space</i>
	<i>Space</i>	<i>Property</i>	<i>Time</i>
	<i>Time</i>	<i>Material</i>	
		<i>Process</i>	
		<i>Operation</i>	
		<i>Patient</i>	
		<i>Product</i>	
		<i>By-product</i>	
		<i>Agent</i>	
		<i>Space</i>	
		<i>Time</i>	

Fonte: Elaborado pelo autor

Em Broughton (2006a), são sugeridas relações entre categorias propostas por S. R. Ranganathan e categorias propostas por membros do CRG. No quadro 3, são apresentadas as relações entre essas categorias. Quanto às categorias fundamentais para domínios específicos, existem categorias definidas, por exemplo, para os seguintes domínios: Arte e Arquitetura, usadas em *Art & Architecture Thesaurus* (AAT) (THE GETTY RESEARCH INSTITUTE, 2015); Engenharia de Software, descritas em Prieto-Diaz e Freeman (1987) e Albrechtsen (1992); classificação de tarefas, descritas em Li e Belkin (2008); Engenharia, descritas em Giess, Wild e McMahon (2008). No quadro 4, são apresentados nomes de categorias fundamentais identificadas nessas fontes de informação, os nomes das categorias nesse quadro estão mantidos no idioma inglês.

Quadro 3 – Relações entre categorias

FONTES	Ranganathan	CRG
CATEGORIAS	<i>Personality</i> <i>Matter</i> <i>Energy</i> <i>Space</i> <i>Time</i>	<i>Thing</i> <i>Kind</i> <i>Part</i> <i>Property</i> <i>Material</i> <i>Process</i> <i>Operation</i> <i>Patient</i> <i>Product</i> <i>By-product</i> <i>Agent</i> <i>Space</i> <i>Time</i>

Fonte: Adaptado de Broughton (2006a)

Quadro 4 – Categorias aplicáveis a domínios específicos

FONTES	AAT	Prieto-Diaz	Albrechtsen	Li	Giess et al.
CATEGORIAS	<i>Associated Concepts</i> <i>Physical Attributes</i> <i>Styles and Periods</i> <i>Agents</i> <i>Activities</i> <i>Materials</i> <i>Objects</i> <i>Brand Names</i>	<i>Function</i> <i>Objects</i> <i>Medium</i> <i>System type</i> <i>Functional area</i> <i>Setting</i>	<i>Scientific Paradigm</i> <i>Method / Technical Approach</i> <i>Scope of Product or Application</i> <i>Type of Product or Application</i> <i>Form</i> <i>Technical Environment</i> <i>Application Area</i> <i>Task Area</i> <i>Status</i>	<i>Source of task</i> <i>Task doer</i> <i>Time</i> <i>Product</i> <i>Process</i> <i>Goal</i> <i>Task characteristics</i> <i>User's perception of task</i> <i>Knowledge of task procedure</i>	<i>Types of process</i> <i>Materials</i> <i>Costs</i> <i>Lead times</i> <i>Production rate</i> <i>Production volume</i>

Fonte: Elaborado pelo autor

3.7.3 Classificação social da informação

Com o advento da Internet, a organização da informação também tem sido realizada pela população em geral (SVENONIUS, 2000), por exemplo, na etiquetagem (*tagging*) social, onde os usuários atribuem etiquetas a recursos para organizar e recuperar informação de modo colaborativo (CHU, 2010; ROWLEY; HARTLEY, 2008; TAYLOR; JOUDREY, 2009). Uma vez atribuídas, as etiquetas aos recursos de informação, podem ser usadas na recuperação da informação. Etiquetar é prover termo, denominado etiqueta (*tag*), para descrever recurso. Na etiquetagem, podem ser usadas quaisquer propriedades e vocabulários para descrever recursos; e não necessariamente o vocabulário usado é controlado. Os termos são geralmente criados com a imposição de poucos controles e de modo que não

necessariamente é colaborativo (GLUSHKO, 2013; HARPRING, 2010; HEDDEN, 2010).

O termo folksonomia (*folksonomy*) designa conjunto de etiquetas resultantes de etiquetagem individual ou colaborativa (ABBAS, 2010). Folksonomias são geralmente construídas em ambientes sociais compartilhados e abertos. Frequentemente, são apresentadas como agregados gerados considerando-se frequências de uso de etiquetas (CHU, 2010; GLUSHKO, 2013; WAL, 2007). Uma nuvem de etiquetas (*cloud of tags*) é um meio de representar agregados de etiquetas. A figura 11 apresenta uma nuvem de etiquetas construída com a ferramenta *WordClouds.com*. Entre as potenciais vantagens de folksonomias, destacam-se: suporte a conceitos emergentes, atuais e diversos; suporte à navegação e a vocabulários de usuários; possibilidade de usuários participarem sem treinamento; redução de barreiras à cooperação entre usuários; elevada probabilidade dos termos refletirem aqueles usados pela comunidade; custos de desenvolvimento e manutenção menores do que aqueles de vocabulários controlados; e possibilidade de pontos de acesso adicionais não incluídos em vocabulários controlados. Algumas das potenciais desvantagens de folksonomias são: termos com natureza pessoal ou informal, redundância entre termos, erros ortográficos, ausência de controle de sinônimos e homônimos, diferentes usuários podem atribuir etiquetas a um mesmo conteúdo em diferentes níveis de abstração, ausência de controle das formas dos termos, ausência de relações hierárquicas ou associativas entre termos, ausência de indicações sobre termos preferidos, e ausência de padronização (ABBAS, 2010; DAVIES, 2010; HARPRING, 2010; MATHES, 2004; ROWLEY; HARTLEY, 2008; TAYLOR; JOUDREY, 2009). Finalmente, Shiri (2012) relaciona casos em que folksonomias foram usadas para complementar vocabulários controlados e prover pontos de acesso adicionais.

Figura 11 – Exemplo de nuvem de etiquetas



Fonte: Elaborado pelo autor

3.8 Indexação da informação

No processo de indexação, são analisados recursos de informação para determinar assuntos e expressá-los concisamente, ocorre análise do recurso e tradução do resultado da análise por meio de um vocabulário. Em geral, a indexação envolve seleção de termos preferidos, a partir de vocabulários controlados ou de outras fontes, para descrever os recursos de informação e visa tornar a recuperação da informação mais eficiente. Nesse contexto, índices são recursos descritivos definidos para possibilitar que coleções sejam eficientemente pesquisadas, termos são usados para representar recursos de informação. Para isso, inicialmente, o responsável pela indexação do recurso adquire familiaridade com o assunto do recurso a ser indexado. Em seguida, identifica e representa conceitos por termos obtidos do recurso ou de vocabulários controlados. Os termos obtidos de vocabulários controlados, geralmente, são denominados descritores (BAEZA-YATES; RIBEIRO-NETO, 2011; CHU, 2010; GLUSHKO, 2013; NISO, 2005; ROWLEY; HARTLEY, 2008; TAYLOR; JOUDREY, 2009). Portanto, o processo de indexação pode ser decomposto em fases. Inicialmente, o recurso é analisado com o objetivo de se identificar o seu assunto. Em seguida, é descrito o assunto e traduzida a descrição do assunto em linguagem apropriada (MAI, 2001). Para representar certos recursos, pode ser necessário combinar termos por processos denominados pré-coordenação ou pós-coordenação. Na pré-coordenação, os termos são combinados na indexação, enquanto que, na pós-coordenação, os termos são combinados nas consultas. A coordenação possibilita maior precisão na definição de conceitos e na recuperação da informação (AITCHISON; BAWDEN; GILCHRIST, 2000; HEDDEN, 2010).

A indexação pode ser manual ou automatizada. A seguir, são listados fatores que sugerem a indexação ser realizada por pessoas: quando se busca qualidade e precisão da indexação e da recuperação da informação, quando houver quantidade gerenciável de recursos, recursos não textuais, recursos de variados tipos e formatos, e recursos sobre variadas áreas de assunto. É sugerida a automação, quando houver grande quantidade de recursos, mudanças frequentes em conteúdos, necessidade de velocidade na indexação, recursos de tipos e formatos comuns, recursos sobre uma área de assunto, e recursos textuais (HEDDEN, 2010). Finalmente, Kowalski e Maybury (2002) consideram abstração de conceitos e julgamento do valor da informação como usos primários da indexação manual, e relacionam estabelecimento de relações de causa e efeito, e determinação da importância relativa de conceitos, como dificuldades encontradas na indexação automática.

Lancaster (2004) observa que muitas vezes são inexpressivas as distinções entre indexação, catalogação e classificação, pois a classificação, em sentido amplo, permeia atividades pertinentes ao armazenamento e recuperação da informação. Para esse autor, a dificuldade pode decorrer da incapacidade de distinguir entre análise conceitual e tradução. A tradução representa resultados da análise conceitual por meio de termos. Ainda segundo

esse autor, a indexação de assuntos é conceitualmente idêntica à catalogação de assuntos. A atividade é classificar assunto, “formar classes de objetos com base em seu conteúdo temático”. A indexação é um processo de classificação e os termos usados na indexação para representar assuntos são rótulos que identificam classes. Por sua vez, Loureiro (2007) destaca a relação entre organização da informação e classificação da informação. Ressalta que classificar é inerente a organizar e que a classificação permeia a organização da informação. Ainda segundo Loureiro (2007), na indexação, quando o assunto do recurso é determinado e são atribuídos termos a ele, ocorre atribuição do recurso a classes identificadas por esses termos. Portanto, esses termos são rótulos identificadores de classes, podem ser denominados rótulos de classes, mas são geralmente denominados índices ou descritores.

3.9 Gestão da informação

Os termos gestão de dado, gestão de recurso de dado, gestão da informação e gestão de recurso de informação, geralmente designam processo ou programa (iniciativa que inclui projetos relacionados) que planeja e executa políticas, práticas e projetos que adquirem, controlam, protegem, entregam e ampliam o valor de ativos de dado e de informação. Nesse contexto, ativo (*asset*) é “recurso com valor reconhecido sob controle de indivíduo ou organização”. São objetivos desse processo ou programa: entender necessidades de informação da organização e de interessados (*stakeholder*); capturar, armazenar, proteger e garantir integridade de ativos de dado e de ativos de informação; continuamente melhorar a qualidade de dado e informação; garantir privacidade e confidencialidade; prevenir uso não autorizado ou inapropriado de dado ou informação; maximizar uso efetivo e valor de ativos de dado e de ativos de informação. A gestão da informação engloba a gestão de conteúdos, que visa organizar, categorizar e estruturar acesso a conteúdo. São atividades da gestão de conteúdos: definir e manter taxonomias, desenvolver metadados, descrever conteúdo por metadados, prover meios para acessar conteúdos, prover meios para recuperar conteúdos e governar para qualidade de conteúdos. Por fim, nesse contexto, “sistema para gestão de conteúdo” é sistema para coletar, organizar, indexar e recuperar conteúdo (DAMA, 2010).

4 Arquitetura da informação

Este capítulo aborda conceitos designados pelo termo “arquitetura da informação”. Entre os elementos que integram este capítulo, é possível destacar os seguintes: definições do termo “arquitetura”, definições do termo “arquitetura da informação”, descrições de elementos da prática da arquitetura da informação, de elementos da disciplina Arquitetura da Informação (AI) e de elementos da Teoria Geral da Arquitetura da Informação (TGAI).

4.1 Definições de arquitetura

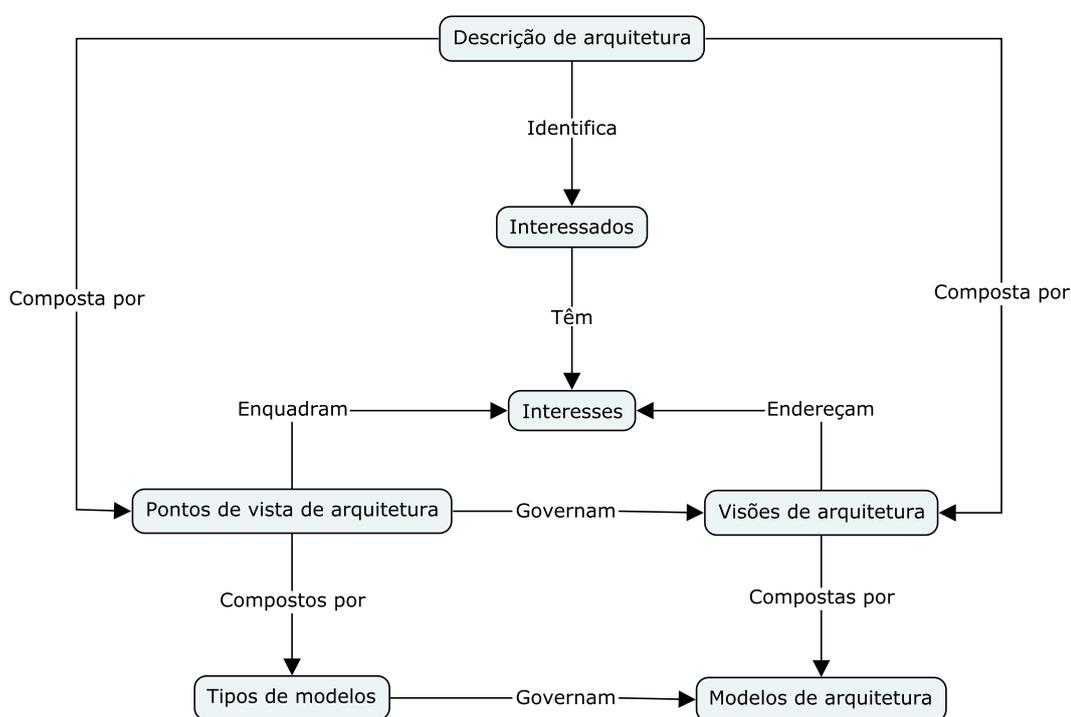
Arquitetura pode ser definida como “arte e técnica de organizar espaços e criar ambientes para abrigar os diversos tipos de atividades humanas, visando também determinada intenção plástica” (HOUAISS; VILLAR; FRANCO, 2001). Na norma ISO (2011e), é proposto modelo conceitual de descrição de arquitetura. O mapa de conceitos na figura 12 foi elaborado a partir desse modelo. Segundo as fontes IEEE (1990) e ISO (2011e), “arquitetura de sistema” é estrutura organizacional de um sistema, e engloba conceitos fundamentais ou propriedades de sistema em seu ambiente (contexto determinante de influências sobre o sistema) incorporadas em seus elementos, relações, princípios de desenho (*design*) e evolução.

4.2 Definições de arquitetura da informação

A definição original do termo “arquitetura da informação” é frequentemente atribuída a Richard Saul Wurman, que, em 1976, definiu o conceito referenciado por esse termo como ciência e arte de criar instruções para espaços organizados (CAWKELL, 2003). Entretanto, segundo Resmini e Rosati (2012), o uso do termo informação com o termo arquitetura é mais antigo. Em artigo de 1964 denominado *Architecture of the IBM System/360*, arquitetura é definida como “a estrutura conceitual e comportamento funcional, a distinguir a organização de fluxos de dados e controles, desenho (*design*) lógico e implementação física”. Outro uso do termo *architecture of information* ocorreu em 1970, no *Xerox Palo Alto Research Center* (PARC).

Segundo Albuquerque e Lima-Marques (2011), o termo “arquitetura da informação” pode designar disciplina, produto ou objeto de estudo de disciplina. Em Resmini e Rosati (2012), o termo designa prática profissional e campo de estudos focados na solução de problemas de acesso e uso à informação. Albuquerque (2010) considera que correntes e definições do termo “arquitetura da informação” podem ser agrupadas nas seguintes perspectivas: arquitetura da informação como desenho (*design*) para ambiente específico;

Figura 12 – Modelo de descrição de arquitetura



Fonte: Elaborado pelo autor a partir de ISO (2011e)

arquitetura da informação como organização de espaço de informação de qualquer tipo; e arquitetura da informação como percepção da realidade. Segundo Siqueira (2012), definições do termo “arquitetura da informação” são variadas e podem expressar conceitos diferentes.

Dillon (2002) considera que existem duas perspectivas principais para definir o termo. A primeira estabelece paralelos entre arquitetura da informação, classificação da informação e recuperação da informação, enquanto a segunda assume que espaços de informação requerem desenho (*design*) em múltiplos níveis, e que a experiência de vida dos usuários nesses espaços é uma preocupação do arquiteto da informação. Para Dillon (2002), independentemente de como o campo identificado pelo termo “arquitetura da informação” seja rotulado, tal como no mundo físico, o domínio da informação será província da arquitetura. A seguir, são apresentadas definições para o termo “arquitetura da informação”.

- (a) Desenho (*design*) estrutural de ambientes compartilhados de informação; combinação de sistemas de organização, rotulagem, busca e navegação em sítios na *Web* e em *intranets*; arte e ciência de dar forma a produtos de informação e experiências para prover suporte à usabilidade e *findability*; disciplina emergente e comunidade de

prática focada em trazer princípios de desenho (*design*) e arquitetura ao cenário digital (MORVILLE; ROSENFELD, 2006).

- (b) Arte e ciência de organizar informação com o objetivo de auxiliar pessoas a satisfazerem eficientemente as suas necessidades de informação; engloba investigação, análise, desenho (*design*) e implementação (HAGEDORN, 2000).
- (c) Termo usado com o objetivo de identificar processo de desenho (*design*), implementação e avaliação de espaços de informação humanamente e socialmente aceitáveis aos seus interessados (*stakeholder*) (DILLON, 2002).
- (d) Como disciplina, arquitetura da informação é uma abordagem especializada para desenho (*design*) de sistemas de organização e interações associadas, para produtos e serviços intensivos em informação. Como atividade, a arquitetura da informação consiste no desenho (*design*) de uma organização da informação abstrata e efetiva e exposição dessa organização com o objetivo de facilitar o uso da informação (GLUSHKO, 2013).
- (e) Prática profissional e campo de estudos focado na solução de problemas de acesso e uso da vasta quantidade de informação atualmente disponível (RESMINI; ROSATI, 2012).

4.3 Elementos da prática da arquitetura da informação

Spencer e Featherstone (2010) relacionam as seguintes responsabilidades da arquitetura da informação na prática: (a) organizar conteúdos ou recursos de informação; (b) descrever conteúdos ou recursos de modo claro; e (c) prover meios para que as pessoas cheguem a conteúdos ou recursos. Morville e Rosenfeld (2006) ressaltam a importância de usuários, contexto e conteúdo, como elementos para representação da arquitetura da informação. Também ressaltam a importância do desenho (*design*) dos seguintes elementos: (a) sistema de organização, que estabelece como a informação é categorizada; (b) sistema de rotulagem, que estabelece como a informação é representada; (c) sistema de navegação, que estabelece aspectos de navegação e movimentação no espaço de informação; e (d) sistema de busca, que estabelece aspectos da busca por informação.

4.4 Elementos da disciplina Arquitetura da Informação

Segundo Albuquerque e Lima-Marques (2011), como disciplina, o termo “Arquitetura da Informação” refere-se a esforço sistemático de identificação de padrões e criação de métodos para definir espaços de informação, cujos propósitos são representação e manipulação de informação, bem como criação de relações entre entidades linguísticas

para definir esses espaços de informação. A disciplina Arquitetura da Informação é um programa de investigação epistemológica da realidade que busca identificar e relacionar, no mínimo e necessariamente, as seguintes dimensões no problema: forma, contexto, manifestação e significado, definidas a seguir. Forma expressa a ideia de organização que dá a disposição de relações entre elementos constituintes. Contexto expressa a inter-relação de circunstância ou coisas que acompanham um fato ou situação. Manifestação entende-se como fato, coisa, fenômeno ou registro, ou seja, refere-se à existência da coisa em si. Por fim, significado é uma correlação semântica atribuída ao fato, coisa, ou fenômeno, dentro de uma intencionalidade funcional ou subjetiva.

Albuquerque e Lima-Marques (2011) também propõem um modelo em camadas para agrupamento de ferramentas para análise e compreensão da Arquitetura da Informação. Segundo essa fonte, qualquer instrumento ou ferramenta para análise de algum aspecto proposto em uma camada, encontra nicho nesse modelo. As camadas são as seguintes: Episteme, Análise, Tratamento, Representação, Armazenamento, Organização, Recuperação e Aplicação. Episteme oferece arcabouço teórico para determinação dos conceitos a serem adotados em todos os níveis; Análise refere-se à análise do contexto, considerando elementos constituintes e ambiente; Tratamento refere-se a tratamento de conteúdos; Representação cuida da descrição dos conteúdos por meio de padrões; Armazenamento, considera questões de armazenamento dos estoques de conteúdos; Organização cuida dos fluxos de relação entre componentes; Recuperação considera mecanismos de recuperação da informação; Aplicação permeia demais níveis na medida em que abarca ferramentas tecnológicas que atendem a todos os níveis.

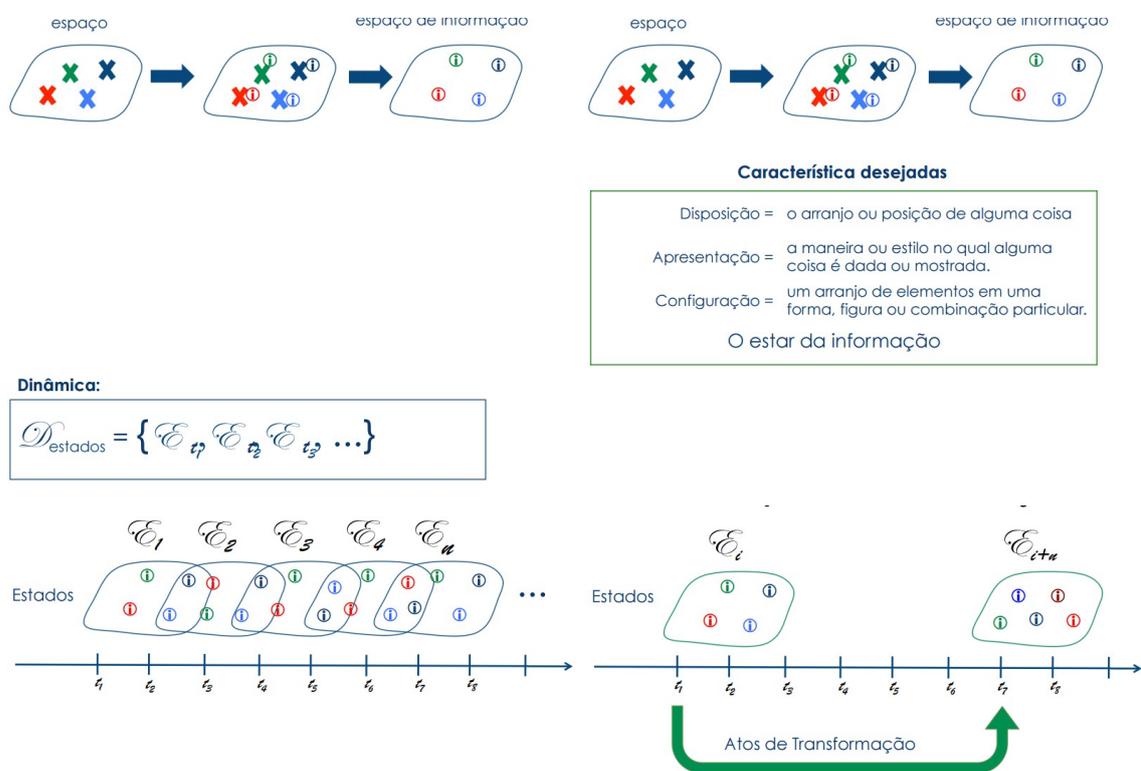
Finalmente, Lacerda e Lima-Marques (2014) relacionam as seguintes características da disciplina Arquitetura da Informação: desenho (*design*) intencional de espaços de informação em seus aspectos cultural, social e tecnológico, como objeto principal de interesse, com o objetivo de promover as experiências dos usuários; de natureza interdisciplinar, com métodos, modelos e teorias influenciadas por outras disciplinas ou derivadas de outras disciplinas; existência de grupos de pessoas focados em ensino, pesquisa e prática de questões de natureza similar, que se encontram regularmente em eventos dedicados ao assunto, que reconhecem existir algo chamado “arquitetura da informação”, que se denominam “arquitetos da informação” ou que assumem cargos assim denominados; papel significativo e necessário na sociedade; alterações de desafios devido ao surgimento de espaços de informação ubíquos (em toda parte e ao mesmo tempo).

4.5 Elementos da Teoria Geral da Arquitetura da Informação

Lima-Marques (2011) propõe um arcabouço teórico para a disciplina Arquitetura da Informação, com os seguintes elementos: espaço, estado, dinâmica, transformação e ato

de transformação. A figura 13 representa elementos desse arcabouço teórico. Segundo esse arcabouço teórico, não há espaço sem distinção; estabelecida distinção é possível indicar espaços; um espaço distinguido tem estado e conteúdo; conteúdo é composto por coisas que têm propriedades; estado é configuração única de informação em intervalo de tempo; espaço de informação é conjunto de informação distinguida em espaço distinguido; transformação é conjunto de eventos, aplicado a estado particular, de modo a provocar mudanças em estados futuros; ato de transformação é conjunto de eventos aplicado a estado particular, por sujeito, de modo a provocar mudanças em estados futuros. Arquitetura da informação consiste na configuração de estados dos elementos constituintes da coisa em si e das suas propriedades, caracterizada pela espaço-temporalidade de informação distinguida.

Figura 13 – Elementos de arcabouço teórico



Fonte: Lima-Marques (2014)

Finalmente, é importante destacar que, segundo Lima-Marques (2011), a disciplina Arquitetura da Informação é aplicável a qualquer espaço de informação e os seguintes termos designam casos dessa disciplina: (a) *Enterprise Information Architecture*, (b) *Information Assurance Architecture*, (c) *e-Gov Information Architecture*, (d) *Federal Enterprise Architecture*, (e) *Strategic Information Architecture*, (f) *Supply Chain Information Architecture*, (g) *Web Information Architecture*, e (h) *Information Security Architecture*.

5 Vocabulários controlados

Este capítulo tem o objetivo de apresentar uma visão abrangente sobre vocabulários controlados. Este capítulo contém definições do termo “vocabulário controlado”, informação sobre propósitos e usos de vocabulários controlados, controle de vocabulário na recuperação da informação, termos em vocabulários controlados, relações em vocabulários controlados, estruturas de vocabulários controlados, desenvolvimento de vocabulários controlados, reuso de vocabulários controlados, tecnologias para representação de vocabulários controlados, normas e padrões sobre vocabulários controlados.

5.1 Definições de vocabulário controlado

Existem diversas definições do termo “vocabulário controlado”, por exemplo: lista de termos autorizados (LANCASTER, 2004); subconjunto controlado de uma linguagem natural, desenvolvido para um fim ou para uma comunidade, composto por uma lista de termos explicitamente enumerados (NISO, 2005); lista de termos explicitamente enumerados com o objetivo de organizar e representar informação para facilitar a recuperação da informação (MAI, 2008); arranjo organizado de palavras e frases para representar conceitos (HARPRING, 2010); lista organizada de palavras e frases, ou sistema de notação, para rotular conteúdo e encontrá-lo via navegação ou busca (WARNER, 2002); lista padronizada de termos selecionados para uso consistente na descrição e indexação de recursos de informação (MILLER, 2011); linguagem de indexação em que terminologia é controlada (LANCASTER, 1986); conjunto de limitações a valores de elementos de metadados, ou conjunto predefinido de valores de elementos de metadados (CAPLAN, 2003). Nesta tese, o termo tem significado abrangente, e designa uma classe de sistema de organização de conhecimento ou uma classe de estrutura para organização de conhecimento (ABBAS, 2010; HODGE, 2000; NISO, 2005).

5.2 Propósitos e usos de vocabulários controlados

Existem diversos propósitos e usos de vocabulários controlados, por exemplo: controle de formas de termos, controle de admissão de nomes próprios, controle de formas de nomes próprios, controle de restrições a significados de termos, controle de sinônimos, controle de termos equivalentes, controle de relações hierárquicas, controle de relações associativas, controle de atualizações, promoção de consistência em processo de indexação de recursos de informação, promoção de consistência entre indexadores e usuários de recursos de informação, tradução entre linguagem natural e linguagem usada

em processo de indexação, suporte a processo de recuperação da informação, suporte a processo de construção de sistema de navegação, fonte de termos de busca, indicação de termos autorizados para uso, distinção entre homógrafos, redução de dispersão de recursos de informação relacionados, garantia de que cada termo tem só um significado e de que cada conceito é representado por apenas um termo preferido, uniformização de formatos de termos, padronização de significados de termos, suporte a processo de organização da informação, indexação manual/intelectual, indexação automatizada por algoritmo, indexação por rotulagem (*tagging*), classificação de coleção de recursos de informação, pesquisa facetada, pesquisa multilíngue, sugestão de recursos de informação em consulta, correção de erro de ortografia em consulta, complementação de termo em consulta, expansão e refino de consulta, suporte a processo de navegação visual, suporte a agrupamento e posicionamento (*ranking*) na representação visual de resultado de pesquisa, integração de fontes, publicação e reuso, suporte à comunicação entre pessoas, suporte a diferentes pontos de vista, suporte à interoperabilidade entre sistemas, suporte à engenharia de sistemas em processo de identificação e especificação de requisitos, promoção da confiabilidade e promoção do reuso (ABBAS, 2010; AITCHISON; BAWDEN; GILCHRIST, 2000; HARPRING, 2010; HEDDEN, 2010; KLESS et al., 2015; LANCASTER, 1986; MADER; HASLHOFER, 2013; NISO, 2005; PRIETO-DIAZ, 1990; SVENONIUS, 2003; TAYLOR; JOUDREY, 2009; USCHOLD; GRUNINGER, 1996; WARNER, 2002).

5.3 Controle de vocabulário na recuperação da informação

O controle de vocabulário é justificado por dificuldades no uso de linguagem natural para recuperar informação em grandes coleções, como nos casos de sinonímia, homografia e polissemia (GLUSHKO, 2013; NISO, 2005; WARNER, 2002). Na recuperação da informação, Kowalski e Maybury (2002) observam que no caso de indexação manual, o controle de vocabulário tende a tornar mais lento o processo de indexação, mas pode simplificar o processo de busca. Também observam que o uso de vocabulários controlados contribui para que os usuários conheçam termos no domínio e termos que melhor descrevem os recursos de informação. Svenonius (2000) relaciona dificuldades no uso de linguagem natural na organização da informação e observa a importância do controle de vocabulário. Finalmente, Svenonius (2003) destaca que a presença de sinônimos e homônimos pode causar falhas na comunicação entre usuários e o sistema de recuperação da informação, resultando na recuperação de pouca informação relevante ou de informação irrelevante.

Aitchison, Bawden e Gilchrist (2000) destacam as seguintes vantagens do uso de linguagem natural em comparação ao controle de vocabulário no processo de recuperação da informação: especificidade, atualidade, mesmos termos usados por autores e usuários. Entre as vantagens do controle de vocabulário, relacionam as seguintes: minimização de esforço na busca por informação, minimização de problemas sintáticos no uso de

termos compostos e minimização da recuperação da informação de interesse periférico. Fidel (1992) reporta estudo que comprova a importância de vocabulários controlados na busca por informação, particularmente no controle de sinônimos. Losee (2015) considera que o controle de vocabulário frequentemente tem impacto positivo na recuperação da informação, particularmente quando é desejável um elevado nível de revocação. Segundo Binkley e Lawrie (2010), é antigo o debate sobre vantagens e desvantagens do controle de vocabulário na recuperação da informação. Esses autores relacionam falsos positivos e falsos negativos como desvantagens do uso de texto livre (*free text retrieval*), e custos de desenvolvimento e manutenção dos vocabulários e da indexação dos recursos de informação, como desvantagens do controle de vocabulário. Na opinião de Binkley e Lawrie (2010), em certos contextos, os potenciais benefícios do controle de vocabulário não justificam os custos dessa abordagem. Segundo essa fonte, a exposição dos usuários a ferramentas que possibilitam o uso de texto livre tende a levar os usuários a terem preferência pelo uso de texto livre.

5.4 Termos em vocabulários controlados

Termos estão entre os elementos que compõem um vocabulário controlado. Um termo é uma sequência de caracteres em linguagem natural ou artificial. Os termos designam conceitos. Cada conceito pode ser designado por mais de um termo e cada conceito independe dos termos usados para designá-lo (AITCHISON; BAWDEN; GILCHRIST, 2000; DODEBEI, 2002; GARSHOL, 2004; GÖDERT; HUBRICH; NAGELSCHMIDT, 2014; GOMES, 1990). Em um vocabulário controlado, se um termo designa conceitos diferentes, ele deve ser qualificado para eliminar ambiguidade, e se mais de um termo é usado para representar um conceito, um dos termos deve ser identificado como preferido (NISO, 2005). Em vocabulários controlados, termo é rótulo para conceito. Em vocabulários controlados diferentes, um mesmo termo pode designar conceitos diferentes (CHU, 2010; HEDDEN, 2010; NISO, 2005). A documentação do vocabulário controlado pode englobar definições de conceitos, definições de regras para uso do vocabulário controlado, e definições de regras para combinar termos (GLUSHKO, 2013). Termos em vocabulários controlados devem ser definidos com o objetivo de promover consistência quando do uso dos mesmos, as definições proveem contexto necessário ao entendimento (DLESE, 2005). Conceitos podem ser definidos explicitamente ou implicitamente. Um conceito pode ser definido implicitamente por sinônimos ou outras relações. A definição deve enfatizar utilidade no domínio.

A utilidade de um vocabulário controlado depende de congruência com o vocabulário dos usuários. Quanto maior a audiência, menor tende a ser o consenso sobre os termos e maior a manutenção decorrente de mudanças na terminologia. No desenvolvimento de vocabulários controlados, diferentes interessados (*stakeholder*) devem ser considerados.

Por exemplo, responsáveis pela manutenção do vocabulário controlado, especialistas em indexação ou no uso do vocabulário controlado, e usuários não especialistas no vocabulário controlado. É também relevante levar em consideração a interoperabilidade entre vocabulários controlados (CAPLAN, 2003; NISO, 2005). A seguir, são listados critérios para escolha de termos no desenvolvimento de vocabulário controlado: garantia (endosso) literária (*literary warrant*), garantia de usuário (*user warrant*), garantia organizacional (*organizational warrant*) e garantia estrutural (*structural warrant*). A garantia literária justifica a escolha pelo uso frequente do termo na literatura, o vocabulário deve ser derivado da literatura que pretende descrever, um termo é justificável apenas se é sabido que ele ocorre na literatura no campo do assunto. A garantia de usuário justifica a escolha pelo uso frequente do termo em solicitações de informação sobre o conceito ou em consultas de usuários, um termo é justificável apenas se é de interesse dos usuários do serviço de informação. A garantia organizacional justifica a escolha do termo pelas características e contexto da organização. A garantia estrutural justifica a escolha do termo por ter uma função estrutural, melhorar precisão, melhorar revocação e/ou facilitar navegação (NISO, 2005; CHU, 2010; LANCASTER, 1986; SVENONIUS, 2000; SVENONIUS, 2003). Sobre esses critérios, Svenonius (2000) observa que a garantia literária pode não ser suficiente, pois o vocabulário de quem cria o recurso pode ser distinto daquele de quem procura o recurso. Para Dodebei (2002), o ideal é a terminologia ter garantia literária e garantia do usuário. Giess, Wild e McMahon (2008) consideram que embora a garantia literária dite que o desenvolvimento ocorra por análise de literatura, essa análise requer entendimento das necessidades dos usuários. Lancaster (1986) sugere que o critério de garantia literária seja estendido da seguinte forma: um termo é justificável apenas se ocorre com frequência suficiente na literatura a ponto de ser considerado significativo e útil para propósitos de recuperação. Na identificação de termos, Lancaster (1986) sugere que sejam realizadas consultas a fontes em que terminologia seja apresentada de modo compacto, como dicionários, glossários, enciclopédias, livros texto abrangentes e resumos (*abstract*). Esse autor considera que a garantia de usuário é importante na definição do nível de especificidade do vocabulário e sugere que termos sejam coletados na literatura e também junto aos usuários.

5.5 Relações em vocabulários controlados

Diversos vocabulários controlados são mais do que listas de termos, eles apresentam estrutura representada por relações (LANCASTER, 2004). Em vocabulários controlados, relações entre termos devem ser recíprocas. Para cada relação entre termo A e termo B, deve haver relação correspondente entre termo B e termo A (NISO, 2005). As relações dependem da natureza e dos objetivos do vocabulário. Algumas delas são específicas a determinados domínios (GÖDERT; HUBRICH; NAGELSCHMIDT, 2014; SVENONIUS,

2000). A fonte ISO (2000) descreve exemplos de relações: hierárquica, genérica, partitiva, associativa, sequencial, temporal e causal. A relação hierárquica pode ser genérica ou partitiva. Na genérica, a intensão de um dos conceitos inclui a do outro conceito e pelo menos uma característica delimitadora adicional. Na partitiva, um dos conceitos é o todo e o outro conceito é parte desse todo, a relação associativa, por sua vez, é uma relação não hierárquica que decorre de experiência, enquanto a relação sequencial é uma relação associativa decorrente de proximidade temporal ou espacial. A relação temporal é uma relação sequencial englobando eventos no tempo, e a relação causal é uma relação associativa que engloba causa e efeito. Por fim, relação em vocabulário controlado também pode classificada como relação de equivalência, relação hierárquica ou relação de associação.

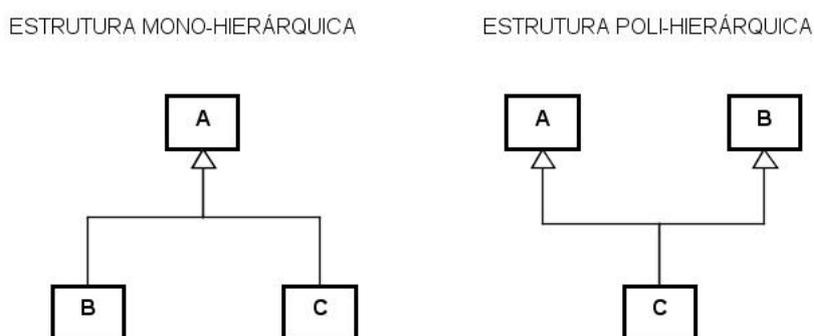
5.5.1 Relação de equivalência

A relação de equivalência ocorre quando um conceito pode ser designado por mais de um termo, quando mais de um termo tem um mesmo significado. Ela ocorre entre termos que designam um mesmo conceito e contribui no controle de dispersão léxica, simbólica e sintática (DODEBEI, 2002; GÖDERT; HUBRICH; NAGELSCHMIDT, 2014; NISO, 2005; ROWLEY; HARTLEY, 2008). Em um sistema de armazenamento e recuperação da informação, as relações de equivalência podem indicar quais termos são preferidos ou autorizados (ABBAS, 2010). Os termos não precisam ser sinônimos, basta que os conceitos designados sejam similares ao ponto de se considerar que os termos designam o mesmo conceito (HEDDEN, 2010). O controle de sinônimos e quase sinônimos (*quasi-synonym*) pode minimizar a disseminação de sinônimos e quase sinônimos em bases de dados e ter impacto positivo na revocação (AITCHISON; BAWDEN; GILCHRIST, 2000). Nesse contexto, quase sinônimo é um termo cujo significado não é exatamente sinônimo do significado de outro termo, mas que pode ser tratado como equivalente no contexto de um vocabulário controlado (NISO, 2005). Em uma relação de equivalência, os termos são membros de um conjunto de equivalência. Na recuperação da informação, o uso de um termo integrante de um conjunto de equivalência recupera todos os recursos que seriam recuperados usando-se todos os outros termos desse conjunto. Escolher um termo preferido e para ele mapear os outros termos é uma técnica para o estabelecimento de relação de equivalência. O termo preferido pode ser escolhido, por exemplo, por meio do princípio do uso comum (SVENONIUS, 2000). Sinônimo, variação ortográfica, acrônimo, termo científico, termo técnico e termo em idioma estrangeiro são classes de termos não preferidos citados em Hedden (2010). Finalmente, é importante ressaltar que as relações de equivalência podem ser úteis no controle de sinônimos, homônimos, erros, mudanças de grafia, abreviações e termos usados em substituição a outros termos (LEISE; FAST; STECKEL, 2002).

5.5.2 Relação hierárquica

Na recuperação da informação, as relações hierárquicas contribuem na solução de problemas que surgem quando os conceitos têm diferentes níveis de especificidade (SVENONIUS, 2000). Essas relações possibilitam aumento da revocação por meio da ampliação de consultas (introdução de termos) ou aumento da precisão por meio do uso de termos mais específicos (AITCHISON; BAWDEN; GILCHRIST, 2000). Uma estrutura hierárquica pode ser poli-hierárquica ou mono-hierárquica. Na estrutura poli-hierárquica, existe termo com mais de um termo imediatamente acima na hierarquia (*parent*), enquanto que, na estrutura mono-hierárquica, cada termo tem um único termo imediatamente acima na hierarquia (GÖDERT; HUBRICH; NAGELSCHMIDT, 2014), como mostra a figura 14.

Figura 14 – Estruturas hierárquicas



Fonte: Elaborado pelo autor

Em um vocabulário controlado, a relação hierárquica pode ser genérica, instância ou todo-parte (NISO, 2005). A relação hierárquica clássica é a relação genérica. Essa relação reflete generalidades e especificidades relativas, ocorre entre gênero e espécie e indica a subordinação entre conceitos. Há relação entre gênero e espécie quando os conceitos têm características idênticas e um deles tem característica a mais do que o outro (DODEBEI, 2002; HEDDEN, 2010; SVENONIUS, 2000). Na relação hierárquica genérica, há relação entre conceito mais abrangente e conceito mais específico. O conceito mais específico possui uma ou mais características, além das características do conceito mais abrangente. Na relação instância, há relação entre entidade e instâncias dessa entidade. Finalmente, na relação todo-parte, há relação entre todo e partes que pertencem ao todo. Essa relação ocorre, por exemplo, nas seguintes situações: entre sistema e órgãos, entre localidades geográficas, entre disciplina e campos de discurso (GÖDERT; HUBRICH; NAGELSCHMIDT, 2014).

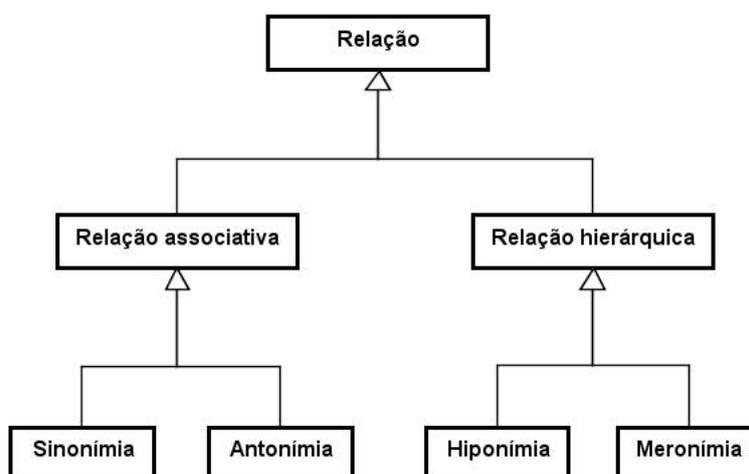
5.5.3 Relação de associação

Uma relação de associação é uma relação que deve ser representada por sugerir termos na indexação ou na recuperação da informação (NISO, 2005). Em uma relação de associação, os termos designam conceitos com significados que são sobrepostos ou que são semanticamente relacionados (GÖDERT; HUBRICH; NAGELSCHMIDT, 2014). Hedden (2010) e Svenonius (2000) alertam sobre inconsistências que podem decorrer de subjetividade ou falta de rigor no estabelecimento de relação de associação. Segundo Gödert, Hubrich e Nagelschmidt (2014), a relação de associação é inadequada à recuperação da informação por processo automatizado e seu entendimento depende de interpretação humana. Finalmente, Tudhope, Alani e Jones (2006) identificam variações da relação de associação. Por exemplo, as seguintes: sobreposição de conceitos, antônimo e dependência.

5.5.4 Relação semântica

Em sentido estrito, o termo “relação semântica” designa relação entre conceitos, enquanto o termo “relação léxica” designa relação entre palavras. Em sentido amplo, o termo “relação semântica” designa relação entre conceitos ou relação entre palavras (HJØRLAND, 2007b). A relação semântica pode ser paradigmática ou sintagmática. A relação paradigmática é uma relação permanente, é uma relação verdadeira *a priori* ou por definição. Por sua vez, a relação sintagmática é uma relação contingencial, é verdadeira em certos contextos (LANCASTER, 1986; STOCK, 2010; SVENONIUS, 2003). Stock (2010) considera sinonímia, antonímia, hiponímia e meronímia como exemplos de relações semânticas. A figura 15 ilustra hierarquia de classes de relações anteriormente referenciadas.

Figura 15 – Hierarquia de classes de relações



Fonte: Adaptado de Stock (2010)

A relação denominada sinonímia é uma relação onde similaridades semânticas são mais salientes do que diferenças. Os termos A e B são sinônimos se representam o mesmo conceito. Dois termos são absolutamente sinônimos se têm o mesmo significado denotativo, conotativo e social. Um termo pode substituir o outro em todos os contextos de ocorrência. O significado denotativo indica qualidades essenciais de um conceito, qualidades que o distinguem de outros conceitos, enquanto o significado conotativo é significado associado ao conceito (as associações são feitas ao conceito quando ele é referenciado). O significado social, por sua vez, é o significado decorrente do uso em determinadas situações sociais e circunstâncias. Antônimos são termos diferentes em forma e significado, transmitem sentidos opostos. Hiponímia e hiperonímia são relações entre significado mais genérico e significado mais específico. O significado mais específico inclui o mais genérico. O termo A é hiperônimo do termo B se o conceito representado pelo termo A é mais geral do que o representado pelo termo B. Se o termo A é hiperônimo do termo B, então o termo B é hipônimo do termo A. Finalmente, meronímia é uma relação de inclusão, é uma relação entre todo e parte. O termo A é merônimo do termo B se o conceito representado pelo termo A é parte do conceito representado pelo termo B. Se o termo A é merônimo do termo B, diz-se que o termo B é holônimo do termo A (CRUSE, 2011; MANGOLD, 2007; MICHEL, 1997; SYAL, 2010).

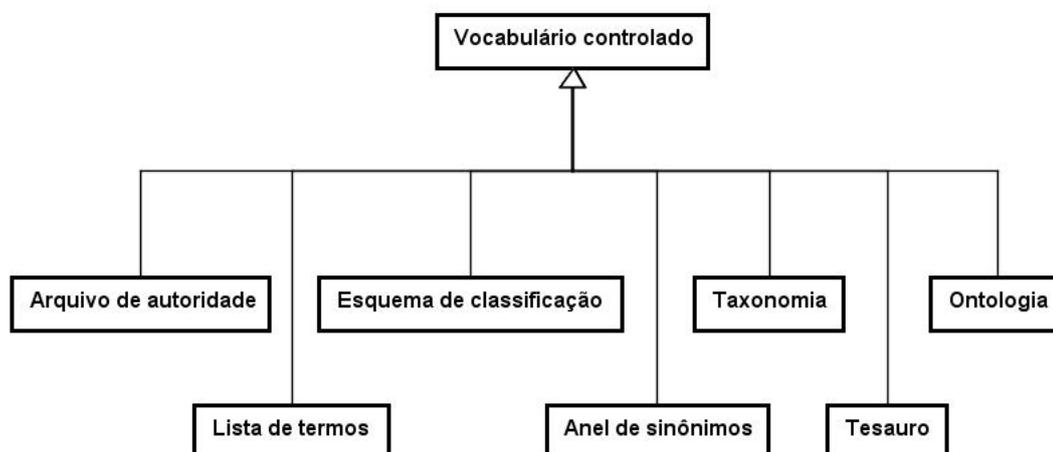
5.6 Estruturas de vocabulários controlados

Embora um vocabulário controlado possa apenas ser um conjunto de termos, frequentemente são impostas estruturas para aproximar termos que designam conceitos relacionados. Arquivo de autoridade, lista de termos, esquema de classificação, anel de sinônimos, taxonomia, tesouro e ontologia são classes de vocabulários controlados (CAPLAN, 2003; DLESE, 2005; KLESS et al., 2015; LANCASTER, 1986; LANCASTER, 2004; NISO, 2005). A figura 16 ilustra hierarquia de classes de vocabulários controlados anteriormente referenciadas.

5.6.1 Arquivo de autoridade

Um arquivo de autoridade é uma compilação de termos autorizados usados por uma organização. Pode ser estruturado como lista de termos preferidos ou de valores aceitos, ou como anel de sinônimos em que os termos foram escolhidos como preferidos ou aceitos (MORVILLE; ROSENFELD, 2006). Em um arquivo de autoridade, não há relações entre termos (CAPLAN, 2003).

Figura 16 – Hierarquia de classes de vocabulários controlados



Fonte: Elaborado pelo autor

5.6.2 Lista de termos

A lista de termos é uma estrutura simples de vocabulário controlado, geralmente usada quando a quantidade de termos no vocabulário é pequena. São características desejáveis em listas de termos: cada termo é único, termos são membros de classe em sistema de classificação, termos têm mesma granularidade ou especificidade, e termos são organizados em alguma ordem lógica. Em uma lista de termos, qualificadores podem ser usados para desambiguar termos idênticos (GARSHOL, 2004; HARPRING, 2010; MILLER, 2011).

5.6.3 Esquema de classificação

O termo classificação é frequentemente usado como abreviação do termo “sistema de classificação” ou do termo “esquema de classificação”, geralmente usados como sinônimos. Esses termos designam lista de classes organizada de acordo com princípios estabelecidos e construída com o propósito de organizar recursos de informação em coleção, entradas em índice, bibliografia ou catálogo, agrupadas com base em similaridades e diferenças para facilitar acesso e recuperação (REITZ, 2012). Os esquemas de classificação capturam distinções e relações entre recursos de informação, possibilitam organizar informação segundo princípios pré-estabelecidos e caracterizam-se por sistemas de notação e por estruturas de relações. Os esquemas de classificação possibilitam agrupar recursos de informação de mesma classe. Podem ser monodimensionais ou multidimensionais. No esquema de classificação monodimensional, a classificação ocorre segundo um aspecto, enquanto no multidimensional, a classificação ocorre segundo mais de um aspecto (ABBAS, 2010; CHU, 2010; GIESS; WILD; MCMAHON, 2008; GLUSHKO, 2013; HJØRLAND,

2006; KWASNIK, 1992; NISO, 2005). Um esquema de classificação geralmente engloba notação usada para simbolizar classes e divisões (REITZ, 2012) e pode ser voltado a um ou a mais domínios.

Kwasnik (1999) destaca que cada processo de classificação da informação tem diferentes objetivos e que cada esquema de classificação da informação tem diferentes propriedades estruturais, apresenta diferentes aspectos positivos e negativos. No processo de desenvolvimento de esquema de classificação, Mai (2004) destaca a importância das seguintes atividades: realizar estudos de discurso e de atividades na organização ou no domínio; aprender a linguagem da comunidade; e estudar as interações de usuários com a informação, hábitos de trabalho e estrutura do domínio. Também destaca que o desenvolvimento de um esquema de classificação não deve ser embasado apenas em diretrizes padronizadas. Segundo Goh et al. (2009), o processo de desenvolvimento de esquema de classificação pode seguir abordagem descendente (*top-down*) ou ascendente (*bottom-up*). Na abordagem descendente, o escopo do domínio é considerado e dividido levando-se em consideração determinadas características, enquanto na abordagem ascendente, a visão do domínio é obtida a partir de conceitos nos recursos de informação. Esses conceitos definem o escopo do domínio e o esquema de classificação é construído por agrupamento desses conceitos.

Alguns autores distinguem entre “esquema de categorização” e “esquema de classificação”. Segundo Jacob (2004), são características de esquema de categorização: composto por categorias que podem estar organizadas hierarquicamente; a participação de uma entidade em uma categoria não proíbe participação em outra categoria; a participação em categorias pode variar ao longo do tempo em decorrência de mudanças de contexto; os critérios para atribuir entidades a categorias podem variar; pode ser usada informação dependente e independente de contexto para definir categorias; as definições de categorias podem mudar em resposta a variações no ambiente; determinados membros podem ser mais típicos de uma categoria do que outros; as relações entre categorias são modificáveis. São características de esquema de classificação: composto por classes mutuamente exclusivas, não sobrepostas e arrumadas em estrutura hierárquica; reflete ordem predeterminada da realidade; cada entidade pode ser membro de uma só classe e cada classe tem rótulo que possibilita associar membros à classe; os membros de uma classe devem apresentar as características da classe e todos os membros da classe são igualmente representativos dessa classe.

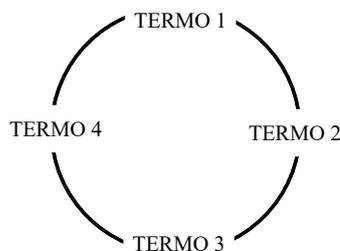
Beghtol (1986) destaca a importância do uso de elementos teóricos e práticos para garantir que um esquema de classificação proveja organização que faça sentido. Ressalta que a ausência desses elementos pode resultar em mistura arbitrária de conceitos, e que definir princípios e prioridades promove o estabelecimento de relações significativas entre elementos no esquema de classificação, assim como entre o esquema de classificação e

os recursos que procura organizar. Além disso, afirma que esses princípios beneficiam os usuários. Entre esses princípios, se encontra a garantia semântica (*semantic warrant*) que embasa o esquema de classificação. Essa garantia é a autoridade invocada quando é preciso justificar ou verificar decisões acerca, por exemplo, de quais classes e conceitos incluir no esquema. Beghtol (1986) relaciona as seguintes garantias semânticas na construção de esquemas de classificação: literária, científica/filosófica, educacional e cultural. Finalmente, também observa que a garantia literária pode ser implementada com base na terminologia de um campo, por exemplo.

5.6.4 Anel de sinônimos

Um anel de sinônimos é um conjunto de sinônimos sem termos preferidos. Nessa estrutura, o controle é estabelecido definindo-se quais termos são equivalentes na recuperação da informação (sinônimos e quase sinônimos). Esse controle minimiza variações introduzidas pela linguagem natural (HEDDEN, 2010; WARNER, 2002). A estrutura é um anel, pois nenhum dos termos é preferido, todos são equivalentes (MILLER, 2011). Em certos contextos, o termo “*synset*” é usado para designar anel de sinônimos. Um anel de sinônimos pode ser usado, por exemplo, na expansão de consulta. Se um anel de sinônimo é usado, quando o usuário pesquisa por termo, são recuperados recursos com qualquer termo no anel (HARPRING, 2010; MILLER, 2011). Na recuperação da informação, anéis de sinônimos podem aumentar revocação, mas reduzir precisão (MORVILLE; ROSENFELD, 2006). Finalmente, podem ser criados conjuntos de sinônimos em que um termo seja escolhido como representação principal do conceito. Esse termo é denominado preferido e os outros são não preferidos. O termo preferido é preferencialmente usado como descritor de recurso de informação na indexação, enquanto os não preferidos são usados como pontos de acesso adicionais (GÖDERT; HUBRICH; NAGELSCHMIDT, 2014). A figura 17 ilustra a estrutura de um anel de sinônimos.

Figura 17 – Estrutura de anel de sinônimos

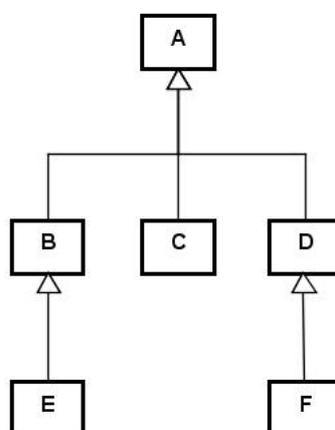


Fonte: Elaborado pelo autor

5.6.5 Taxonomia

O termo taxonomia se originou nas palavras gregas *taxis* (arranjo ou ordem) e *nomos* (lei ou ciência) e designa diversos conceitos (GILCHRIST, 2003; HEDDEN, 2010), por exemplo, meio de organizar conceitos (HEDDEN, 2010), estrutura hierárquica em que relações são limitadas a relações “é um tipo de” (*is a kind of*) (JAKUS et al., 2013), vocabulário controlado composto por termos preferidos conectados em hierarquia ou polihierarquia (NISO, 2005) e classificação baseada em assunto hierarquicamente organizada (GARSHOL, 2004). A figura 18 ilustra possíveis relações em uma estrutura de taxonomia.

Figura 18 – Exemplo de estrutura de taxonomia



Fonte: Elaborado pelo autor

As taxonomias focam relações entre termos em áreas de conhecimento relevantes a comunidades específicas, possibilitam controlar ambiguidades e sinônimos e podem ser usadas para estruturação de diretórios; indexação, classificação e categorização da informação; formulação e filtragem de consultas; e construção de sítios (GILCHRIST, 2003; NISO, 2005; ROWLEY; HARTLEY, 2008). Em uma taxonomia, pode existir uma hierarquia ou um conjunto de hierarquias. Taxonomias podem ser facetadas. Essas taxonomias geralmente são compostas por conjuntos de hierarquias, em que cada hierarquia descreve o domínio a partir de um ponto de vista. Preferencialmente, esses pontos de vista são ortogonais, de forma que cada termo não surja em mais de uma faceta (HEDDEN, 2010; TZITZIKAS, 2007).

5.6.6 Tesouro

O termo tesouro se originou no dicionário analógico *Thesaurus of English words and phrases* de Peter Mark Roget publicado em 1852 (GOMES, 1990). No contexto da recuperação da informação, esse termo foi originalmente usado por Peter Luhn da empresa

IBM. O primeiro tesouro para controle de vocabulário na recuperação da informação provavelmente foi o desenvolvido pela empresa DuPont em 1959. Os primeiros tesouros amplamente disponíveis foram o *Thesaurus of Armed Services Technical Information Agency (ASTIA) Descriptors* e o *Chemical Engineering Thesaurus* (LANCASTER, 1986; SHIRI, 2012). Atualmente, há tesouros para diversos domínios, por exemplo, o *IEEE Thesaurus*, um tesouro composto por termos técnicos e científicos usados em ciências e engenharias (IEEE, 2014a). No quadro 5, são apresentados nomes de alguns tesouros e nomes de domínios para os quais foram desenvolvidos. A título de exemplo, a figura 19 apresenta a resposta de um tesouro a uma consulta.

Quadro 5 – Relação de tesouros em domínios

NOME	DOMÍNIO
AGROVOC	Agricultura
Art & Architecture Thesaurus	Arte, arquitetura e cultura
ASIS Thesaurus of Information Science and Librarianship	Ciência da Informação e Biblioteconomia
ERIC Thesaurus	Educação
Eurovoc	Pluridisciplinar
IEEE Thesaurus	Engenharia, técnica e ciência
NASA Thesaurus	Engenharia e ciência
NCI Thesaurus (NCIt)	Medicina
STW Thesaurus for Economics	Economia
TCI – Tesouro em Ciência da Informação	Ciência da Informação
The USAID Thesaurus	Pluridisciplinar
UNESCO Thesaurus	Educação, cultura, ciências naturais, ciência social e humana, comunicação e informação

Fonte: Elaborado pelo autor

Um tesouro é um vocabulário controlado estruturado em que podem existir relações de equivalência, relações hierárquicas e relações associativas. A figura 20 ilustra possíveis relações em uma estrutura de tesouro. Tesouros podem ser usados para melhorar a recuperação da informação (MORVILLE; ROSENFELD, 2006). Para Aitchison, Bawden e Gilchrist (2000), o propósito primário do uso de tesouros é recuperar informação e um propósito secundário é entender áreas de conhecimento. Segundo Gomes (1990), os tesouros são componentes de sistemas de recuperação da informação, a função primária de um tesouro é representar conceitos em recursos e em solicitações de busca. Um tesouro geralmente aborda um domínio de conhecimento. E embora não exista um tesouro considerado geral, podem existir tesouros sobre diferentes assuntos, desenvolvidos com o objetivo de serem compatíveis. Miller (1997) relaciona aspectos estruturais e métodos de construção como diferenças entre esquema de classificação e tesouro. Em um processo de indexação, o uso de tesouro pode prover maior nível de especificidade do que o uso de vocabulário controlado estruturado como taxonomia (DAVIES, 2010; HEDDEN, 2010;

Figura 19 – Resposta de um tesouro a uma consulta

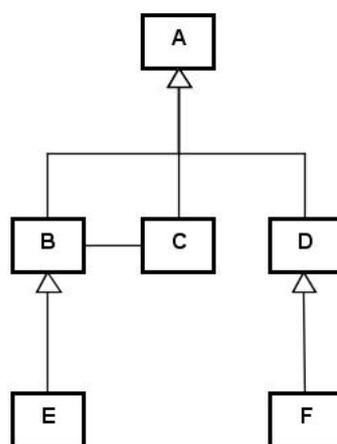
The screenshot shows the AGROVOC thesaurus interface. On the left is a hierarchical tree of categories. The main content area displays the results for the search term 'thesaurus'. The preferred term is 'thesaurus', defined as 'A book of synonyms, often including related and contrasting words and antonyms'. It lists broader concepts like 'taxonomy (information management)' and 'subject headings', and components like 'indexing of information'. It also provides translations in various languages including Arabic, Chinese, Czech, French, German, Italian, Korean, Portuguese, Russian, Spanish, Telugu, and Turkish. At the bottom, it lists closely matching and exact match concepts from other sources like dbpedia.org, aims.fao.org, and cat.aii.caas.cn.

Fonte: FAO (2015a)

REDMOND-NEAL; HLAVA, 2005). Entre os usos de tesouros, é possível listar os seguintes: entendimento de terminologia em área de conhecimento, fonte de termos para etiquetas ou rótulos usados na recuperação da informação, fonte de termos usados em anotações exploradas na recuperação da informação, indexação de recursos de informação, fonte de termos sugeridos em consultas, expansão automática de consultas, suporte à navegação, fonte de termos para metadados, suporte à construção de interfaces voltadas à busca facetada e suporte à busca exploratória (HEDDEN, 2010; REDMOND-NEAL; HLAVA, 2005; SHIRI, 2012). Aitchison, Bawden e Gilchrist (2000) observam que, no contexto da recuperação da informação, os tesouros podem ser usados tanto na indexação quanto na busca (*search*); na indexação, mas não na busca; na busca mas não na indexação; não ser usados na indexação e nem na busca. Segundo Aitchison, Bawden e Gilchrist (2000), o uso de tesouros na indexação e na busca é o uso clássico e predomina na prática. No processo de busca, o sistema de recuperação da informação geralmente usa o tesouro para sugerir termos aos usuários. Essa sugestão pode ocorrer por oferta de termos para escolha ou por meio da expansão de consultas. Considerando o uso, Morville e Rosenfeld (2006)

relacionam os seguintes tipos de tesouro: tesouro clássico (usado na indexação e busca), tesouro de indexação e tesouro de busca. Gomes (1990) classifica os tesouros em monolíngue, multilíngue, macrotesouro (termos representam conceitos amplos), microtesouro (termos representam conceitos específicos em área de conhecimento restrita), voltado a uma missão (podem ser multidisciplinares) e dedicado a um assunto. Segundo Lancaster (1986), os microtesouros podem ser desenvolvidos para integração a tesouros mais genéricos. Ainda segundo essa fonte, essa estratégia promove compatibilidade entre vocabulários, reduz a proliferação e a sobreposição de vocabulários similares.

Figura 20 – Exemplo de estrutura de tesouro



Fonte: Elaborado pelo autor

5.6.7 Ontologia

O termo ontologia se origina das palavras gregas *ontos* (ser ou o que existe) e *logos* (razão ou conhecimento) (ORG, 2007). Em Filosofia, esse termo pode designar teoria sobre a natureza da existência, dos tipos de coisas que existem, ou disciplina que estuda essas teorias (BERNERS-LEE; HENDLER; LASSILA, 2001). Ainda em Filosofia, ontologia pode também designar teoria do que existe, estudo dos tipos de entidades na realidade e das relações entre essas entidades (ARP; SMITH; SPEAR, 2015). O termo também designa ramo da Filosofia que visa prover classificação de entidades e relações em todas as esferas da realidade, tanto material quanto abstrata (DAVIES, 2010; ORG, 2007). Em outros contextos, há também definições do termo ontologia, como as seguintes: especificação explícita de uma conceituação, onde conceituação é visão simplificada e abstrata do mundo que se deseja representar para determinado propósito (GRUBER, 1993); representação formal de conceitos em um domínio de discurso e de relações entre esses conceitos (REIMER, 2011b); modelo de domínio de interesse descrito em uma linguagem lógica (DAVIES, 2010); definição formal de conceituação compartilhada, onde

conceituação é interpretação estruturada de parte do mundo, usada por pessoas para pensar e comunicar sobre o mundo (BORST, 1997); descrição ou representação explícita de parte de uma conceituação, onde conceituação é visão do mundo, forma de pensar sobre um domínio, geralmente expressa por conjunto de conceitos, suas definições e suas inter-relações (USCHOLD, 1996).

5.6.7.1 Elementos de ontologias

Como vocabulário controlado, os componentes de ontologias são conceitos, classes de conceitos, relações entre classes, indivíduos, instâncias que representam indivíduos, atributos de conceitos, instâncias e valores de atributos (CALERO; RUIZ; PIATTIN, 2006; JAKUS et al., 2013). Em ontologias, as relações entre conceitos têm significados e podem ser específicas a domínios (HEDDEN, 2010). As definições de conceitos e as relações entre conceitos impõem estrutura ao domínio e restringem interpretações (USCHOLD, 1998). Arp, Smith e Spear (2015) destacam a importância das definições de conceitos no uso consistente de uma ontologia, no suporte a inferências computacionais e na restrição à organização da ontologia. Finalmente, Arp, Smith e Spear (2015) também destacam que, em uma ontologia, deve haver definição para cada termo, exceto para aqueles muito genéricos.

5.6.7.2 Propósitos e usos de ontologias

Existem motivações, para desenvolvimento e uso de vocabulários controlados com relações semânticas mais expressivas do que as encontradas nas estruturas de vocabulários controlados anteriormente descritas. Nesse contexto, são relevantes vocabulários controlados frequentemente classificados como ontologias. Existem diversos propósitos e usos de ontologias, por exemplo, os seguintes: melhoria da interação de usuários com sistemas de organização do conhecimento, particularmente na formulação de consultas e na navegação; melhoria do processo de aprendizado dos usuários sobre domínios; suporte à expansão de consultas; suporte a processos executados por indexadores humanos e por sistemas automatizados de indexação; suporte a aplicações na *Web* semântica; entendimento, integração, estruturação e recuperação da informação; promoção de consistência na descrição da informação; aquisição, análise, estruturação, unificação, compartilhamento, comunicação e reuso de conhecimento; definição de relações entre conceitos; definição de metadados; redução de ambiguidades e incompatibilidades; processamento de linguagem natural; desenho (*design*) e integração de bases de dados; educação; modelagem organizacional; descrição, categorização e indexação de recursos; explicitação de suposições sobre domínio; separação de conhecimento operacional de conhecimento sobre domínio; melhoria da comunicação entre pessoas ou entre sistemas de software; promoção de interoperabilidade entre sistemas de software; facilidade de acesso a fontes de informação; entendimento e mapeamento de termos; identificação, especificação e documentação de

software; e promoção de comportamento uniforme e previsível de software (ARP; SMITH; SPEAR, 2015; DOERR, 2008; GARZAS; PIATTINI, 2005; GIUNCHIGLIA; DUTTA; MALTESE, 2009; GÓMEZ-PÉREZ; CORCHO; FERNÁNDEZ-LÓPEZ, 2004; ROWLEY; HARTLEY, 2008; RUIZ; HILERA, 2006; NOY; MCGUINNESS, 2001; SICILIA, 2006; SIMMONS; DILLON, 2006; SOERGEL et al., 2004; USCHOLD; JASPER, 1999).

5.6.7.3 Classes de ontologias

As ontologias podem variar, por exemplo, em formalismo, assunto e propósito (CORCHO; POVEDA-VILLALÓN; GÓMEZ-PÉREZ, 2015; GÓMEZ-PÉREZ; CORCHO; FERNÁNDEZ-LÓPEZ, 2004; USCHOLD, 1996). Existem diversas classificações propostas para ontologias. Por exemplo, classificações quanto ao nível de formalismo. Em ontologias, diferentes níveis de formalismo podem ser adotados (BORST, 1997). Quanto ao nível de formalismo, as ontologias podem ser classificadas em: altamente informal, expressa informalmente em linguagem natural; informal estruturado, expressa em forma restrita e estruturada de linguagem natural; semiformal, expressa em uma linguagem artificial formalmente definida; e rigorosamente formal, definida por termos meticulosamente definidos com semântica formal, teoremas e provas de propriedades como robustez (*soundness*) e completude (*completeness*) (USCHOLD, 1996). Giunchiglia, Dutta e Maltese (2009) classificam as ontologias em nível crescente de formalismo: classificações de usuários, esquemas de classificação enumerados, tesouros, esquemas de classificação facetados e ontologias descritas na *Web Ontology Language* (OWL). Considerando assunto, Uschold (1996) relaciona as seguintes classes de ontologia: ontologia de domínio; ontologia de problema, método ou tarefa; ontologia de representação ou meta-ontologia. Guarino (1998) relaciona as seguintes classes, considerando o nível de dependência em relação a tarefa ou a ponto de vista: ontologia de topo (*top-level ontology*), ontologia de domínio, ontologia de tarefa e ontologia de aplicação. As ontologias de topo descrevem conceitos genéricos que independem de problema ou de domínio particular. As ontologias de domínio descrevem conceitos em domínios específicos. As ontologias de tarefa descrevem conceitos relacionados a tarefas genéricas. As ontologias de aplicação descrevem conceitos em domínios e tarefas particulares. A título de ilustração, no quadro 6, são apresentados nomes de algumas ontologias geralmente classificadas como ontologias de topo.

Considerando níveis de usabilidade e reusabilidade, Corcho, Poveda-Villalón e Gómez-Pérez (2015) sugerem as seguintes classes: ontologia de nível superior, ontologia de domínio e ontologia focada em aplicação. Ontologias de nível superior abordam conceitos aplicáveis a diferentes domínios, enquanto as ontologias de domínio abordam domínios específicos e são reusáveis nesses domínios. As ontologias focadas em aplicações abordam aplicações específicas em domínios específicos. Considerando generalidade do assunto, Arp, Smith e Spear (2015) distinguem entre ontologia de topo e ontologia de domínio. Considerando propósito da ontologia, esses autores distinguem entre ontologia de aplicação

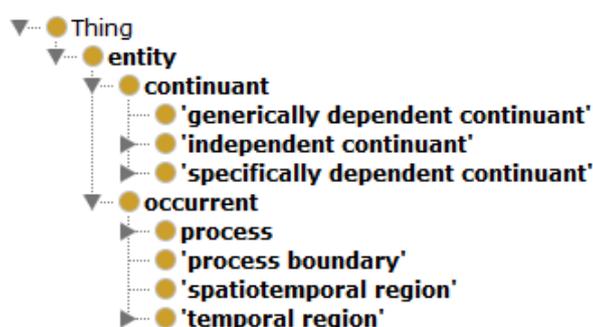
Quadro 6 – Nomes de ontologias de topo

SIGLA	NOME
BFO	Basic Formal Ontology
DOLCE	Descriptive Ontology for Linguistic and Cognitive Engineering
SUMO	Standard Upper Merged Ontology
UFO	Unified Foundational Ontology
WordNet	WordNet

Fonte: Elaborado pelo autor

e ontologia de referência. Uma ontologia de topo é uma representação de alto nível comum a diversos domínios, enquanto uma de domínio é uma representação de domínio. Ainda segundo [Arp, Smith e Spear \(2015\)](#), ontologia de aplicação é desenvolvida para aplicação ou tarefa específica, enquanto ontologia de referência é representação abrangente de entidades em domínio. Trechos de ontologias de referência podem ser usados no desenvolvimento de ontologias de aplicação. A título de ilustração, a figura 21 apresenta trecho da *Basic Formal Ontology* (BFO), uma ontologia classificada por [Arp, Smith e Spear \(2015\)](#) como ontologia de topo, enquanto a figura 22 apresenta trecho de resposta de consulta à *Protein Ontology* (PRO), ontologia classificada por [Arp, Smith e Spear \(2015\)](#) como uma ontologia de referência.

Figura 21 – Trecho da *Basic Formal Ontology*



Fonte: Elaborado pelo autor

[Corcho, Poveda-Villalón e Gómez-Pérez \(2015\)](#) contrastam ontologia leve (*lightweight ontology*) e ontologia pesada (*heavyweight ontology*). Segundo esses autores, as diferenças entre elas são quantidade e características dos axiomas. Em ontologias leves, geralmente há definições de conceitos, definições de propriedades e taxonomias de conceitos, enquanto as ontologias pesadas podem conter outros tipos de restrições e axiomas. Esses autores

outros esquemas de classificação, conceitos são organizados em estruturas sem semântica formal, e que servem como mapas de domínios e podem ser usadas na recuperação e organização da informação. A fonte [DLESE \(2005\)](#) considera que as seguintes características distinguem ontologias de tesouros: ontologias geralmente não relacionam termos preferidos; em ontologias, conceitos e relações são geralmente descritos em modo legível por máquina. Segundo [Soergel et al. \(2004\)](#), tesouros têm classes limitadas de relações e relações com semântica imprecisa. Segundo esses autores, essas limitações podem implicar em existência de relações que não refletem as relações estabelecidas por pessoas, definições pobres, limitações ao processamento automatizado, presença de ambiguidades, e estrutura semântica irregular e imprevisível. São características de ontologias, segundo esses autores: conceitos precisamente definidos; relações explicitamente nomeadas e especificadas por regras e por restrições; semântica precisa; possibilidade de processamento automatizado e realização de inferências a partir de conhecimento representado na ontologia. Para [Gödert, Hubrich e Nagelschmidt \(2014\)](#), na transição de tesouro para ontologia, a interpretação cognitiva é substituída por especificação formal que possibilita processamento por máquina.

[Rowley e Hartley \(2008\)](#) consideram que o uso dos termos taxonomia e ontologia em diversas disciplinas resultou em várias definições desses termos, sendo difícil propor definições que atendam a todos os significados. Ainda segundo essa fonte, esses termos são muitas vezes intercambiáveis. [Kless et al. \(2015\)](#) observam que comparações entre vocabulários controlados e ontologias realizadas por diferentes autores muitas vezes não apresentam distinções claras entre essas classes de vocabulários, e que existem opiniões divergentes quanto a similaridades, diferenças, funções e propósitos dessas classes de vocabulários. Para [Jimeno-Yepes et al. \(2009\)](#), os tesouros têm o propósito primário de organizar termos, enquanto as ontologias são especificações explícitas de conceituações. Ainda segundo essa mesma fonte, as ontologias têm propósitos mais específicos do que os tesouros, os consumidores de ontologias são aplicações computacionais e não seres humanos, e as ontologias não precisam relacionar variantes de termos. Para [Harpring \(2010\)](#), no contexto da Ciência da Computação, ontologias geralmente são especificações formais e legíveis por máquina de modelos conceituais, não são vocabulários controlados, mas os usam para domínios definidos. Esse autor também considera que, embora ontologias tenham características em comum com taxonomias e tesouros, ontologias usam estritas relações semânticas para representar conhecimento em formas legíveis por máquinas. Finalmente, considera que, enquanto taxonomias e tesouros são ferramentas para catalogação e recuperação da informação, ontologias são formas de representação de conhecimento em domínios.

5.7 Desenvolvimento de vocabulário controlado

O desenvolvimento de vocabulário controlado pode adotar abordagem com participação de comitê, abordagem empírica ou combinar essas abordagens. Na abordagem com comitê, especialistas no domínio do vocabulário controlado, apoiados por especialistas em desenvolvimento de vocabulários controlados, relacionam termos e estabelecem relações. Pode ser adotado método descendente, no qual são inicialmente identificados termos mais genéricos, ou método ascendente, no qual são inicialmente identificados termos mais específicos. Na abordagem empírica, por sua vez, são revisadas decisões na medida em que é adquirida experiência. Nessa abordagem, pode ser adotado método dedutivo, método indutivo ou combinação desses métodos. No método dedutivo, são coletados termos. E em seguida, especialistas os revisam e estabelecem relações entre eles. Caso existam hierarquias, geralmente estas são construídas seguindo-se método descendente. No método indutivo, são estabelecidas relações na medida em que são coletados termos. Caso existam hierarquias, geralmente são construídas seguindo-se método ascendente (NISO, 2005). Quanto aos métodos, Lancaster (1986) considera que o método ascendente é mais confiável. Harpring (2010) sugere que, no início do desenvolvimento de vocabulário controlado, sejam definidos: aderência a padrões, método de desenvolvimento, plano de manutenção, estrutura desejada, tipos de relações e apresentações, políticas para termos compostos e sinônimos, e garantias aceitáveis.

Nos métodos de desenvolvimento de vocabulários controlados, geralmente são executadas as seguintes atividades: coletar, selecionar e classificar termos; decidir sobre o uso de nomes próprios; definir relações entre termos; determinar propósito, escopo e audiência; analisar domínio, literatura, necessidades, atores (pessoas envolvidas em atividades) e atividades; definir escopos de termos; definir e restringir significados de termos; definir formas de termos; escolher entre sinônimos ou entre termos cujos significados são diferentes no uso comum, mas que podem ser usados para representar um mesmo conceito; estabelecer relações entre sinônimos ou entre quase sinônimos; controlar sinônimos; estabelecer relações de associação; distinguir entre homógrafos; eliminar ambiguidades; testar e validar vocabulário; focar nos usuários; escolher termos apropriados aos usuários; atentar para critérios de suficiência, necessidade e padronização; atentar para que cada termo represente conceito no nível mais específico apropriado ao uso; atentar para que cada termo represente um conceito; qualificar o termo se ele representar conceitos diferentes; se diferentes termos representam um mesmo conceito, definir um termo como preferido e os outros como sinônimos ou pseudônimos; preparar produto final (ABBAS, 2010; GLUSHKO, 2013; HARPRING, 2010; MAI, 2006; MAI, 2008; NISO, 2005; ROWLEY; HARTLEY, 2008; SVENONIUS, 2000; SVENONIUS, 2003). O desenvolvimento de vocabulário controlado pode usar processamento de linguagem natural, inteligência artificial e análise de citações (HJØRLAND, 2002). Em mapeamento sistemático de literatura sobre vocabulários contro-

lados em Engenharia de Software, realizado pelo autor desta tese, as seguintes atividades foram identificadas em processos de desenvolvimento de vocabulários controlados: revisar literatura, realizar *workshop*, analisar documentação sobre software, avaliar ferramenta, analisar projeto de software, realizar entrevista, realizar estudo de caso, realizar pesquisa com pessoa que atue na área abordada pelo vocabulário controlado, realizar revisão por especialista na área abordada pelo vocabulário controlado, selecionar classe a partir de vocabulário controlado existente, fundir resultado de pesquisa a vocabulário controlado existente, definir critério de desenho (*design*), e refinar vocabulário controlado existente.

Considerando que diferentes domínios podem impor diferentes demandas a sistemas de recuperação e organização da informação, no desenvolvimento e implementação de vocabulários controlados, é importante entender práticas de trabalho, necessidades de informação, usos de recursos e linguagens adotadas em domínios e em comunidades específicas (ABBAS, 2010; HJØRLAND, 2002). Segundo Mai (2006), os métodos de desenvolvimento de vocabulários controlados frequentemente prestam pouca atenção à análise de atores (pessoas envolvidas em ações), necessidades e tarefas nos domínios de aplicação. Para Mai (2006), considerar objetivos e usos dos recursos e propósitos dos vocabulários controlados resulta em vocabulários controlados que melhor atendem atividades, necessidades e demandas dos atores. Mai (2008) considera que desenvolver vocabulário controlado, focado em domínio, requer conhecimento sobre indexação e comportamento informacional e sugere abordagem de desenvolvimento de vocabulário controlado focada no entendimento de restrições a comportamentos informacionais em domínios. Nesse contexto, restrições são fatores externos aos atores e comuns a todos os atores em um contexto ou domínio. As restrições limitam e capacitam os atores a realizarem os seus trabalhos. Sugere ainda o uso do arcabouço *Cognitive Work Analysis* (CWA) no desenvolvimento de vocabulários controlados, o qual pode ser usado na análise de domínio de trabalho e de requisitos que o domínio apresenta aos atores. Pode também ser usado no desenvolvimento de recomendações de desenho (*design*) de sistemas voltados a facilitar a interação entre pessoas e informação. Para desenvolver essas recomendações, procura-se entender trabalho, comportamento informacional, contexto de trabalho e motivos de ações de atores. Esse arcabouço define estrutura para análise de interações entre ser humano e informação, foca restrições que moldam comportamentos de atores na busca por informação, que moldam necessidades de informação, que limitam e capacitam ações, e que influenciam necessidades de informação no domínio. Por meio desse arcabouço, são analisadas restrições e fatores que afetam o trabalho e o comportamento informacional. Esse arcabouço define dimensões para análise de restrições, em que cada dimensão contribui para o entendimento do domínio, de atividades no domínio, de recursos e de valores dos atores. São exemplos de dimensões: ambiente de trabalho, domínio de trabalho, organizacional, atividade, características de ator. A dimensão “ambiente de trabalho” engloba elementos fora do domínio do ator, mas que afetam o domínio do ator. A dimensão “domínio de trabalho” examina o trabalho

no domínio. A dimensão organizacional examina como trabalho é dividido entre atores no domínio de trabalho e examina a natureza do domínio de trabalho, visa entender a estrutura do domínio e determina restrições impostas por estrutura, cultura e valores da organização. A dimensão atividade examina atividades dos atores. A dimensão “característica de ator” identifica características de grupos de atores. As dimensões anteriormente relacionadas são interdependentes e podem ser usadas alternadamente em um processo (FIDEL; PEJTERSEN, 2004; MAI, 2006).

5.7.1 Desenvolvimento de taxonomia

Em Hedden (2010), é descrito método de desenvolvimento de taxonomias com fases de planejamento e de desenvolvimento. A fase de planejamento tem o objetivo de identificar propósitos da taxonomia, usuários da taxonomia, escopo da taxonomia, recursos necessários ao projeto. Na fase de desenvolvimento, são definidos aspectos relacionados à estrutura da hierarquia: se a estrutura é poli-hierárquica, quantidade de termos no topo da hierarquia, quantidade de níveis hierárquicos e qual informação se deve associar a cada termo. A abordagem de desenvolvimento pode ser descendente, ascendente ou mista. Hedden (2010) relaciona as seguintes atividades no desenvolvimento de taxonomias: coletar termos, identificar categorias no topo da hierarquia ou de facetas por meio de análise de conceitos, listar termos, identificar termos nos primeiros níveis hierárquicos, realizar revisão preliminar da taxonomia, definir níveis hierárquicos restantes, implementar taxonomia usando ferramenta, testar e revisar taxonomia. Considerando que taxonomias podem sofrer manutenção, em Hedden (2010), é sugerida a definição de políticas, processos e métodos de manutenção.

5.7.2 Desenvolvimento de tesouro

Para o desenvolvimento de tesouros, Hedden (2010) sugere as seguintes atividades: coletar termos a partir de amostras de conteúdo, analisar termos coletados e organizá-los em listas, importar listas de termos em software de gerenciamento de tesouro, construir tesouro usando ferramenta de gerenciamento de tesouro, consultar referências e especialistas, implementar tesouro e testar tesouro. Também sugere a definição de relações e atributos na medida em que cada termo é incluído, a definição de relações e atributos após incluir vários termos, e a adoção de abordagem que combine abordagens que foram anteriormente relacionadas. Para Aitchison, Bawden e Gilchrist (2000), no desenvolvimento de tesouros deve-se definir: fronteiras do assunto ao qual o tesouro se destina, procurando identificar áreas em que o tratamento deve ser profundo e áreas em que o tratamento pode ser superficial; tipos e quantidades de recursos de informação; crescimento das quantidades de recursos de informação; idiomas; perfis de usuários; questões que o sistema deverá ser capaz de responder; recursos financeiros e recursos de pessoal. Por fim, ainda sobre o

desenvolvimento de tesouro, [Stellato et al. \(2015\)](#) destacam a importância da colaboração na distribuição de trabalho e no alcance de consenso.

No desenvolvimento de tesouros, as relações entre termos devem ser apresentadas e identificadas de modo padronizado. Além de termos e relações entre termos, os tesouros podem conter regras de uso e informação sobre conceitos. A estrutura de tesouros e os princípios de desenvolvimento dos mesmos são abordados em diversas publicações e normas de organizações internacionais ([ABBAS, 2010](#); [HARPRING, 2010](#); [NISO, 2005](#); [KLESS et al., 2015](#); [REDMOND-NEAL](#); [HLAVA, 2005](#)). [Hjørland \(2002\)](#) observa que tesouros frequentemente são vocabulários controlados para domínios específicos e que as metodologias para seu desenvolvimento podem ser consideradas uma forma de análise de domínio. Também observa a relação existente entre o método analítico facetado usado na construção de classificações, e metodologias dominantes usadas na construção de tesouros.

Na construção de tesouros, podem ser usadas classificações facetadas e análise de facetas. Segundo [Aitchison, Bawden e Gilchrist \(2000\)](#), na construção de tesouros, classificações facetadas podem ser usadas na determinação de relações entre conceitos, por exemplo. A análise de facetas pode ser usada para organizar os termos em um tesouro e gerar a estrutura do tesouro no nível macro ([MARTÍNEZ et al., 2011](#)). Nos tesouros cujas estruturas hierárquicas são subdivididas em facetas, os termos membros de cada faceta compartilham características que os distinguem de membros de outras facetas. No desenvolvimento de vocabulários controlados no domínio da engenharia e de assuntos relacionados, um dos primeiros usos de análise de facetas ocorreu no desenvolvimento do tesouro *Thesourofacet* (descrito em [Aitchison \(1970\)](#)) ([BARRE, 2010](#)). Um exemplo mais recente de tesouro com estrutura organizada em facetas é o *Art & Architecture Thesaurus*, no domínio da arte e arquitetura, ([THE GETTY RESEARCH INSTITUTE, 2015](#)).

5.7.3 Desenvolvimento de ontologia

A engenharia de ontologias é disciplina que engloba processos de desenvolvimento de ontologias, ciclo de vida de ontologias, métodos, metodologias, ferramentas e linguagens para facilitar o desenvolvimento de ontologias ([GÓMEZ-PÉREZ; CORCHO; FERNÁNDEZ-LÓPEZ, 2004](#)). Entre as contribuições da engenharia de ontologias, é possível listar as seguintes: linguagens, ferramentas e metodologias de desenvolvimento ([CORCHO; POVEDA-VILLALÓN; GÓMEZ-PÉREZ, 2015](#)). O ciclo de vida de ontologia pode ser dividido em fases. Em [Neuhaus et al. \(2013\)](#), é proposto um modelo de ciclo de vida de ontologia composto por fases com as seguintes responsabilidades: desenvolvimento de requisitos, análise ontológica, desenho (*design*) de ontologia, desenho (*design*) de sistema, desenvolvimento e reuso de ontologia, desenvolvimento e integração de sistema, implantação, operação e manutenção. A ordem e a frequência de execução de atividades no ciclo de vida proposto dependem dos processos seguidos. [Neuhaus et al. \(2013\)](#) também

relacionam perguntas que devem poder ser respondidas ao final de cada fase no ciclo de vida.

No desenvolvimento de ontologia, diversos cenários existem. Por exemplo, ontologia única (*single ontology*), conjunto de ontologias interconectadas e rede de ontologias. Ontologia única é ontologia sem relações com outras ontologias. Conjunto de ontologias interconectadas é composto por ontologias com algum tipo de relação dependente de domínio. Rede de ontologias é coleção de ontologias separadas e relacionadas por relações variadas. A seguir, são listadas possíveis relações entre ontologias em uma rede: ontologia em desenvolvimento é versão de ontologia existente, ontologia em desenvolvimento importa ontologia existente, ontologia em desenvolvimento estende ontologia existente, ontologia em desenvolvimento é composta por módulos, na ontologia em desenvolvimento existem componentes mapeáveis para ontologias existentes. Por fim, a seguir são relacionados cenários no desenvolvimento de redes de ontologias: desenvolvimento da especificação à implementação; reuso e reengenharia de recursos não ontológicos; reuso de recursos ontológicos; reuso e reengenharia de recursos ontológicos; reuso e fusão (*merging*) de recursos ontológicos; reuso, fusão e reengenharia de recursos ontológicos; reuso de padrões de projeto (*design pattern*) de ontologias; reestruturação de recursos ontológicos; adaptação de ontologia a outra linguagem ou comunidade (BAONZA, 2010).

O processo de desenvolvimento de ontologia é composto por atividades por meio das quais as necessidades dos usuários são traduzidas em ontologia. As atividades de desenvolvimento podem ser organizadas em classes, como gerenciamento, pré-desenvolvimento, desenvolvimento, pós-desenvolvimento e suporte. Considerando essas classes, agendar, controlar e garantir qualidade são atividades de gerenciamento, enquanto estudar ambiente e estudar viabilidade são atividades de pré-desenvolvimento. Especificar, conceituar, formalizar e implementar são atividades de desenvolvimento. No pós-desenvolvimento, são executadas atividades relacionadas à manutenção e ao uso. Adquirir conhecimento, avaliar, integrar, documentar, fundir (*merge*), gerir configuração e alinhar são atividades de suporte (BAONZA, 2010; GÓMEZ-PÉREZ; CORCHO; FERNÁNDEZ-LÓPEZ, 2004). No desenvolvimento de ontologia, geralmente são também executadas atividades para elicitare e especificar requisitos. Neuhaus et al. (2013) consideram que um modo de especificar requisitos é formular questões de competência, que são questões em linguagem natural que a ontologia deve ser capaz de responder em determinados cenários.

A importância de cada atividade no desenvolvimento de uma ontologia depende da característica da aplicação, do nível de automação usado, da complexidade da ontologia, das fontes de informação e das experiências dos envolvidos. Especialista no domínio modelado, engenheiro de ontologia especializado em representar conhecimento e em ferramentas de desenvolvimento e usuário são possíveis papéis dos envolvidos (ROWLEY; HARTLEY, 2008; SIMPERL; TEMPICH, 2006). Em Arp, Smith e Spear (2015), são relacionadas as

seguintes atividades no desenvolvimento de ontologias: demarcar assunto, coletar informação, estruturar termos em hierarquia, promover coerência, promover compatibilidade com outras ontologias, facilitar compreensão (particularmente por definições de termos) e formalizar ontologia em linguagem usável por computador. Também apresentam observações sobre o desenvolvimento de ontologias, como: ontologia deve descrever a realidade, e ontologias existentes devem ser consideradas como referências e reusadas quando possível. Sugerem que, no projeto de ontologias, deve-se balancear utilidade e realismo; iniciar com aspectos do domínio mais fáceis de serem entendidos; usar nomes no singular; evitar uso de acrônimos e de abreviações; associar cada termo a um identificador alfanumérico único; usar termos unívocos (com o mesmo significado em todos os usos); usar expressões unívocas para relações; distinguir o que é genérico daquilo que é particular; prover definições para termos não raiz; capturar características essenciais nas definições; evitar circularidades nas definições; nas definições, usar termos mais simples do que os termos sendo definidos; usar definições que possam ser substituídas pelos termos sendo definidos sem que isso altere significados.

Neuhaus et al. (2013) observam que não existe consenso sobre metodologia de desenvolvimento de ontologias. Segundo Uschold e Gruninger (1996), as metodologias de desenvolvimento de ontologias devem englobar fases para identificação de propósito e de escopo; construção da ontologia, que engloba captura da ontologia, codificação da ontologia e integração com ontologias existentes; avaliação; e documentação. Consideram ainda que essas metodologias devem englobar técnicas, métodos e princípios para cada fase da metodologia, indicações de relações existentes entre fases, entradas e saídas de cada fase.

As metodologias de desenvolvimento de ontologias podem ser classificadas como metodologias sem reuso de ontologias ou reusando ontologias sem transformá-las, metodologias para reengenharia de ontologias, e metodologias para desenvolvimento colaborativo de ontologias. As metodologias de desenvolvimento de ontologias variam quanto à dependência em relação à aplicação que usa a ontologia (FERNÁNDEZ-LÓPEZ; GÓMEZ-PÉREZ, 2002). Existem outros critérios para a classificação de metodologias de desenvolvimento de ontologias, por exemplo, modelo de processo de desenvolvimento; nível de detalhamento; proposta de processos; recomendações para formalização; estratégia para identificar conceitos; modelo de ciclo de vida; técnicas; ontologias e sistemas desenvolvidos; suporte para construção colaborativa, reuso e interoperabilidade; nível de dependência quanto às aplicações (FERNÁNDEZ-LÓPEZ; GÓMEZ-PÉREZ, 2002; CORCHO; FERNÁNDEZ-LÓPEZ; GÓMEZ-PÉREZ, 2003; IQBAL et al., 2013). Outras análises de metodologias de desenvolvimento de ontologias e propostas de classificações são encontradas em Jones, Bench-Caponand e Visser (1998), Fernández-López e Gómez-Pérez (2002), Corcho, Fernández-López e Gómez-Pérez (2003) e Iqbal et al. (2013). A seguir, são descritas metodologias de desenvolvimento frequentemente referenciadas em fontes de informação

sobre ontologias.

5.7.3.1 *Ontology Development 101*

Um processo de desenvolvimento de ontologia frequentemente referenciado é descrito em [Noy e McGuinness \(2001\)](#). Esse processo é iterativo e composto pelas seguintes atividades: definir domínio e escopo da ontologia, avaliar possibilidade de reuso de ontologias, enumerar termos importantes na ontologia, definir classes (classes representam conceitos no domínio) e hierarquia de classes, definir propriedades das classes, definir facetas de *slots*, e criar instâncias. Para definir domínio e escopo da ontologia, é sugerida a avaliação dos seguintes aspectos: qual é o domínio abordado pela ontologia, qual é a finalidade da ontologia, quais são os tipos de questões que devem ser respondidas pela informação na ontologia, quem usará a ontologia e quem será responsável pela manutenção da ontologia. Na enumeração de termos importantes, é sugerida a geração de uma lista abrangente de termos, sem preocupação quanto à sobreposição de conceitos, às relações entre termos, às propriedades de conceitos, ou à distinção entre classes e *slots*. A definição da hierarquia de classes pode seguir abordagem descendente, ascendente ou mista. Na abordagem descendente, a definição de classes que representam conceitos mais genéricos é seguida da definição de classes que representam conceitos mais específicos. Por sua vez, na abordagem ascendente, a definição de classes que representam conceitos mais específicos é seguida pela definição de classes que representam conceitos mais genéricos. Na abordagem mista, inicialmente são definidas classes que representam conceitos de destaque e, em seguida, esses conceitos são generalizados ou especializados. As classes são identificadas a partir da lista de termos anteriormente elaborada e são organizadas hierarquicamente por meio da identificação de classes cujos objetos são também objetos de outras classes. A identificação de propriedades que se tornarão *slots* das classes é iniciada por meio da análise da lista de termos anteriormente elaborada. Para cada *slot*, podem ser definidos tipo de valor, faixa de valores e cardinalidade (quantos valores o *slot* pode ter). Finalmente, são criadas instâncias de classes. Para isso, é necessário escolher a classe, criar a instância e atribuir valores a *slots*. Em [Noy e McGuinness \(2001\)](#), também são feitas recomendações para definição de classes, hierarquias de classes etc.

5.7.3.2 *METHONTOLOGY*

A metodologia *METHONTOLOGY* é descrita em [Fernández, Gómez-Pérez e Juristo \(1997\)](#). Nessa metodologia, as atividades são distribuídas nas seguintes fases: Especificação, Aquisição de conhecimento, Conceituação, Integração, Implementação, Avaliação e Documentação. As atividades na fase Especificação têm o objetivo de produzir uma especificação informal, semiformal ou formal. A especificação deve conter ao menos o propósito, o nível de formalidade e o escopo da ontologia. Na fase “Aquisição de conhecimento”, são sugeridas entrevistas com especialistas e análise de textos. Na fase

Conceituação, o conhecimento no domínio é estruturado em um modelo conceitual no qual é usado o vocabulário identificado na fase Especificação. Esse modelo inclui representações intermediárias que descrevem o domínio de aplicação. As atividades dessa fase são: construir glossário de termos, agrupar e representar conceitos e verbos relacionados. A fase Integração tem o objetivo de avaliar a conveniência de reuso de definições em ontologias existentes. Nessa fase, é sugerida a construção de um documento em que sejam relacionados, para cada termo cuja definição seja reusada, nome do termo no modelo conceitual, nome da ontologia com a definição sendo reusada e nome do termo reusado. A fase Implementação resulta na ontologia codificada em uma linguagem. Nessa fase, é usado ambiente de desenvolvimento que suporte as ontologias selecionadas na fase Integração. A fase Avaliação descreve como a ontologia foi avaliada, técnicas usadas, erros identificados e fontes de conhecimento usadas. Nessa fase, a ontologia é verificada e validada. A documentação da ontologia é composta por artefatos resultantes das fases anteriores.

Toda uma fase é dedicada na *METHONTOLOGY* à conceituação. Essa fase resulta em modelo conceitual e nela são sugeridas, como atividades, construir glossário de termos, construir grupos de conceitos e verbos relacionados, descrever conceitos e verbos. Para descrever os conceitos, é sugerido executar as atividades descritas em [Gómez-Pérez, Fernández e Vicente \(1996\)](#): identificar conceitos, instâncias, atributos e valores em dicionário de dados; classificar grupos de conceitos em árvores de classificação; descrever constantes em tabela de constantes; descrever instâncias atributos de instâncias, atributos de classes e fórmulas em tabelas; agrupar atributos em árvores de classificação de atributos. No dicionário de dados, são sugeridos para cada conceito: nome, sinônimos e acrônimos, descrição de significado do conceito, instâncias, atributos de classe ou propriedades relevantes do conceito, atributos de instância ou propriedades relevantes que descrevem instâncias do conceito. A árvore de classificação de conceitos organiza os conceitos em taxonomia composta por hierarquia de classes. Para cada conjunto disjunto de conceitos, é sugerida a construção de uma árvore de classificação de conceitos, que leva à construção de ontologia no domínio. A construção de árvore de classificação tem o objetivo de evitar repetição de conceitos, redundâncias e relações circulares. Todo conceito em árvore de classificação é definido no dicionário de dados. Para descrever verbos, na *METHONTOLOGY*, é sugerida a construção de dicionário de verbos, com significados dos verbos expressos de modo declarativo, e a construção de tabelas que especifiquem condições a serem satisfeitas antes de ações ou condições garantidas após ações.

5.7.3.3 *On-To-Knowledge Methodology*

A metodologia *On-To-Knowledge Methodology* (OTKM) ([SURE; STAAB; STU- DER, 2004; STAAB et al., 2001](#)) é composta pelas fases Estudo de viabilidade, Pontapé inicial (*kickoff*), Refino, Avaliação, e Manutenção. As atividades na fase “Estudo de viabilidade” procuram identificar problemas, oportunidades e potenciais soluções, e visam

prover suporte a decisões de viabilidade econômica e técnica do projeto, considerando a perspectiva organizacional. As atividades na fase “Pontapé inicial” têm o objetivo de especificar requisitos da ontologia. Essa especificação é realizada em um artefato onde é descrita a área de aplicação da ontologia e ao que a ontologia deve prover suporte. Nesse artefato, são também relacionadas as fontes de informação necessárias à descrição semiformal da ontologia. Esse artefato será usado na tomada de decisão sobre a inclusão, exclusão e estrutura hierárquica de conceitos na ontologia e deve conter meta da ontologia, domínio e escopo, aplicações às quais é provido suporte, fontes de informação, usuários e cenários de uso, questões de competência, e relação de ontologias potencialmente reusáveis. Nessa fase, também é elaborada uma descrição semiformal da ontologia, que pode ser na forma de grafo associado a texto descritivo. A fase Refino visa desenvolver ontologia que atenda aos requisitos especificados e é composta pelas seguintes atividades: entrevistar especialistas no domínio, construir taxonomia com conceitos identificados, construir ontologia preliminar a partir da taxonomia por meio de acréscimo de relações variadas à taxonomia, expressar a ontologia preliminar em linguagem formal. Essas atividades podem ser realizadas ciclicamente e podem ser reusadas ontologias identificadas. Na fase Avaliação, são avaliados aspectos tecnológicos, satisfação dos usuários e se a ontologia satisfaz aos requisitos especificados. Nessa fase, a ontologia é testada no ambiente de aplicação. Avaliação e Refino são fases cíclicas. A partir de resultados da avaliação, atividades de refino podem ser realizadas para revisar e expandir a ontologia. A fase Manutenção adequa a ontologia a mudanças.

5.7.3.4 *NeOn Methodology*

Em Baonza (2010) e [Ontology Engineering Group \(2015\)](#), é descrita a *NeOn Methodology*, metodologia para construir redes de ontologias (coleção de ontologias relacionadas) e evoluir ontologias em ambientes distribuídos. Os componentes dessa metodologia são: cenários para a construção colaborativa de ontologias e de redes de ontologias, glossário de termos sobre processos e atividades na construção de redes de ontologias, guias (*guideline*) para processos e atividades. Nos cenários de desenvolvimento de rede de ontologias, são realizadas atividades voltadas à aquisição de conhecimento, documentação, gestão de configuração e avaliação. Os cenários enfatizam reuso e reengenharia de recursos ontológicos e não ontológicos. A seguir, são listados cenários descritos na metodologia: da especificação à implementação (rede de ontologias desenvolvida sem reuso de recursos existentes); reuso e reengenharia de recursos não ontológicos; reuso de recursos ontológicos; reuso e reengenharia de recursos ontológicos; reuso e fusão (*merging*) de recursos ontológicos; reuso, fusão e reengenharia de recursos ontológicos; reuso de padrões de projeto de ontologias (*ontology design pattern*); reestruturando recursos ontológicos; e localizando recursos ontológicos. Esses cenários podem ser combinados e são decompostos em processos e atividades. Para processos e atividades, são fornecidos guias com nome, definição, entradas, saídas,

identificação dos envolvidos, e momento no qual processo ou atividade deve ser executado.

Baonza (2010) também define modelos de ciclo de vida de projetos de redes de ontologias. Esses modelos descrevem como organizar processos e atividades em fases e se baseiam em modelos de ciclo de vida propostos em Engenharia de Software. Cascata e iterativo-incremental são modelos de ciclo de vida propostos para projetos de redes de ontologias. No modelo de ciclo de vida denominado cascata, uma fase precisa ser concluída antes que a seguinte inicie. Para o modelo cascata ser aplicado, é recomendado conhecer os requisitos no início do desenvolvimento. Nesse modelo de ciclo de vida, é importante que os requisitos se mantenham estáveis durante o desenvolvimento. Esse modelo de ciclo de vida é recomendado em projeto de curta duração, projeto com requisitos fechados, projeto que vise desenvolver ontologia existente em nova linguagem ou formalismo, e projeto de ontologia que abranja domínios pequenos e entendidos. No modelo de ciclo de vida iterativo-incremental, as atividades são distribuídas em iterações (miniprojetos com durações definidas). Esse modelo de ciclo de vida é recomendado em projeto com muitos participantes, projeto que envolva domínio pouco entendido, projeto cujos requisitos não sejam conhecidos no início do desenvolvimento ou cujos requisitos mudem durante o desenvolvimento.

Um dos focos da *NeOn Methodology* é a especificação de requisitos de ontologias. As atividades sugeridas para especificar requisitos são: identificar propósito, escopo e linguagem de implementação; identificar usuários finais; identificar usos; identificar requisitos; agrupar requisitos funcionais; validar requisitos; priorizar requisitos; extrair terminologia e sua frequência. A *NeOn Methodology* também foca o planejamento e agendamento (*scheduling*). No agendamento, são identificados processos e atividades, encadeamento entre eles e recursos para desenvolvimento da ontologia. Para auxiliar na seleção do modelo de ciclo de vida do projeto, é sugerido definir, por exemplo, se recursos não ontológicos serão usados no desenvolvimento. Na *NeOn Methodology*, são propostas correspondências entre fases no modelo de ciclo de vida, processos e atividades. Para o planejamento e agendamento dos projetos, é proposta ferramenta e as seguintes atividades: selecionar modelo de ciclo de vida, selecionar cenários, atualizar plano, estabelecer restrições de recursos e atribuições. A *NeOn Methodology* também foca o reuso de recursos ontológicos. Na *NeOn Methodology*, processos e atividades são classificados em processos e atividades de gerenciamento, orientadas a desenvolvimento, e de suporte. O grupo “processos e atividades orientadas a desenvolvimento” é dividido em processos e atividades de pré-desenvolvimento, de desenvolvimento, e de pós-desenvolvimento. Na *NeOn Methodology*, conceituação é definida como “atividade de organizar e estruturar informação obtida durante processo de aquisição, em modelos significativos no nível de conhecimento e de acordo com o documento de especificação de requisitos”, é alocada no grupo “processos e atividades de desenvolvimento”.

5.7.4 Ferramentas de desenvolvimento

Em processo de desenvolvimento e em processo de publicação de vocabulário controlado podem ser usadas ferramentas tais como editor de planilha, ferramenta para suporte à gestão de qualidade e ferramenta para identificação automatizada de defeitos (BLU-MAUER, 2016; HEDDEN, 2010; WILL, 2013). Entre as ferramentas para desenvolvimento de vocabulários controlados, *VocBench* e *TemaTres* são exemplos. *VocBench* é ferramenta para desenvolvimento e gerenciamento de tesouros, listas de autoridade e glossários, provê suporte a fluxo editorial para facilitar o desenvolvimento colaborativo de terminologias, dados ligados (*linked data*) e *Web* semântica. Essa ferramenta é um produto de software aberto desenvolvido pela Organização das Nações Unidas para Alimentação e Agricultura (FAO) e pela *Università degli Studi di Roma Tor Vergata*. A comunidade de usuários dessa ferramenta é principalmente composta por organizações públicas. A ferramenta *VocBench* é usada no desenvolvimento do *AGROVOC*, tesouro multilíngue composto por termos em áreas de interesse da FAO (FAO, 2015b; FAO, 2015a). O *TemaTres*, por sua vez, é um pacote de software composto por servidor e por ferramenta para desenvolvimento e gerenciamento de taxonomias, tesouros e vocabulários multilíngues, que possibilita importação e exportação de vocabulários representados em formatos variados (FERREYRA, 2016). Para publicação de vocabulário controlado, pode-se usar ferramentas como *Skosmos* (vide Suominen et al. (2015)) e *SKOS Play* (vide Francart (2017)), que possibilitam publicação de vocabulários controlados representados como *Simple Knowledge Organization System* (SKOS).

Existem diversas ferramentas para suporte a processo de desenvolvimento de ontologia. Corcho, Fernández-López e Gómez-Pérez (2003) e Khondoker e Mueller (2010) relacionam essas ferramentas considerando os seguintes critérios: desenvolvedor da ferramenta, versão atual, data da versão atual, arquitetura de software, possibilidade de extensão, meio para armazenamento de ontologias, gerenciamento de cópias de segurança, interoperabilidade, paradigma de representação de conhecimento, suporte metodológico, serviços de inferência, recursos relacionados à usabilidade, domínio de aplicação, satisfação de usuários. Um arcabouço usado no desenvolvimento de ontologias e de aplicações que usam ontologias é o *Protégé*, software aberto composto por conjunto de ferramentas (vide Research (2015)). Ferramentas para suporte à avaliação da qualidade de ontologia são relacionadas em Aruna, Saranya e Bhandari (2011). Ainda sobre o desenvolvimento de ontologias, vale destacar que existem diversas linguagens voltadas à especificação de ontologias. Algumas dessas linguagens são usadas no desenvolvimento de aplicações para a rede Internet. Em Corcho e Gómez-Pérez (2000), são avaliadas diversas linguagens usadas na especificação de ontologias.

5.8 Reuso de vocabulário controlado

Antes de desenvolver um novo vocabulário controlado, é importante avaliar se um vocabulário controlado existente atende aos requisitos. Por exemplo, diversos autores destacam a importância de identificar tesouros existentes e definir se eles podem ser aproveitados, se eles podem ser adaptados ou se é necessário desenvolver um novo tesouro. Sobre a seleção de ontologia, pode ser orientada por método para esse fim. Em [Baonza \(2010\)](#), são relacionadas as seguintes atividades no reuso de recursos ontológicos: procurar recurso, avaliar recurso, comparar recursos, selecionar recurso e integrar recurso. Finalmente, vale destacar a existência de diretórios de vocabulários controlados, Por exemplo, [Basel University Library \(2017\)](#). Esses diretórios podem auxiliar na localização de vocabulários controlados existentes.

Algumas fontes de ontologias são: construção manual de ontologia, construção automática de ontologia, reuso de ontologia existente, extensão de ontologia por etiquetagem social ([REIMER, 2011b](#)). Em [Lozano-Tello e Gómez-Pérez \(2004\)](#), é descrito um método para processo de seleção de ontologia que relaciona características para comparação de ontologias, organizadas nas seguintes dimensões: conteúdo e organização de conteúdos da ontologia, linguagem usada na implementação da ontologia, metodologia usada no desenvolvimento da ontologia, ferramenta usada na construção e edição da ontologia, e custos do uso da ontologia. Em cada dimensão, as características são ainda organizadas em classes denominadas fatores. Por fim, no método descrito em [Lozano-Tello e Gómez-Pérez \(2004\)](#), é sugerido o uso de processo, denominado *Analytic Hierarchy Process* (AHP), para auxiliar na escolha da ontologia.

É importante destacar que as ontologias podem também ser desenvolvidas a partir de outras estruturas de vocabulários controlados, por exemplo, a partir de taxonomias acrescentando-se relações semânticas, restrições, regras de inferência e axiomas ([JAKUS et al., 2013](#)). Segundo [Jimeno-Yepes et al. \(2009\)](#), vocabulários controlados estruturados como tesouros também podem auxiliar no desenvolvimento de ontologias, como na seleção de termos, normalização de termos (conversão de múltiplos termos em um termo), aquisição de conhecimento e conceituação. Finalmente, em [Baonza \(2010\)](#) são descritos cenários de desenvolvimento de ontologias em que são usados esquemas de classificação, tesouros e folksonomias.

5.9 Tecnologias de representação

A representação de vocabulário controlado pode ser realizada por meio de tecnologias como *Resource Description Framework* (RDF), dados ligados (*linked data*), *Simple Knowledge Organization System* (SKOS) e *Web Ontology Language* (OWL) 2. Em [Gödert, Hubrich e Nagelschmidt \(2014\)](#), essas tecnologias são denominadas “tecnologias semânticas

para representação de conhecimento” (*semantic technologies for knowledge representation*).

5.9.1 Resource Description Framework

O *Resource Description Framework* (RDF) é um arcabouço proposto pelo *World Web Consortium* (W3C) para codificação e processamento de metadados. Por meio do RDF, é possível publicar e interligar informação sobre recursos na *World Wide Web*. O RDF pode ser usado para descrever recursos em qualquer domínio. Nesse contexto, o termo recurso designa “coisa de interesse”. O RDF pode contribuir na melhoria das capacidades de motores de busca, na descrição e classificação da informação, na descrição de páginas, na descrição de direitos de propriedade intelectual e na interoperabilidade entre aplicações (CAPLAN, 2003; LASSILA, 1997; POMERANTZ, 2015; YU, 2011). Em RDF, informação é representada por enunciados. Um enunciado RDF é denominado tripla. Cada tripla é composta por sujeito, predicado e objeto. Na figura 23, são apresentados elementos em tripla RDF.

Figura 23 – Elementos em tripla RDF

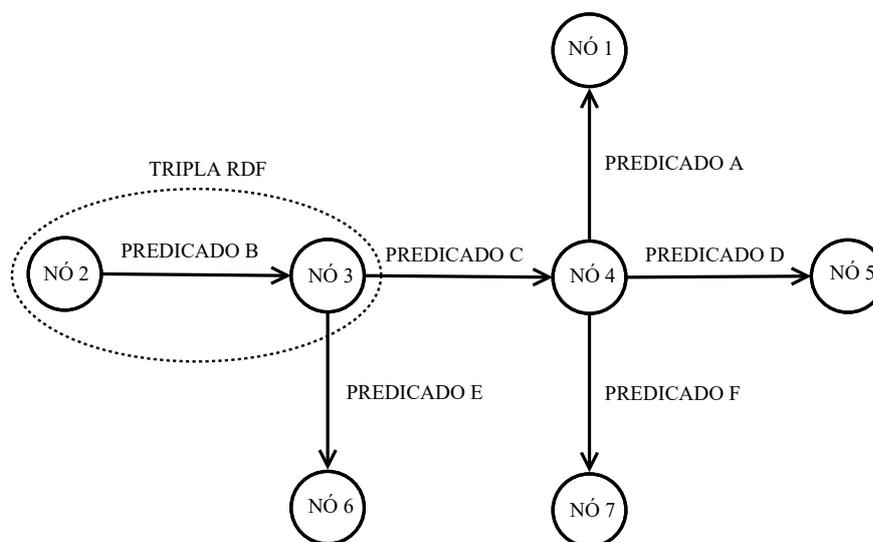


Fonte: Elaborado pelo autor

Cada tripla representa um fato e define uma relação entre recursos. Em cada tripla, sujeito e objeto são recursos, enquanto o predicado nomeia a relação entre recursos. Tripas podem ser armazenadas em arquivos ou em bases de dados. Bases de dados especialmente desenvolvidas para armazenar triplas são denominadas *triple stores*. Múltiplas triplas podem formar grafo onde sujeitos e objetos são nós, e predicados são arcos. Por meio desses grafos, é possível estabelecer relações entre metadados. Esses grafos podem ser escritos em vários formatos. Por exemplo, RDF/XML e RDFa. O formato RDF/XML define sintaxe por meio da qual é possível representar grafos RDF como documentos XML, enquanto o RDFa possibilita inserir triplas RDF em documentos HTML. A figura 24 apresenta triplas RDF em um grafo.

Em tripla RDF, nome de recurso pode ser globalmente identificado por meio de *International Resource Identifier* (IRI). Um IRI pode ocorrer em qualquer posição de uma tripla (sujeito, predicado ou objeto). O RDF é neutro quanto a aspectos semânticos e pode ser usado para descrever recursos em qualquer domínio. Para prover informação semântica, podem ser usados vocabulários com o RDF, sendo a definição de vocabulários suportada

Figura 24 – Exemplo de grafo RDF



Fonte: Elaborado pelo autor

por *RDF Schema*. O RDF não define termos para descrever classes de coisas e relações em domínios específicos. Esses termos se originam em vocabulários representados por meio de *Simple Knowledge Organization System* (SKOS) ou de linguagens como *Web Ontology Language* (OWL). Novos vocabulários podem ser definidos ou vocabulários podem ser reusados. *DCMI Metadata Terms*, *Friend-of-a-Friend* (FOAF), *Semantically-Interlinked Online Communities* (SIOC), *Description of a Project* (DOAP), *Good Relations Ontology*, *Creative Commons* (CC), *Bibliographic Ontology*, *OAI Object Reuse and Exchange* e *Review Vocabulary*, são alguns vocabulários. Finalmente, nível de popularidade (adoção), manutenção ativa, cobertura e expressividade são atributos relevantes na seleção de um vocabulário (CHU, 2010; HEATH; BIZER, 2011; HOOLAND; VERBORG, 2014; SCHREIBER; RAIMOND, 2014; YU, 2011). A título de ilustração, a figura 25 apresenta trecho de código RDF com termos do vocabulário FOAF.

5.9.2 Dados ligados

Dados ligados (*linked data*) é um modelo composto por práticas para publicar e interligar informação estruturada na *World Wide Web*. Esse modelo tem o objetivo de facilitar a descoberta e a integração de informação em fontes diversas. A seguir, são listados benefícios desse modelo: possibilidade de formulação de consultas mais expressivas do que comparações entre palavras-chave; uniformização e facilidade de integração devido ao compartilhamento de modelo de representação da informação; possibilidade de recuperar descrições de recursos identificados por *Uniform Resource Identifier* (URI) via *Hypertext*

Figura 25 – Exemplo de código RDF com termos do FOAF

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/">
  <foaf:Person>
  <foaf:name>Fernando Albuquerque</foaf:name>
  <foaf:title>Mr</foaf:title>
  <foaf:givenname>Fernando</foaf:givenname>
  <foaf:family_name>Albuquerque</foaf:family_name>
  <foaf:mbox rdf:resource="mailto:fernando@cic.unb.br"/>
  <foaf:homepage rdf:resource="http://cic.unb.br/~fernando/">
  <foaf:workplaceHomepage rdf:resource="http://cic.unb.br"/>
  </foaf:Person>
</rdf:RDF>
```

Fonte: Elaborado pelo autor

Transfer Protocol (HTTP); promoção de coerência por meio de ligações entre entidades em diferentes conjuntos de dados; facilidade de publicação e atualização pode reduzir o tempo para disponibilizar informação; flexibilidade, adaptabilidade e eficiência na gestão da informação; eficiência na recuperação e integração da informação; e facilidade de integração de dados em diversas fontes por meio de vocabulários compartilhados. O modelo dados ligados (*linked data*) adota RDF e mecanismo de acesso padronizado em HTTP, e resulta em espaço de dados estruturado como grafo composto por enunciados RDF (AUER, 2014; BAUER; KALTENBÖCK, 2012; HEATH; BIZER, 2011).

Berners-Lee (2006) relaciona as seguintes práticas no modelo dados ligados (*linked data*): uso de URI como nomes de “coisas”; uso de URI HTTP de modo que as pessoas possam consultar esses nomes; prover informação usando padrões quando alguém consultar um URI; inclusão de ligações para outros URI de modo a ser possível descobrir mais coisas. Berners-Lee (2006) também sugere as seguintes fases na transição para o modelo dados ligados (*linked data*): disponibilização de dados na *World Wide Web* em qualquer formato, mas com licença aberta; disponibilização de dados estruturados legíveis por máquina; disponibilização de dados estruturados legíveis por máquina em formatos não proprietários; disponibilização de dados segundo o anteriormente exposto, mais uso de padrões W3C para identificar coisas, com o objetivo de possibilitar que as pessoas possam apontar coisas; e disponibilização de dados segundo o anteriormente exposto, mais ligação a dados de outras pessoas para prover contexto.

5.9.3 Simple Knowledge Organization System

A recomendação do *World Wide Web Consortium* (W3C) denominada *Simple Knowledge Organization System* (SKOS) é baseada no RDF e provê modelo para representar vocabulários controlados estruturados como taxonomias ou tesouros. A recomendação SKOS possibilita a publicação e o uso de vocabulários como dados ligados (*linked data*). Com ela é possível desenvolver novos vocabulários ou portar vocabulários existentes. Os vocabulários controlados são expressos de modo a torná-los legíveis por máquina e são compostos por conceitos identificados por URI. Também é possível estabelecer relações hierárquicas ou relações associativas entre conceitos. Os vocabulários controlados podem ser representados por agrupamentos de conceitos em esquemas de conceitos, isto é, meios para publicar vocabulário. A interligação de conceitos em diferentes esquemas de conceitos possibilita a construção de uma rede de esquemas de conceitos, assim como o estabelecimento de relações entre conceitos com significados comparáveis (ALLEMANG; HENDLER, 2007; CORCHO; POVEDA-VILLALÓN; GÓMEZ-PÉREZ, 2015; ISAAC; SUMMERS, 2009; MILES; BECHHOFFER, 2009). Finalmente, no quadro 7, são listados elementos integrantes do vocabulário SKOS.

Quadro 7 – Elementos do vocabulário SKOS

DESCRIÇÃO	URI	DESCRIÇÃO	URI
Classe Concept	skos:Concept	Relações semânticas	skos:broader
Esquemas de conceitos	skos:ConceptScheme		skos:broaderTransitive
	skos:inScheme		skos:narrower
	skos:hasTopConcept		skos:narrowerTransitive
	skos:topConceptOf		skos:related
Rótulos (<i>labels</i>) léxicos	skos:altLabel		skos:semanticRelation
	skos:hiddenLabel	Coleções de conceitos	skos:Collection
	skos:prefLabel		skos:OrderedCollection
Notações	skos:notation		skos:member
	Propriedades de documentação		skos:changeNote
skos:definition			Propriedades de mapeamento
skos:editorialNote		skos:closeMatch	
skos:example		skos:exactMatch	
skos:historyNote		skos:mappingRelation	
skos:note		skos:narrowMatch	
skos:scopeNote	skos:relatedMatch		

Fonte: Adaptado de Miles e Bechhofer (2009)

Taxonomias e tesouros podem ser implementados como vocabulários *Simple Knowledge Organization System* (SKOS) com o objetivo de possibilitar sua publicação como dados ligados (*linked data*). Taxonomias e tesouros implementados desse modo podem ser

usados em diversas situações e atender necessidades semânticas de diversas organizações e sistemas de recuperação da informação. Por exemplo, a implementação de um tesouro como SKOS possibilita que ele reuse conjuntos de dados (*data set*) disponíveis na nuvem *Linked Open Data* (LOD) e que seja publicado e ligado a essa nuvem como triplas RDF, promovendo uso, validação e expansão. A estrutura de um tesouro SKOS é composta pelos seguintes níveis: nível conceitual, nível em que são identificados conceitos e são estabelecidas relações; nível de correspondência terminológica, nível em que termos são associados a conceitos; nível léxico, nível em que relações são definidas com o objetivo de interconectar termos (AMERI; URBANOVSKY; MCARTHUR, 2012; CARACCILO et al., 2012).

5.9.4 *Web Ontology Language*

Entre as linguagens para representar ontologias, se encontra *Web Ontology Language* (OWL) 2. Nesse contexto, uma ontologia é um conjunto de enunciados sobre um domínio de interesse, é composta por classes, propriedades e relações. A partir dos enunciados, podem ser realizadas inferências. Os objetos são denominados instâncias, as categorias são denominadas classes e as relações são denominadas propriedades. As classes podem ser organizadas em hierarquias e classes complexas podem ser descritas a partir de outras classes. Também é possível descrever uma classe por meio da enumeração dos indivíduos dessa classe (HITZLER et al., 2012; YU, 2011). Sobre o uso de linguagens para representar ontologias, Oren et al. (2006) destaca que apenas o uso delas não resulta em termos ontológicos, que uma ontologia denota um entendimento compartilhado, que uma linguagem para representar ontologias pode ser usada para capturar esse entendimento, mas antes é necessário alcançar esse entendimento.

5.10 Normas e padrões sobre vocabulário controlado

Em organizações (empresas, instituições etc.), a adoção de padrões pode resultar em benefícios tais como melhorar desempenho, reduzir riscos, disseminar conhecimento, melhorar compartilhamento de práticas de gerenciamento, melhorar interoperabilidade entre sistemas (BSI, 2015; ISO, 2015a; ISO, 2012b; ROWLEY; HARTLEY, 2008). Em sentido amplo, o termo padrão engloba desde recomendação em organização específica até padrão formalmente estabelecido por organização oficial dedicada à padronização. Padrão que se origina em organização oficial voltada à padronização é geralmente denominado “padrão de direito”, enquanto padrão que decorre de uso frequente em uma população é geralmente denominado “padrão de fato” (KOWALSKI; MAYBURY, 2002; MILLER, 2011). Existem diversos padrões relacionados à recuperação e organização da informação: padrões sobre codificação de dado, criação de conteúdo, catalogação, descrição de recurso, vocabulário controlado e esquema de metadados (ABBAS, 2010). Harpring (2010)

observa que os vocabulários controlados devem ser desenvolvidos de acordo com padrões estabelecidos, que devem ser usados vocabulários controlados compatíveis com padrões e que padrões facilitam a integração de vocabulários controlados. Lancaster (1986) informa que os padrões relacionados ao desenvolvimento de vocabulários controlados estruturados como tesouros surgiram na medida em que a experiência na construção desses vocabulários foi sendo registrada e codificada, e que esses padrões foram influenciados por teorias de classificação bibliográfica.

Existem normas sobre vocabulários controlados originadas em organizações como *International Organization for Standardization* (ISO), *American National Standards Institute/National Information Standards Organization* (ANSI/NISO), *Institute of Electrical and Electronic Engineers* (IEEE), e Associação Brasileira de Normas Técnicas (ABNT). A seguir, são listadas normas nesse contexto: ISO 25964-1:2011, ISO 25964-2:2011, ISO 5963:1985, ISO 214:1976 e ANSI/NISO Z39.19:2005. A norma ISO 25964-1:2011 *Thesauri and interoperability with other vocabularies - Part 1: Thesauri for information retrieval* engloba recomendações para desenvolvimento e manutenção de tesouros usados em aplicações de recuperação da informação. A norma ISO 25964-2:2011 *Thesauri and interoperability with other vocabularies - Part 2: Interoperability with other vocabularies* engloba elementos em vocabulários relacionados a aspectos de interoperabilidade (ISO, 2011b; ISO, 2011c). A norma ISO 5963:1985 *Documentation - Methods for examining documents, determining their subjects, and selecting indexing terms* engloba técnicas para análise de recursos de informação aplicáveis em indexação (ISO, 1985). A norma ISO 214:1976 *Documentation - Abstracts for publications and documentation* engloba recomendações para preparação e apresentação de resumos (*abstract*) (ISO, 1976). Finalmente, a norma ANSI/NISO Z39.19:2005 *Guidelines for the Construction, Format, and management of Monolingual Controlled Vocabularies* contém recomendações para construir, formatar e gerir vocabulários controlados. Essas recomendações são voltadas ao desenvolvimento de vocabulários controlados monolíngues (compostos por termos de só um idioma), particularmente, ao desenvolvimento de vocabulário controlado composto por termos no idioma inglês (NISO, 2005).

6 Metadados

Metadados são elementos importantes no contexto da recuperação da informação e no domínio da Arquitetura da Informação. Um possível uso de vocabulários controlados é na codificação de metadados. Nesse contexto, termos de vocabulário controlado podem ser usados como valores de elementos de metadados. Este capítulo apresenta uma visão geral sobre metadados. Este capítulo contém definições do termo metadado e também inclui informação sobre os seguintes assuntos: propósitos, usos, armazenamento, atributos, qualidade, classes, esquemas, perfis de aplicação, gestão e desenvolvimento de metadados. Por fim, este capítulo aborda normas sobre metadados.

6.1 Definições de metadado

Metadados são recursos de informação descritivos, contêm descrições de conteúdos de recursos de informação primários e pontos de acesso, e podem ser usados para auxiliar na recuperação de recursos de informação primários (ABBAS, 2010; TAYLOR; JOUDREY, 2009). A figura 26 consiste de hierarquia que ilustra o anteriormente exposto. Metadado pode descrever recurso contendo informação estruturada ou informação não estruturada, é composto por propriedades e por valores de propriedades (DAMA, 2010; MILLER, 2011).

Figura 26 – Metadado e conceitos relacionados



Fonte: Elaborado pelo autor

A origem do termo metadado é frequentemente atribuída a Jack E. Myers, que o usou em 1969 (GREENBERG, 2005). Apesar da origem do termo ser recente, metadados são usados há anos em catálogos de arquivos e de bibliotecas. Historiadores consideram que o primeiro catálogo de biblioteca foi criado por Calímaco para a Biblioteca de Alexandria

por volta de 245 AC. Entretanto, até meados dos anos 1990, o termo metadado foi usado principalmente por responsáveis por gerenciamento de dados, desenvolvimento e manutenção de sistemas. A popularização do termo ocorreu a partir da década de 1990, quando metadados passaram a ser criados para descrever informação na rede Internet, particularmente, na *World Wide Web*. Atualmente, metadados são relevantes em diversos domínios e contextos, por exemplo, em projetos colaborativos, na *web* semântica e em organizações responsáveis por coleções de recursos de informação (GILLILAND, 2008; CHU, 2010; DCMI, 2015; POMERANTZ, 2015).

O termo metadado é popularmente definido como “dado sobre dado” ou “informação sobre informação”, mas pode ter diferentes significados em diferentes comunidades e contextos devido a diferentes características e usos dos recursos de informação (CAPLAN, 2003; TAYLOR; JOUDREY, 2009). Day (2005) sugere que a definição do termo considere usos e funções de metadados. A seguir, são apresentadas algumas definições de metadado: informação criada ou capturada sobre recurso de informação por meio da qual é possível realizar certas funções em relação ao recurso de informação sobre o qual é o metadado (MILLER, 2011); descrição estruturada dos atributos essenciais de objeto de informação (GILL, 2008); informação estruturada ou descritiva sobre recurso, informação coletada e registrada sobre recurso (DLESE, 2005); informação estruturada sobre recurso de informação de qualquer tipo de mídia ou formato (CAPLAN, 2003); informação estruturada que descreve, explica, localiza e facilita recuperação, uso ou gestão de recurso de informação (NISO, 2004); informação estruturada associada a objeto com propósito de descoberta, descrição, uso, gestão e preservação (NISO, 2007); informação sobre objeto digital ou físico (IEEE, 2002); enunciado sobre recurso potencialmente informativo (POMERANTZ, 2015); descrição de informação em rede e de recursos digitais segundo padrão ou arcabouço criado para esse propósito, dados de catalogação e indexação criados para qualquer tipo de documento por meio de métodos tradicionais de descrição e organização da informação (CHU, 2010); qualquer enunciado sobre recurso de informação (GARSHOL, 2004). Por fim, vale destacar que Taylor e Joudrey (2009) consideram que várias definições de metadado têm em comum que metadado é informação estruturada para descrever atributos de recursos de informação e promover identificação, descoberta, seleção, uso, acesso e gerenciamento de recurso de informação.

6.2 Propósitos e usos de metadado

Os propósitos e usos de metadados em coleções de recursos de informação são: descoberta, descrição, organização, identificação, compartilhamento, reuso, recuperação, gerenciamento, documentação de autoria e de autenticidade de recursos de informação; estabelecimento de contexto; promoção da recuperação, preservação e interoperabilidade; gestão de conteúdos; gestão de direitos sobre recursos de informação; e refletir propó-

sitos e perspectivas das organizações que criam e gerenciam o recurso de informação (CHEN; CHEN; LIN, 2003; DAY, 2005; DCMI, 2015; DLESE, 2005; DUVAL et al., 2002; GARSHOL, 2004; HAYNES, 2004; ISO, 2009a; NISO, 2004). Em bibliotecas, metadados são geralmente desenvolvidos e usados para facilitar o acesso a conteúdos de coleções, e são frequentemente usados em catálogos em linha (*online*) de acesso público ou em bases de dados comerciais. Nesse contexto, os metadados geralmente são armazenados em registros que seguem regras de catalogação, estruturação e valores de dados (GILLILAND, 2008). É importante destacar que coleções podem ser pouco úteis sem metadados. É também importante destacar o papel dos metadados no acesso a recursos de informação em rede e na promoção da interoperabilidade. Buscas distribuídas a diferentes servidores ou buscas a repositórios de metadados centralizados são estratégias nesse contexto. Na primeira estratégia, a consulta é distribuída a diferentes computadores por meio de protocolos que possibilitam recuperar informação em redes de computadores. Os resultados são depois consolidados e integrados. Na segunda estratégia, os metadados são coletados periodicamente e armazenados em repositórios nos quais são realizadas buscas. Esse processo é denominado *metadata harvesting*. Nessa estratégia, os metadados podem ser enviados pelas fontes de metadados ou coletados automaticamente e regularmente para criar agregados de metadados. Esse processo ocorre por meio de protocolos apropriados (IEEE, 2002; MARCONDES; SAYÃO, 2001; MILLER, 2011). Ainda sobre interoperabilidade, há diferentes níveis de interoperabilidade entre aplicações que usam metadados. Na comunidade *Dublin Core*, por exemplo, são sugeridos os seguintes níveis de interoperabilidade: compartilhamento de vocabulários definidos em linguagem natural (os participantes concordam com os termos e as definições dos mesmos); compartilhamento de vocabulários baseados em semântica formal; compartilhamento de sintaxe abstrata para registros de metadados; registros aderentes a conjuntos de restrições, usando os mesmos vocabulários e refletindo modelos compartilhados do mundo (DCMI, 2015).

Entretanto, há também evidências de limitações associadas ao uso de metadados em determinados contextos. Hawking e Zobel (2007) relatam experimentos em que metadados descritivos em sítios de organizações agregaram pouco valor no posicionamento (*ranking*) de resultados de consultas. Entre as dificuldades associadas ao uso de metadados nesse contexto, os autores relacionam diferenças entre o vocabulário usado nas consultas e o vocabulário usado nos metadados; omissão de metadados, cópia de metadados de um recurso de informação para outro e cópia de metadados de um campo para outro por autores dos recursos de informação; metadados desatualizados; inconsistência na interpretação dos campos de metadados; dificuldade na distinção entre páginas, em decorrência do compartilhamento de descritores; dificuldade dos autores identificarem termos apropriados; e confusão em decorrência dos metadados serem usualmente invisíveis aos usuários. Ainda nesse contexto, Pomerantz (2015) descreve processo denominado “*keyword stuffing*”, processo no qual metadados com termos irrelevantes são associados a

páginas HTML com o objetivo de aumentar a frequência de recuperação dessas páginas por motores de busca.

6.3 Registro de metadado

Um registro de metadados é o agrupamento de propriedades e valores associados sobre um recurso de informação. Cada par composto por propriedade e valor é um ponto de acesso ao recurso de informação. Para um dado esquema de metadados, é recomendado existir só um registro de metadados para cada recurso. Um registro de metadados é composto por enunciados sobre um recurso. Recursos de informação são então descritos por enunciados. O enunciado é a unidade básica em metadados. Cada enunciado é composto por sujeito, predicado e objeto. O predicado estabelece relação entre sujeito e objeto. Sujeito é a entidade descrita, enquanto objeto é a entidade usada para descrever o sujeito. Registro de metadados pode ser incorporado ao recurso de informação sendo descrito ou ser armazenado separadamente. Metadados armazenados nos recursos de informação descritos, frequentemente são criados quando são criados os recursos (DUVAL *et al.*, 2002; MILLER, 2011; NISO, 2004; POMERANTZ, 2015). Enunciados que compõem registros de metadados podem ser armazenados como campos em bancos de dados. No desenvolvimento desses bancos de dados, geralmente é necessário definir para cada campo: obrigatoriedade, cardinalidade (quantidade de instâncias do campo em cada registro), restrições associadas aos tipos e valores de conteúdos do campo, e tamanho do campo. No desenvolvimento de software para acesso a esses bancos de dados, geralmente também é definida a visibilidade desses campos aos usuários (MILLER, 2011).

6.4 Atributos de metadado

Fonte, método de criação, natureza, estado, estrutura, semântica, nível de granularidade da descrição e nível de detalhamento da descrição são atributos relacionados a metadados. A fonte pode ser interna, intrínseca ou externa, enquanto o método de criação pode ser manual, automatizado ou semi-automatizado. Quanto à natureza, o metadado pode ser criado por especialista ou por não especialista. O estado pode ser estático, dinâmico, longo prazo ou curto prazo, enquanto a estrutura, pode ser estruturada ou não estruturada. A semântica pode ser controlada (adere a vocabulário, forma e regras padronizadas) ou não controlada, e o nível de granularidade da descrição, pode ser coleção, recurso de coleção ou parte de recurso de coleção. A granularidade depende de fatores tais como tipo de recurso, abordagem de organização da informação e necessidade de usuário. Quanto ao detalhamento, a descrição pode ser detalhada ou simplificada. Uma descrição detalhada pode melhorar a precisão da recuperação da informação, mas pode ser mais cara e dificultar a manutenção da consistência. Por sua vez, uma descrição simplificada pode

ser mais fácil de criar, ter menor custo e facilitar a interoperabilidade, mas pode gerar mais resultados falsos e requerer maior esforço para identificação de informação relevante (DAVIES, 2010; DUVAL et al., 2002; GILLILAND, 2008; TAYLOR; JOUDREY, 2009).

6.5 Qualidade de metadado

A qualidade de metadados é importante em vários contextos. Day (2005) destaca que, na comunidade científica, o aumento da informação em recursos digitais, a crescente quantidade de dados gerados e a ênfase no acesso aberto, tornam crucial metadados de qualidade. A fonte NISO (2007) relaciona as seguintes recomendações para qualidade de metadados: aderência a esquemas padronizados apropriados a recursos, usuários, uso atual e uso futuro da coleção; suporte à interoperabilidade; atributos de recursos expressos por termos controlados; inclusão das condições e termos de uso do recurso; suporte à gestão e preservação de recursos; registros de metadados com atributos de qualidade como autoridade, autenticidade, persistência e identificação única.

6.6 Classes de metadados

Há diversas propostas de classes de metadados, por exemplo, metadado descritivo, administrativo e estrutural. Metadados descritivos distinguem recursos, visam facilitar a descoberta, a identificação e a seleção de recursos. Essa classe facilita a identificação de conteúdos, contextos e significados de recursos em coleções. Podem ainda facilitar a recuperação de recursos individuais e de conjuntos de recursos a partir de características compartilhadas. Metadados administrativos proveem informação que facilita gerir recursos. Por meio dessa classe de metadado é possível prover informação relevante na gestão de recursos, preservação de recursos ou controle de acesso a recursos. Metadados estruturais descrevem como os recursos são compostos e registram relações existentes entre partes (CAPLAN, 2003; GREENBERG, 2005; MILLER, 2011; NISO, 2004; NISO, 2007; POME-RANTZ, 2015). Em Yee et al. (2003), são relacionadas as seguintes classes de metadados: plano, hierárquico e facetado (composto por conjuntos ortogonais de categorias). Em Stoica e Hearst (2007), são descritos metadados hierárquicos facetados e seu uso no contexto de interfaces com os usuários. Brasethvik (2004) sugere as classes metadado contextual e metadado semântico. Metadado contextual descreve propriedade contextual, por exemplo, autor, título, data de modificação e localização. Metadado semântico, por sua vez, descreve assunto. Taylor e Joudrey (2009) sugerem classificação de metadados de acordo com níveis de complexidade, por exemplo, simples (extraídos dos recursos), formato estruturado (conjuntos de elementos criados para o usuário em geral) e formato rico (conjuntos de elementos criados de acordo com padrões de codificação e de conteúdo). Por fim, segundo

Taylor e Joudrey (2009), as fronteiras entre classes de metadados não são fixas, e as definições de classes de metadados são frequentemente imprecisas.

6.7 Esquema de metadados

Esquemas de metadados são conjuntos de elementos (*element set*) de metadados, criados para certos tipos de recursos. São compostos por conjuntos de elementos com regras de uso definidas para propósitos particulares e englobam, por exemplo, regras de sintaxe. Um esquema de metadados é uma coleção de elementos com atributos formalizados em uma especificação, agrupados para prover suporte a funções como descoberta e administração de recursos. Em um esquema de metadados, os valores dos elementos podem ou não ser controlados e cada elemento é uma classe de enunciados que podem ser feitos sobre recurso (CAPLAN, 2003; GREENBERG, 2005; NISO, 2005; NISO, 2004; POMERANTZ, 2015). Para Duval et al. (2002), os seguintes princípios são relevantes a esquemas de metadados: modularidade que possibilite criar esquemas de metadados a partir de esquemas existentes, extensibilidade que possibilite acomodar necessidades específicas às aplicações, refinamento que possibilite escolha de níveis de detalhamento adequados às aplicações, e respeito a diversidades linguística e cultural.

Esquemas de metadados podem diferir dependendo dos elementos que os compõem, requisitos de codificação, vocabulários controlados usados e foco. Como possíveis focos destacam-se a descoberta, a identificação e a administração de recursos (TAYLOR; JOURNAL, 2009). Alguns esquemas de metadados são destinados à descrição de recursos em geral; outros, são destinados à descrição de certos tipos de recursos. Os esquemas de metadados podem definir aspectos semânticos, regras de conteúdo e regras de sintaxe. Os aspectos semânticos abordam significados de elementos de metadados, enquanto regras de conteúdo especificam aspectos relacionados à seleção e representação dos valores dos elementos. As regras de sintaxe abordam a codificação dos elementos, definem formatos dos elementos (CAPLAN, 2003). As especificações de esquemas de metadados podem ser recomendações flexíveis ou regras detalhadas. Uma especificação pode conter definições de elementos, definições de relações entre elementos, definições de valores aceitáveis para elementos, identificadores de fontes que definam valores aceitáveis para elementos, recomendações de sintaxe, definições sobre cardinalidade de elementos e refinamentos de definições de elementos. Para especificar sintaxe ou vocabulário, ao esquema de metadados, pode ser associado um esquema de codificação, geralmente composto por regras sobre a representação de elementos do esquema de metadados. É importante destacar que há arcabouços desenvolvidos para auxiliar processos de análise de esquemas de metadados (GREENBERG, 2005; POMERANTZ, 2015).

6.8 Perfil de aplicação de metadados

Para suprir as necessidades de informação de certas aplicações ou comunidades, podem ser necessários elementos de diferentes esquemas de metadados ou elementos de parte de um esquema de metadados. Nesse caso, pode ser definido perfil de aplicação (*application profile*) de metadados. Um perfil de aplicação de metadados pode promover o compartilhamento de informação em uma comunidade ou entre comunidades. Perfil de aplicação de metadados é mais específico do que esquema de metadados. A definição de um perfil de aplicação de metadados geralmente inclui a definição de elementos de metadados relevantes em contexto específico e pode englobar as seguintes atividades: descrever recomendações, regras e restrições para criar metadados; definir quais elementos usar; escolher e combinar elementos de diferentes esquemas; refinar definições de elementos; restringir valores de elementos; definir relações entre elementos e os seus valores; selecionar vocabulários controlados; configurar padrões (COYLE, 2009; DUVAL et al., 2002; TAYLOR; JOUDREY, 2009; NILSSON, 2008b; NILSSON, 2008a).

6.9 Gestão de metadados

O termo “gestão de metadados” pode designar processo executado em ciclo de vida de recurso de informação com os seguintes objetivos: criação, armazenamento, integração, controle, garantia da qualidade e segurança de metadados; suporte ao uso de metadados; facilitar o acesso a metadados. Esse processo pode englobar as seguintes atividades: entender requisitos de metadados; definir arquitetura de metadados; desenvolver e manter padrões de metadados; implementar ambiente gerenciado de metadados; criar, manter, integrar, consultar, reportar, distribuir, entregar e analisar metadados; gerenciar repositórios de metadados. São possíveis resultados da gestão de metadados: metadados, repositórios de metadados, modelos e arquiteturas de metadados, análises de metadados e procedimentos de controle de metadados. Embora a gestão de metadados seja responsabilidade típica de profissionais da informação, com a disponibilização em linha (*online*) de recursos de informação, os usuários de sistemas de informação têm também participado da gestão de metadados (GILLILAND, 2008; DAMA, 2010; ISO, 2009a).

Existem metodologias e modelos que podem ser usados na gestão de metadados. Chen, Chen e Lin (2003) descrevem a metodologia *Metadata Lifecycle Model* (MLM), composta por processo com as seguintes responsabilidades: aquisição de necessidades de metadados, revisão de padrões e projetos relevantes de metadados, investigação de necessidades de metadados, identificação de estratégias para esquemas de metadados e interoperabilidade com padrões conhecidos de metadados, preparação da especificação de requisitos de metadados, avaliação de sistemas de metadados, preparação de guia de melhores práticas, desenvolvimento de sistema de metadados, manutenção de serviço

de metadados, e avaliação de desempenho de metadados. Essas responsabilidades são distribuídas nos grupos Avaliação de requisitos e análise de conteúdo, Especificação de requisitos de sistema, Sistema de metadados, Serviço e avaliação. Möller (2013) sugere modelo de ciclo de vida que pode ser usado para classificar, comparar ou construir modelos de ciclo de vida de sistemas centrados em dados (*data centric*). Esse modelo é composto pelas fases Desenvolvimento de ontologia, Planejamento, Criação, Arquivamento, Refino, Publicação, Acesso, Uso externo, Retroalimentação, e Término. A fase “Desenvolvimento de ontologia” desenvolve modelo de domínio para metadados. Essa fase pode apresentar um modelo próprio de ciclo de vida. Na fase Criação, ocorre a criação de dados e metadados. Esses dados podem ser novos ou importados. Na fase Arquivamento, ocorrem processos como indexação e catalogação. Na fase Refino, ocorrem modificações aos dados, por exemplo, por meio de sua organização. Na fase Publicação, os dados são tornados acessíveis aos usuários. Na fase Acesso, os dados são acessados. Na fase “Uso externo”, ocorre o uso dos dados fora do sistema, por exemplo, por exportação de dados. Na fase Retroalimentação, ocorrem críticas dos usuários a dados e a metadados. Finalmente, na fase Término, o dado é removido do sistema. Nesse modelo de ciclo de vida, são papéis de participantes: criador de dados, criador de metadados, administrador ou usuário final.

6.10 Desenvolvimento e uso de metadados

Taylor e Joudrey (2009) relacionam as seguintes atividades na criação de metadados para sistemas de informação: prover descrição de recurso de informação e acesso a ela; prover informação necessária ao gerenciamento, preservação e estrutura de recurso de informação; e codificar. No desenvolvimento de coleções digitais, Miller (2011) destaca a fase em que ocorre desenho (*design*) de esquema de metadados e criação de metadados. No desenho (*design*) do esquema de metadados, sugere as seguintes atividades: analisar contexto, conteúdo e usuários com o objetivo de determinar requisitos funcionais do sistema de informação; selecionar e desenvolver conjunto de elementos de metadados; estabelecer especificações para elementos de metadados e para banco de dados que os armazenará; estabelecer vocabulários controlados e esquemas de codificação para elementos dos metadados; desenvolver guias para conteúdos dos elementos de metadados; documentar esquema de metadados. Para Miller (2011), analisar contexto engloba identificar aspectos do contexto que influenciem o desenho (*design*) dos metadados. Por sua vez, analisar conteúdo engloba identificar tipos de recursos, assuntos abordados e foco da coleção. Finalmente, analisar usuários engloba identificar usuários, necessidades de informação dos usuários, comportamentos dos usuários na busca por informação e atributos relevantes à localização e identificação dos recursos. A partir da análise de contexto, conteúdo e usuários, Miller (2011) sugere que sejam determinados requisitos funcionais do sistema de informação. Para facilitar a identificação desses requisitos, sugere que diversos aspectos sejam definidos: como

os usuários desejam consultar a coleção, quais atributos dos recursos serão consultados, qual informação dos recursos interessa aos usuários, qual informação é importante à identificação e interpretação dos recursos, como os recursos serão ordenados e agrupados pelos usuários, como os usuários desejam limitar resultados de consultas, como elementos dos metadados serão estruturados, para quais elementos dos metadados é necessário definir padrões de codificação e vocabulários controlados, como os elementos serão mapeados para o conjunto de elementos definido. Em seguida, ocorrem seleção e desenvolvimento do conjunto de elementos. Para isso, existem as seguintes alternativas: desenvolver conjunto de elementos sem referenciar padrões existentes, selecionar padrão de conjunto de elementos que atenda aos requisitos funcionais, desenvolver conjunto de elementos e mapeá-lo para conjunto de elementos padronizado. No desenho (*design*) do conjunto de elementos, Miller (2011) sugere que sejam considerados contexto, conteúdo, requisitos funcionais, produtos usados, esquemas suportados, comunidade, interoperabilidade, facilidade de compartilhamento, viabilidade futura e reuso. Em seguida, são especificados elementos e campos no banco de dados que armazenará os metadados. Podem ser definidos sobre os elementos: obrigatoriedade, cardinalidade, restrições de tipo, restrições de valor, restrições de conteúdo, tamanho do campo, visibilidade de elementos e visibilidade de campos. Para definir vocabulário controlado, Miller (2011) sugere analisar assunto do conteúdo e identificar vocabulários usados em produtos de software e pela comunidade. Também sugere que vocabulários controlados sejam desenvolvidos se não forem identificados vocabulários existentes que atendam às necessidades. Em seguida, ocorre o desenvolvimento de guias de conteúdo, que podem definir fontes de informação preferidas e procedimentos. Finalmente, ocorre a documentação do esquema de metadados. Após o desenho (*design*) do esquema de metadados, recursos de informação são analisados e descritos. Miller (2011) observa que o processo de criação de metadados, embora seja frequentemente denominado “descrição de recurso”, geralmente envolve mais do que descrever características de recursos de informação.

A fonte CIESIN (1998) descreve um processo de desenvolvimento de metadados, cujas atividades são: identificar recurso de dado/informação (engloba determinar importância e propriedade dos recursos para a coleção, assim como determinar se existe metadado para o recurso); revisar recurso de dado/informação (engloba determinar, o assunto, a audiência e o modo de uso do recurso, informação descritiva a ser incluída, informação em falta, e modo de acesso pelos usuários); criar registro de metadado (engloba definir título, escrever resumo, determinar termos de indexação e palavras-chave, completar conteúdo, registrar referências para fontes usadas na criação do registro de metadado); revisar registro de metadado (engloba revisão interna e externa do registro de metadado); editar conteúdo de metadado (se necessário) e realizar revisão final; e liberar registros para catálogo. CIESIN (1998) também inclui procedimentos de manutenção de metadados, que englobam as seguintes atividades: definir prazo para revisão periódica dos registros de metadados,

revisar conteúdos, atualizar conteúdos, realizar revisões internas e revisões externas. Em revisões de registros de metadados, são considerados atributos como atualidade, acurácia, consistência e completude.

O desenvolvimento de repositório organizado e gerenciado por meio de metadados requer atividades como: conceber produto; planejar aparência e características técnicas dos recursos; estimar requisitos de armazenamento; determinar necessidades de pessoal; selecionar e contratar pessoal; determinar e obter infraestrutura requerida; estimar custos, assegurar fundos e garantir continuidade dos mesmos; definir esquema de metadados; definir natureza dos valores de dados que podem ocupar os elementos; definir sintaxe para troca de metadados; desenho (*design*) de ferramenta para construir metadados; desenho (*design*) de banco de dados para armazenar metadados; reunir informação de direitos e créditos; treinar responsáveis por digitalizar recursos; digitalizar recursos ou coletar recursos digitais; desenho (*design*) e construção de interface com usuário; programar funções de busca e navegação; avaliar produto quanto à acessibilidade; realizar teste do produto; prover canais para retroalimentação (*feedback*) dos usuários; e disponibilizar produto. Quanto a atividades relevantes à recuperação da informação nesses repositórios, as seguintes podem ser relacionadas: identificar atributos pelos quais recursos serão procurados pelos usuários, definir e rotular atributos dos recursos como elementos em um esquema de metadados, e garantir que o conteúdo de cada campo pesquisável seja informado de modo a promover a recuperação da informação. Finalmente, vale destacar que, embora a criação manual de metadados possa resultar em metadados de qualidade, essa abordagem pode não ser conveniente quando for grande a quantidade de recursos de informação. Nesses casos, podem ser usados algoritmos para automatizar o processo (CHOPEY, 2005; STOICA; HEARST, 2007).

6.11 Anotação

O termo anotação é frequentemente usado para designar a atribuição de informação a recurso de informação, ou resultado desse processo. Pode também designar processo de atribuir metadado a recurso de informação. A anotação de um recurso de informação pode ser por pessoa, usar sugestões geradas automaticamente ou ser automático. A anotação pode ser informal, formal ou semântica. Na anotação formal, é usada linguagem formal, mas não termos ontológicos, enquanto que na notação semântica, são usados termos ontológicos, a terminologia tem significado que corresponde a uma conceituação compartilhada, são usados termos socialmente aceitos e compreendidos (HOBEL; REVENKO, 2016; KIRYAKOV et al., 2004; OREN et al., 2006). No contexto de metadados e ontologias, Davies (2010) define anotação semântica como processo em que metadados estabelecem ligações entre informação nos recursos e nas ontologias. Na recuperação da informação, Ontotext (2015) destaca que a anotação semântica é capaz de gerar resultados não explicitamente relacionados à consulta

original, além de consultas e filtragens complexas. Para [Hobel e Revenko \(2016\)](#), o uso de anotações com termos de vocabulário controlado, pode prover consistência, tradução, controle de foco e termos alternativos.

6.12 Normas e padrões sobre metadados

A gestão de metadados pode ser realizada mantendo-se compatibilidade com padrões. Isso pode resultar em benefícios, por exemplo, na integração de coleções de diferentes organizações ou quando a coleção de uma organização precisa ser pesquisada por mecanismos externos à organização ([CHOPEY, 2005](#)). Existem padrões sobre metadados de propósito geral, metadados de propósito educacional, metadados de objetos culturais, metadados de recursos visuais, metadados de arquivos e preservação, registros de metadados, estruturas de dados, conteúdos de dados, valores de dados, formato de dados ou codificação ([ABBAS, 2010](#); [MILLER, 2011](#)). Exemplos de normas são ISO 704:2009, ISO 1087-1:2000, ISO 2382:2015, ISO/IEC 10241:1992, ISO/IEC 11179, ISO 23081-1:2009, ISO 23081-2:2009 e ISO 23081-3:2011. A norma ISO 704:2009 *Terminology work – Principles and methods* estabelece princípios e métodos para preparação e compilação de terminologias em diversos campos, descreve relações entre objetos, conceitos e suas representações terminológicas, e estabelece princípios para a formação de designações e definições ([ISO, 2009c](#)). A norma ISO 1087-1:2000 *Terminology work - Vocabulary - Part 1: Theory and application* provê descrição de conceitos no campo da terminologia e esclarece o uso de termos nesse campo ([ISO, 2000](#)). As normas ISO/IEC 10241-1:2011 *Terminological entries in Standards - Part 1: General requirements and examples of presentation* e ISO/IEC 10241-2:2012 *Terminological entries in Standards – Part 2: Adoption of standardized terminological entries* especificam e exemplificam requisitos terminológicos em documentos de organizações normalizadoras ([ISO, 2011a](#); [ISO, 2012a](#)). A norma ISO 2382:2015 *Information Technology - Vocabulary* define vocabulário em Tecnologia da Informação ([ISO, 2015b](#)). A norma ISO/IEC 11179 *Metadata registries* aborda registros de metadados (*metadata registry*). Finalmente, ISO 23081-1:2009 *Information and documentation - Records management Processes - Metadata for records - Part 1: Principles*, ISO 23081-2:2009 *Information and documentation - Managing metadata for records - Part 2: Conceptual and implementation issues* e ISO 23081-3:2011 *Information and documentation - Managing metadata for records - Part 3: Self-assessment method* são normas que abordam conceitos sobre metadados e estabelecem arcabouço para definir e avaliar metadados. Ainda sobre normas e padrões, a seguir são descritos o *Dublin Core Metadata Element Set* e a norma *IEEE Learning Object Metadata*.

6.12.1 Dublin Core

O *Dublin Core Metadata Element Set* foi originalmente proposto em *workshop* realizado em 1995 na cidade de Dublin em Ohio nos Estados Unidos da América. Esse

workshop foi motivado pelos seguintes fatos: dificuldade na indexação de recursos na Internet; uso de registros resultantes de padrões para catalogação consumia muito tempo para serem criados e atualizados; na medida que o escopo da indexação era ampliado por ferramentas de busca, muita informação era recuperada. Nessa conferência foi sugerido um conjunto de elementos de metadados para descrever recursos em redes de computadores como a Internet (DCMI, 2015; WEIBEL, 1995).

Metadados *Dublin Core* podem ser representados (codificados) em formatos variados. Por exemplo, formatos texto (JOHNSTON, 2007), X/HTML (JOHNSTON, 2008), XML (POWELL, 2003; JOHNSTON; POWELL, 2008) e RDF (NILSSON et al., 2008). No contexto do *Dublin Core*, propriedades são usadas para descrever recursos. Classes são grupos de recursos com propriedades em comum. Tipos de dados (*datatype*) especificam como valores são estruturados. Esquemas de codificação de vocabulários (*vocabulary encoding scheme*) identificam vocabulários controlados cujos termos podem ser usados como valores (RÜHLE; BAKER; JOHNSTON, 2011). *Contributor, Coverage, Creator, Date, Description, Format, Identifier, Language, Publisher, Relation, Rights, Source, Subject, Title* e *Type* são propriedades no *Dublin Core Metadata Element Set* (DCMI, 2012b). *Name, Label, URI, Definition* e *Type of Term* são atributos de propriedades. A título de ilustração, a figura 27 apresenta a definição de uma propriedade por meio de atributos. Um modelo abstrato que define componentes em metadados *Dublin Core* é encontrado em Powell et al. (2007). Esse modelo é composto por *DCMI Resource Model, DCMI Description Set Model* e *DCMI Vocabulary Model*. Esses modelos são descritos em diagramas na linguagem de modelagem UML. Em Nilsson (2008b) é descrito um arcabouço que define componentes de perfis de aplicação (*application profile*) *Dublin Core*. Um *Dublin Core Application Profile* (DCAP) especifica e descreve metadados usados em aplicações, descreve o que a comunidade pretende com a aplicação, caracteriza o que é descrito por metadados, enumera termos a serem usados e regras para uso dos mesmos, e define sintaxe para codificar dados (COYLE, 2009).

6.12.2 IEEE *Learning Object Metadata*

IEEE (2002) é uma norma para metadados que visa facilitar localização, avaliação, aquisição, troca, compartilhamento e uso de objetos de aprendizagem por estudantes, instrutores e processos automatizados de software. Um objeto de aprendizagem é uma entidade, digital ou não, usada para aprendizagem, educação ou treinamento. Em IEEE (2002), os elementos que descrevem objetos são agrupados nas seguintes categorias: genérica, ciclo de vida, meta-metadados, técnica, educacional, direitos, relação, anotação e classificação. A categoria genérica agrupa informação genérica sobre o objeto como um todo. A categoria ciclo de vida agrupa informação sobre a história e estado atual do objeto. A categoria meta-metadados agrupa informação sobre a instância de metadados. A

Figura 27 – Exemplo de definição de propriedade

Term Name: contributor	
URI:	http://purl.org/dc/terms/contributor
Label:	Contributor
Definition:	An entity responsible for making contributions to the resource.
Comment:	Examples of a Contributor include a person, an organization, or a service.
Type of Term:	<u>Property</u>
Refines:	http://purl.org/dc/elements/1.1/contributor
Has Range:	http://purl.org/dc/terms/Agent
Version:	http://dublincore.org/usage/terms/history/#contributorT-001

Fonte: [DCMI \(2012a\)](#)

categoria técnica agrupa informação sobre requisitos e características técnicas do objeto. A categoria educacional agrupa informação sobre características educacionais e pedagógicas do objeto. A categoria direitos agrupa informação sobre direitos de propriedade e condições de uso do objeto. A categoria relação agrupa informação sobre relações entre objetos. A categoria anotação provê comentários sobre o uso educacional do objeto e informação sobre as autorias e datas dos comentários. Finalmente, a categoria classificação descreve o objeto de acordo com um sistema de classificação particular. Em conjunto, essas categorias compõem o esquema LOMv1.0. Para cada elemento em LOMv1.0 são definidos nome, explicação, tamanho, ordem e exemplo.

7 Elementos de modelagem

Este capítulo apresenta uma visão geral sobre elementos de modelagem. Este capítulo contém definições do termo modelo e informação sobre propósitos de modelos, usos de modelos, processo de modelagem, princípios de modelagem e análise de domínio. Considerando a abrangência do assunto abordado neste capítulo, este capítulo propositalmente não inclui todos os resultados da pesquisa bibliográfica realizada sobre elementos de modelagem. Entretanto, inclui referências para fontes de informação que podem ser consultadas por interessados nos assuntos abordados. Por fim, vale destacar que a fonte [Albuquerque \(2014\)](#) contém informação complementar sobre modelagem no contexto da Arquitetura da Informação.

7.1 Definições de modelo

Segundo [ISO \(2011e\)](#), o termo “modelo” é usado de forma distinta por diferentes comunidades. Nessa norma, “M é um modelo de S, se M pode ser usado para responder questões sobre S”. Segundo [IEEE \(2012b\)](#), o termo modelo designa representação de algo, representação essa que suprime certos aspectos do que é modelado. Segundo [Guizzardi \(2005\)](#), modelo é uma abstração de uma dada porção da realidade, articulada de acordo com conceituação de um domínio. O termo conceituação também tem diversas definições. Por exemplo, as seguintes: visão simplificada e abstrata do mundo que se deseja representar para determinado propósito ([GRUBER, 1993](#)); interpretação estruturada de parte do mundo, usada por pessoas para pensar e comunicar sobre o mundo ([BORST, 1997](#)); conjunto de conceitos usado para articular abstrações de estado de coisas em um dado domínio ([GUIZZARDI, 2005](#)); modelo abstrato que possibilita descrever algo relevante do mundo ([CORCHO; POVEDA-VILLALÓN; GÓMEZ-PÉREZ, 2015](#)). Nesse contexto, portanto modelo e conceituação designam entidades abstratas. O termo “modelo conceitual” é também usado em diversas fontes de informação. Em [IEEE \(2012b\)](#), esse termo é definido como representação composta por conceitos relevantes a um empreendimento, que suprime certos aspectos do que é modelado. Finalmente, o termo “modelo de domínio” designa modelo conceitual que engloba definições de funções, objetos e relações em um domínio ([KANG et al., 1990](#)). Nesse contexto, domínio é a área de conhecimento sobre a qual o modelo de domínio é construído. Um modelo de domínio geralmente resulta de análise de domínio.

7.2 Propósitos e usos de modelos

Modelos podem ter propósitos e usos variados. Por exemplo, em ciclo de vida de sistema de informação, os modelos podem ter os seguintes propósitos e usos: promover comunicação entre desenvolvedores e usuários do sistema, facilitar entendimento de domínio, documentação de requisitos, suporte a processo de aquisição e desenho (*design*) de sistema (MOODY, 2005; WAND; WEBER, 2002). Durante o ciclo de desenvolvimento de um sistema de informação, modelos podem ser usados em disciplinas como requisitos, desenho (*design*), implementação, manutenção e teste.

7.3 Modelagem

Nesta tese, o termo modelagem designa processo de representação por meio de modelo. O termo “modelagem conceitual” é encontrado em diversas fontes. Segundo Mylopoulos (1992), modelagem conceitual é descrever formalmente aspectos do mundo físico e social em torno de nós, com propósitos de entendimento e comunicação. Modelagem de domínio, no contexto desta tese, designa processo de representação por meio de modelo de domínio. A modelagem de domínio pode ser realizada por meio da execução de atividades integrantes de processo de análise de domínio. Essas atividades podem ser prescritas, por exemplo, em método de análise de domínio.

7.3.1 Princípios de modelagem

A fonte IEEE (2014c) sugere os seguintes princípios para o processo de modelagem: modelar o essencial (a modelagem deve abstrair informação não essencial); prover perspectiva (a modelagem deve organizar informação com o objetivo de focar a modelagem em aspectos relevantes a determinadas perspectivas, por meio de notação, vocabulário, ferramentas e métodos apropriados); e habilitar comunicação efetiva (a modelagem deve resultar em modelos que facilitem a comunicação da informação a interessados). Embora a fonte IEEE (2014c) enfoque processo de modelagem de software, os princípios anteriormente listados são também relevantes em outros contextos.

7.4 Análise de domínio

Existem várias definições do termo “análise de domínio”, por exemplo: método para entender necessidades de informação e práticas em domínios e comunidades (ABBAS, 2010); definição da base de conhecimento de uma comunidade (SMIRAGLIA, 2012); processo por meio do qual a informação usada no desenvolvimento de software é identificada, capturada e organizada com o propósito de torná-la reusável na criação de novos sistemas (PIETO-DIAZ, 1990); processo de identificação, coleta, organização e representação da

informação relevante em um domínio, baseado no estudo de sistemas existentes e de suas histórias de desenvolvimento, no conhecimento capturado de especialistas no domínio, em teorias inerentes e tecnologias emergentes no domínio (KANG et al., 1990); processo que objetiva definir estrutura de domínio e requisitos, e capturá-los em modelo de domínio (HOLIBAUGH, 1993). Nesta tese, como em Albrechtsen (1992), o termo “análise de domínio” designa processo em que domínio é analisado para construir modelo de domínio.

No contexto da Ciência da Informação, Hjørland e Albrechtsen (1995) sugerem a análise de domínio como um paradigma, como um arcabouço metodológico. Segundo Hjørland (2008b), análise de domínio é abordagem teórica para organização do conhecimento com as seguintes pressuposições: indexação deve refletir necessidades de dado grupo de usuários ou propósito; qualquer descrição ou representação de recurso de informação pode ser mais ou menos adequada a determinadas tarefas; uma descrição nunca é objetiva e neutra; o objetivo não é padronizar descrições ou criar uma só descrição para diferentes grupos alvo; diferentes pontos de vista necessitam de diferentes sistemas de organização; a representação de recurso de informação visa possibilitar que os usuários façam discriminações relevantes; recursos de informação devem ser vistos com os olhos dos seus usuários. Em Hjørland (2002) são sugeridas abordagens para análise de domínio, por exemplo, as seguintes: produção de guias de literatura, classificações e tesouros especializados; realização de pesquisas sobre indexação e recuperação da informação em domínios especializados; estudos empíricos de usuários em diferentes campos; estudos bibliométricos e históricos; estudos sobre estruturas e tipos de documentos; estudos epistemológicos e críticos; estudos terminológicos; linguagens para propósitos especiais; estudos de discurso; estudos de estruturas e de instituições em comunicação científica; análise de cognição profissional.

Em Ciência da Informação, podem ser usados diversos métodos, teorias e ferramentas para análise de domínio. Alguns exemplos: análise bibliométrica, análise de citações, análise de cocitações, análise conceitual, análise de conteúdo, análise de discurso, análise de facetas, análise de rede social, análise terminológica, *co-word analysis*, *Cognitive Work Analysis* (CWA), entrevista, estudo de caso, *Informetric analysis*, mapa de conteúdos, mapa de tópicos *Pathfinder network*, revisão de literatura e *Work task oriented methodology* (WOM) (SMIRAGLIA, 2015). Em outras disciplinas, a análise de domínio é também relevante. Por exemplo, são propósitos da análise de domínio no contexto da Engenharia de Software: reduzir custos de adaptação de software; prover suporte a reuso de software, processo de software, tecnologia de software e experiência em software. Análise de domínio é particularmente relevante ao reuso em desenvolvimento de software (FERRÉ; VEGAS, 1999; JATAIN; GOEL, 2009). Finalmente, em Valerio, Succi e Fenaroli (1997) são relacionadas a seguintes fases da análise de domínio: caracterização de domínio, planejamento de projeto, análise de dados e modelagem de domínio.

7.4.1 Métodos de análise de domínio

Existem diversos métodos de análise de domínio. A seguir, são relacionadas atividades nesses métodos: caracterizar domínio, coletar dados, analisar dados, classificar e avaliar modelo de domínio (FERRÉ; VEGAS, 1999; JATAIN; GOEL, 2009). Jatain e Goel (2009) e Jatain (2009) comparam métodos de análise de domínio usados em Engenharia de Software, tais como, *Feature Oriented Domain Analysis Method* e *Joint Object Oriented Domain Analysis*.

7.4.1.1 *Feature Oriented Domain Analysis Method*

O *Feature Oriented Domain Analysis Method* engloba: análise de contexto, análise e modelagem de domínio, e modelagem de arquitetura. A fase análise e modelagem de domínio é a mais relevante em termos de conceituação. Nela são analisadas e modeladas similaridades e diferenças no domínio. Nessa fase, são encontrados os processos análise de características, construção de modelo de entidade relacionamento (MER) e análise funcional. O processo análise de características engloba coleta de documentos; identificação, abstração, classificação e definição de características; e validação de modelo. Nesse contexto, característica (*feature*) é atributo de sistema que afeta diretamente os usuários finais. Esse processo visa construir modelo que represente entendimento sobre características de aplicações no domínio. Esse modelo é construído por meio de classificação e estruturação de características, indicação de características opcionais e alternativas, definições de características e regras de composição de características. O processo de construção de modelo de entidade relacionamento tem o objetivo de construir modelo que represente conhecimento sobre o domínio por meio de entidades e relações no domínio. Os elementos principais nesse modelo são classes que representam abstrações de objetos no domínio e relações que representam generalizações e agregações. Finalmente, o processo de análise funcional tem o objetivo de identificar aspectos comuns e diferenças nas funcionalidades providas por aplicações no domínio sendo analisado (KANG et al., 1990).

7.4.1.2 *Joint Object Oriented Domain Analysis*

O método *Joint Object Oriented Domain Analysis* é embasado em técnicas e notação de análise orientada a objetos (*object oriented analysis*) propostas em Coad e Yourdon (1990). Esse método tem o objetivo de definir modelo de domínio a ser usado em desenvolvimento de componentes de software reusáveis. Nesse método, as atividades são agrupadas nas seguintes fases: preparação de domínio, definição de domínio e modelagem de domínio. A fase responsável pela preparação de domínio tem o objetivo de adquirir conhecimento sobre o domínio, identificar e coletar fontes de informação. Na fase responsável pela definição de domínio, são encontradas atividades para definição de contexto do domínio, análise de artefatos de sistemas, identificação de estrutura do

domínio, análise de cenários, análise de dependências externas, documentação de definição de domínio, revisão e atualização de documentação. Finalmente, a fase responsável pela modelagem de domínio engloba atividades com os seguintes objetivos: exame da história de vida de objetos e respostas a eventos, identificação e percorrimto de cenários, abstração e agrupamento de objetos, revisão e atualização de modelo. O modelo resultante inclui definições de classes, diagramas de estrutura do domínio, diagramas de assunto e diagramas de cenários ([HOLIBAUGH, 1993](#)).

8 Elementos de qualidade

Este capítulo tem o objetivo de apresentar uma visão abrangente sobre o assunto qualidade, e contém definições do conceito designado pelo termo “qualidade” e de conceitos correlatos, informação sobre qualidade da informação, qualidade de modelo conceitual, qualidade de vocabulário controlado, avaliação da qualidade de vocabulário controlado, características de qualidade, medidas de qualidade e modelo de qualidade.

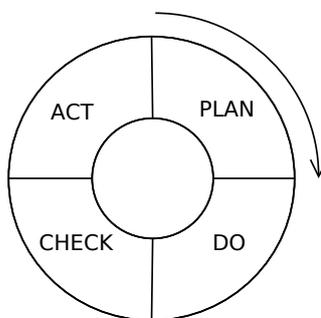
8.1 Definições de qualidade e de conceitos correlatos

Há diversas definições propostas para o conceito designado pelo termo qualidade (BIZER, 2007). Essas definições variam entre pessoas e entre áreas de conhecimento (ASQ, 2017). Segundo Joseph M. Juran, qualidade é adequação ao uso (*fitness for use*) (HOYER; HOYER, 2001). Hoyer e Hoyer (2001) contrastam definições de qualidade e destacam a proposta por Walter Shewhart, na qual são encontrados os seguintes princípios: qualidade tem dimensão subjetiva, o que o cliente deseja; qualidade tem dimensão objetiva (propriedades do produto) independente do que o cliente deseja; uma dimensão da qualidade é o valor recebido pelo preço pago; padrões de qualidade devem ser expressos por características físicas e quantitativamente mensuráveis de produto. No contexto técnico, o termo qualidade pode ser definido como habilidade de produto, serviço, sistema, componente ou processo atender necessidades, expectativas ou requisitos de cliente ou usuário (ISO, 2010); grau com o qual um conjunto de características inerentes satisfaz requisitos (PMI, 2013); produto ou serviço livre de deficiências (ASQ, 2017). Por sua vez, garantia de qualidade pode ser definida como padrão planejado e sistemático de todas as ações necessárias para prover confiança adequada que um item ou um produto atende a requisitos técnicos estabelecidos (ISO, 2010), ou conjunto de atividades definidas para avaliar processo pelo qual produtos são desenvolvidos e fabricados (ISO, 2010). Portanto, garantia de qualidade engloba atividades sistemáticas e planejadas implementadas em sistema de qualidade, por meio das quais se provê confiança que produto ou serviço irá satisfazer aos requisitos de qualidade (ASQ, 2017). Controle de qualidade é um conjunto de atividades desenhadas (*designed*) para avaliar qualidade de produtos desenvolvidos ou fabricados (ISO, 2010). Segundo ASQ (2017), os termos “garantia de qualidade” e “controle de qualidade” frequentemente designam ações para garantir qualidade de produto, serviço ou processo. Por fim, o termo “gestão da qualidade” designa sistema de gestão para alcançar a maior satisfação do cliente ao menor custo para a organização, enquanto melhora continuamente processos. Nesse sistema, estrutura, responsabilidades e procedimentos são documentados para alcançar gestão efetiva da qualidade (ASQ, 2017).

8.1.1 *Plan-Do-Check-Act*

No contexto da gestão da qualidade, é importante destacar o ciclo denominado *Plan-Do-Check-Act* (PDCA) como um meio para promoção da melhoria da qualidade. O ciclo PDCA é composto pelas fases Planejar (*Plan*), Executar (*Do*), Conferir (*Check*) e Agir (*Act*). Na fase Planejar, ocorre planejamento para entrega dos resultados. Na fase Executar, o plano é executado. Na fase Conferir, é comparado resultado da fase anterior com o que havia sido anteriormente planejado. Finalmente, na fase Agir, são executadas atividades corretivas para alcançar mudanças desejadas. A figura 28 apresenta as fases do ciclo PDCA. Em projeto, esse ciclo pode ser repetido várias vezes e pode ser usado em diferentes contextos (ASQ, 2017).

Figura 28 – Fases no ciclo *Plan-Do-Check-Act*



Fonte: Adaptado de ASQ (2017)

8.2 Qualidade da informação

A importância da qualidade da informação para a qualidade de sistemas de informação é reconhecida há anos (vide Ivanov (1972)). Nas organizações, informação é um importante ativo e pode ser caro lidar com problemas decorrentes de falta de qualidade da informação (BIZER, 2007). English (2009) e Wang, Pierce e Madnick (2014) apresentam uma relação de casos que ilustram custos e consequências decorrentes de problemas provocados por baixa qualidade da informação ou por baixa qualidade de sistemas de informação. Segundo Miller (1996), informação de qualidade é informação que atende a determinados critérios. Identificar a qualidade da informação engloba identificar atributos de qualidade da informação relevantes e determinar como eles afetam o usuário da informação. Finalmente, quanto aos custos decorrentes de baixa qualidade da informação, Wang, Pierce e Madnick (2014) observam que são difíceis de quantificar, pois envolvem componentes tangíveis e intangíveis.

Pesquisas sobre qualidade da informação são realizadas há anos. Há diversas propostas de conceituação do termo “qualidade da informação” e diversos métodos para

avaliação e melhoria da qualidade da informação (BIZER, 2007; CALAZANS, 2008). Em Ivanov (1972), conceituações de qualidade da informação são organizadas de acordo com contextos onde são propostas: planejamento e desenvolvimento de sistemas de informação, estudo de custo e de valor da informação, estudo de desenvolvimento de produto de informação corporativo, desenvolvimento e preparação de instalação de sistema de informação corporativo, auditoria de sistema de processamento de dados, uso de sistema de gerenciamento de dados, economia da informação, análise de erros e de suas consequências em sistema de controle, desenho (*design*) e uso de banco de dados. Eppler e Wittig (2000) relacionam as seguintes abordagens na definição de qualidade da informação: qualidade da informação definida como informação adequada ao uso (*fit for use*) por consumidores de informação; qualidade da informação é a característica da informação atender ou exceder expectativas de cliente; informação de qualidade é informação que atende a especificações ou a requisitos; qualidade da informação é a característica da informação ser de alto valor para os seus usuários; grau com o qual informação tem características de conteúdo, forma e tempo que lhe dão valor a usuários finais específicos; qualidade da informação é a diferença entre informação requerida determinada por objetivo e informação obtida; qualidade da informação é a característica da informação atender requisitos funcionais, técnicos, cognitivos e estéticos de produtores, administrações, consumidores e especialistas de informação.

8.3 Qualidade de modelo conceitual

Embasado na norma ISO 9000, Moody (2005) define qualidade de modelo conceitual como “totalidade de aspectos e características de um modelo conceitual que influencia a sua habilidade de satisfazer necessidades declaradas ou implícitas”. Segundo IEEE (2014c), os níveis de qualidade de modelos podem ser caracterizados, por exemplo, por níveis de completude, consistência e correção. Lindland, Sindre e Sølvsberg (1994) sugerem os seguintes tipos de qualidade de modelo: sintática, semântica e pragmática. A qualidade sintática é relativa a quão bem o modelo corresponde à linguagem; a qualidade semântica é relativa a quão bem o modelo corresponde ao domínio; a qualidade pragmática é relativa a quão bem o modelo corresponde à interpretação da sua audiência. Por fim, segundo Piattini et al. (2005), qualidade é conceito com múltiplas dimensões. Para análise da qualidade de modelos conceituais, esses autores sugerem que seja feita distinção entre as dimensões tipo de qualidade, objeto de estudo, objetivo de pesquisa, característica de qualidade, método de pesquisa e tipo de proposta.

Moody (2005) destaca que existe relação entre a qualidade de modelos conceituais e a qualidade de sistemas de informação. Também destaca que o desenvolvimento de modelos conceituais de melhor qualidade contribui para o desenvolvimento de sistemas de informação de melhor qualidade. Ainda segundo Moody (2005), a qualidade de modelo

conceitual pode, por exemplo, afetar tempo, custo, esforço e qualidade de resultados de processo de desenvolvimento de sistema de informação. Há diversas fontes de informação sobre atributos de qualidade de modelos conceituais. Em [Moody \(2005\)](#) e em [Piattini et al. \(2005\)](#), são identificadas várias delas. [Moody \(2005\)](#) relaciona as seguintes dificuldades no processo de avaliação da qualidade de modelo conceitual: falta de guias amplamente aceitas, pouco acordo sobre o que consiste um bom modelo, falta de padrões para avaliação da qualidade de modelos conceituais, falta de consenso sobre como avaliar a qualidade de modelos conceituais. [Shanks e Darke \(1997\)](#) sugerem que a garantia da qualidade em modelagem conceitual deve englobar o processo de modelagem e o modelo resultante. Finalmente, [Moody \(2005\)](#) diferencia entre qualidade de produto e qualidade de processo, e destaca a importância da qualidade de processo.

Para caracterizar pesquisas sobre qualidade de modelos conceituais, a abordagem descrita em [Piattini et al. \(2005\)](#) sugere que sejam analisadas as seguintes dimensões: classe de qualidade, objeto de estudo, objetivo de pesquisa, característica de qualidade, contexto de estudo, método de pesquisa e tipo de proposta. As classes de qualidade sugeridas são: qualidade física, qualidade sintática, qualidade semântica, qualidade semântica percebida, qualidade pragmática, qualidade social, qualidade perceptiva (*perceptual quality*), qualidade descritiva e qualidade inferencial (*inferential quality*). Quanto ao objeto de estudo, são sugeridos qualidade de processo de modelagem conceitual, qualidade de linguagem conceitual e qualidade de modelo conceitual. Os objetivos da pesquisa, podem ser variados: entendimento da qualidade, medição da qualidade, avaliação da qualidade, garantia da qualidade e melhoria da qualidade. Quanto às outras dimensões na abordagem, são também diversas.

8.4 Qualidade de vocabulário controlado

A qualidade de vocabulário controlado é importante ao sucesso de sistema de informação onde o vocabulário é usado. Defeitos em vocabulário controlado podem resultar, por exemplo, em dificuldade de leitura por ser humano ou em dificuldade na realização de inferência por máquina. Nesse contexto, defeito é uma imperfeição ou deficiência em produto onde esse produto não atende aos requisitos ou especificações e precisa ser reparado ou substituído ([PMI, 2013](#)). São possíveis defeitos em vocabulários controlados: conceito não documentado, conflito entre termos, conceito órfão (conceito sem relação com outro conceito), relação hierárquica cíclica, relação associativa sem valor, omissão de conceito, falta de termos em vocabulário controlado multilíngue e cobertura incompleta do escopo ([BLUMAUER, 2016](#)). Há diversas recomendações para qualidade de vocabulários controlados em normas, guias de melhores práticas, metodologias e teorias. Por exemplo: [Aitchison, Bawden e Gilchrist \(2000\)](#), [Working Group on Guidelines for Multilingual Thesauri \(2009\)](#), [Harpring \(2010\)](#), [Hedden \(2010\)](#), [ISO \(2011b\)](#), [NISO \(2005\)](#), [Soergel](#)

(2002) e Schulz et al. (2012).

8.4.1 Avaliação da qualidade de vocabulário controlado

Para garantir a qualidade de um produto, avaliar o produto é importante. Avaliar consiste em determinar sistematicamente o grau com o qual certo critério é atendido (IEEE, 2016). Em Neuhaus et al. (2013), é sugerida definição de avaliação de ontologia, que, nesta tese, será generalizada para outras classes de vocabulários controlados. Generalizando a definição nessa fonte, avaliação de vocabulário controlado é processo que engloba atividades executadas com o objetivo de coletar informação sobre propriedades de vocabulário controlado, comparar resultados com conjunto de requisitos e aferir a adequação do vocabulário controlado ao propósito especificado. Existem várias abordagens para avaliação da qualidade de vocabulário controlado, por exemplo, as seguintes: avaliação heurística, onde especialista ou grupo de especialistas avalia o vocabulário controlado; modelagem de afinidade, onde usuários agrupam termos do vocabulário controlado, atribuem graus de similaridade a termos do vocabulário controlado e resultados são contrastados com a estrutura proposta para o vocabulário controlado; e teste de usabilidade, abordagem holística onde são usados guias e métodos geralmente originados na área denominada Interação Humano-Computador (IHC). A avaliação heurística pode ser informal e qualitativa (especialistas compartilham suas reações e opiniões) ou formal e quantitativa (especialistas avaliam o vocabulário controlado segundo lista de critérios) (NISO, 2005). A avaliação de vocabulário controlado pode seguir abordagem intrínseca ou extrínseca. Na intrínseca, o objeto de análise é o vocabulário controlado. Na extrínseca, o objeto de análise é o comportamento do vocabulário controlado em processo de indexação ou de recuperação da informação. Quanto ao momento da avaliação, pode ser durante o desenvolvimento do vocabulário controlado ou durante o seu uso. Por fim, a avaliação pode focar a qualidade da informação ou a do produto (DUQUE-RAMOS et al., 2011; MARTÍNEZ et al., 2011; NISO, 2005; STVILIA, 2007).

A avaliação de vocabulário controlado estruturado como taxonomia ou tesauro pode focar, por exemplo, aspectos estruturais, aspectos formais, validade semântica e adequação ao contexto de aplicação (BERMEJO; RUBIO; ROJO, 1989). Acerca da avaliação de ontologia, Neuhaus et al. (2013) destacam a sua importância, e sugerem que seja incorporada ao ciclo de vida da ontologia e realizada considerando-se requisitos cuidadosamente identificados. Na avaliação de ontologia, diversos métodos e perspectivas podem ser adotados (BRANK; GROBELNIK; MLADENIĆ, 2005, 2005; HLOMANI; STACEY, 2014; HARTMANN et al., 2005; RAAD; CRUZ, 2015). A avaliação pode focar ferramenta, linguagem, conteúdo, metodologia, custo, sintaxe, vocabulário, estrutura ou estatísticas de uso (LOZANO-TELLO; GÓMEZ-PÉREZ, 2004; STRASUNSKAS; TOMASSEN, 2008). Gavrilova, Gorovoy e Bolotnikova (2010) sugerem que os métodos de

avaliação de ontologias podem ser classificados considerando-se propósito, objeto de análise, meio de análise, grau de automação e estágio de aplicação. Segundo [Brank, Grobelnik e Mladenić \(2005\)](#), os métodos de avaliação frequentemente adotam uma das seguintes abordagens: comparação da ontologia a um padrão (*golden standard*), que pode ser uma ontologia; aplicação da ontologia e avaliação de resultados; comparação com fontes de dados sobre o domínio da ontologia; pessoas avaliam até que ponto a ontologia atende a critérios, padrões, requisitos etc. Em [Stvilia \(2007\)](#), são relacionadas as seguintes alternativas de avaliação: uso de informação sobre grafo de classes (vide exemplos em [Kang et al. \(2004\)](#) e [Gavrilova, Gorovoy e Bolotnikova \(2010\)](#)), análise de estrutura léxica ou linguística, avaliação externa por ferramentas ou usuários, e combinação dessas abordagens.

Considerando que a estrutura da ontologia pode ser complexa, [Brank, Grobelnik e Mladenić \(2005\)](#) sugerem que diferentes níveis da ontologia sejam enfocados, em vez da ontologia como um todo. Sugerem os seguintes níveis: léxico, vocabulário ou camada de dados; hierarquia ou taxonomia; outras relações semânticas; contexto ou nível de aplicação; nível sintático; estrutura, arquitetura, desenho (*design*). Segundo [Brank, Grobelnik e Mladenić \(2005\)](#), não existe uma abordagem de avaliação de ontologia que seja a melhor ou a preferida. A escolha da abordagem depende, por exemplo, do propósito da avaliação, do propósito da aplicação onde a ontologia é usada e do aspecto avaliado. Segundo [Neuhaus et al. \(2013\)](#), falta consenso sobre como avaliar ontologias. Além disso, técnicas e ferramentas de avaliação não são amplamente usadas, o que pode resultar em ontologias de pouca qualidade. Segundo [Duque-Ramos et al. \(2011\)](#), a avaliação de ontologia engloba posicionamento (*ranking*), seleção e comparação. Segundo essa fonte, esses aspectos têm sido abordados por meios diversos, sem que nenhum deles tenha se tornado padrão e há necessidade de métodos de avaliação padronizados. Finalmente, é importante destacar que existem diversas ferramentas para a avaliação de ontologias, tais como *ODEclean*, *ODEval*, *OntoCheck*, *MonKi* e *Ontology Pitfall Scanner! (OOPS!)* ([POVEDA-VILLALÓN; GÓMEZ-PÉREZ; SUÁREZ-FIGUEROA, 2014](#)).

8.5 Características e medidas de qualidade

O termo “característica de qualidade” pode ser definido como fator, elemento ou medida que define e diferencia processo, função, produto, serviço ou outra entidade ([ASQ, 2017](#)). Na garantia e no controle da qualidade, medições e medidas são relevantes. Medições podem ser realizadas sobre produto, processo, recurso (item usado por processo) ([BASILI; CALDIERA; ROMBACH, 1994](#)). O termo medição pode ser definido como processo de comparação quantitativa entre resultados e requisitos ([ASQ, 2017](#)). [Archer e Stinson \(1995\)](#) definem medida como valor numérico calculado a partir de dados de entrada e observam que o termo métrica é usado em certos contextos. Segundo [ASQ \(2017\)](#), métrica é padrão para medição. Para [ISO \(2010\)](#), métrica é medida quantitativa do grau com o qual um

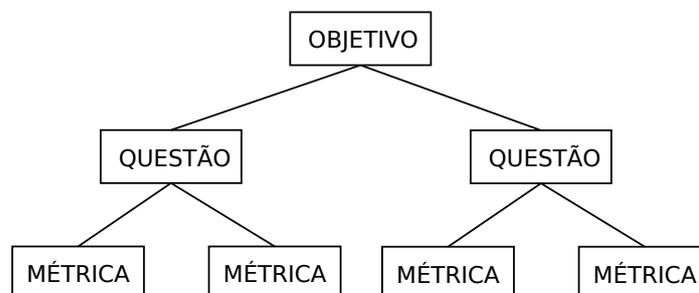
sistema, componente, ou processo possui um dado atributo. Por fim, em [PMI \(2013\)](#), o termo “métrica de qualidade” é definido como “descrição de atributo de projeto ou de produto e como o medir”.

[Archer e Stinson \(1995\)](#) relacionam as seguintes características como relevantes para que medidas sejam úteis: medida é robusta (o cálculo da medida poder ser repetido e o resultado não é sensível a pequenas mudanças no ambiente, ferramenta ou observador); medida é precisa; processo de coleta de dados para medida é objetivo; escala para realizar comparações entre medidas de mesmo tipo; e medida é significativa. [Archer e Stinson \(1995\)](#) também sugerem que existem situações nas quais é necessário um grupo de medidas para caracterizar aquilo que é medido. [Orr \(1979\)](#) relaciona os seguintes critérios a serem considerados na decisão sobre adequação de determinada medida a uma aplicação e sobre determinada medida ser preferível a outras medidas: adequação, informatividade, validade, reprodutibilidade, comparabilidade e praticidade.

8.5.1 Goal Question Metrics

Existem métodos que podem ser usados no planejamento de medições, por exemplo, o método *Goal Question Metrics* (GQM), descrito em [Basili, Caldiera e Rombach \(1994\)](#) e [Solingen e Berghout \(1999\)](#). Segundo esse método, para que uma medida tenha propósito, é necessário definir os objetivos a alcançar. O GQM engloba os níveis conceitual (objetivo), operacional (questão) e quantitativo. O GQM inclui atividades responsáveis por identificar objetivos, elaborar questões que definam objetivos de medições e especificar medições. A figura 29 apresenta elementos do GQM. Para cada objetivo, é registrada a seguinte informação: questão, objeto (produto, processo, recurso), ponto de vista e propósito. Essa informação pode ser registrada em formulário que apresente estrutura similar àquela no quadro 8.

Figura 29 – Elementos do GQM



Fonte: Adaptado de [Basili, Caldiera e Rombach \(1994\)](#)

Quadro 8 – Exemplo de formulário

Analisar	Vocabulário controlado
Para o propósito de	Medição
Com respeito à	Estrutura
Do ponto de vista do	Pesquisador
No contexto de	Pesquisa sobre qualidade de vocabulário controlado

Fonte: Adaptado de [Basili, Caldiera e Rombach \(1994\)](#)

8.5.2 Características e medidas de qualidade da informação

Em [Arouck \(2011\)](#), [Lee et al. \(2002\)](#), [Miller \(1996\)](#) e [Wand e Wang \(1996\)](#), são propostas definições de dimensões e características de qualidade da informação. No quadro 9, são apresentados termos que designam diversas características de qualidade da informação. Com o objetivo de minimizar mal-entendidos, nesse quadro, são apresentados os termos em inglês e português. Os termos nesse quadro foram identificados em [Wand e Wang \(1996\)](#) e a tradução segue [Arouck \(2011\)](#) ou fontes sobre o tema qualidade. [Eppler e Wittig \(2000\)](#) relacionam dimensões e características de qualidade da informação identificadas em arcabouços de qualidade (*quality framework*). Segundo [Wand e Wang \(1996\)](#), não há consenso sobre o conjunto de características de qualidade da informação e de quais são as definições mais apropriadas para essas características. [Wang, Pierce e Madnick \(2014\)](#) consideram que, embora existam diversas definições para características de qualidade da informação, existem dificuldades para se obter definições rigorosas. [Miller \(1996\)](#) observam que as características de qualidade da informação dependem das perspectivas dos usuários e que os usuários mudam suas necessidades de qualidade da informação com o passar do tempo. Por fim, vale destacar que, além do termo “característica de qualidade”, os termos “dimensão de qualidade” e “critério de qualidade” são usados em algumas fontes sobre qualidade da informação.

Quadro 9 – Características de qualidade da informação

TERMOS EM INGLÊS	TERMOS EM PORTUGUÊS	TERMOS EM INGLÊS	TERMOS EM PORTUGUÊS
Accuracy	Acurácia	Level of detail	Pormenorização
Comparability	Comparabilidade	Precision	Precisão
Completeness	Completeness	Quantitativeness	Mensurabilidade
Conciseness	Concisão	Relevance	Relevância
Consistency	Coerência	Reliability	Confiabilidade
Currency	Atualidade	Scope	Escopo
Flexibility	Flexibilidade	Timeliness	Oportunidade
Freedom from bias	Livre de preconceitos	Understandability	Compreensibilidade
Informativeness	Valor informativo		

Fonte: Elaborado pelo autor

Em fontes de informação onde são propostas definições de características de qualidade da informação, as características são frequentemente agrupadas em classes. Em algumas fontes, o termo “dimensão de qualidade” designa classe de características relacionadas. Em Wang e Strong (1996) é proposto arcabouço onde características de qualidade da informação são agrupadas nas seguintes classes: intrínseca, contextual, representacional e acessibilidade. A classe intrínseca engloba características de qualidade que dependem da própria informação. A classe contextual engloba características de qualidade que dependem de aspectos como preferências de usuários. A classe representacional engloba características relacionadas à representação da informação. A classe acessibilidade engloba características relacionadas ao acesso à informação. Em Stvilia et al. (2007), é proposto arcabouço onde as características de qualidade da informação são agrupadas nas classes intrínseca, relacional/contextual e reputacional. Em Wand e Wang (1996), as características de qualidade são intrínsecas à informação ou relacionadas ao sistema de informação. Segundo Naumann e Rolker (2000), a qualidade da informação depende da percepção do usuário, da própria informação e do processo de acesso à informação. Existem características de qualidade que são determinadas por usuários, pois dependem de suas experiências. Essas características são classificadas como critério de sujeito, por exemplo, compreensibilidade. Por sua vez, existem características de qualidade que são determinadas por análise da informação. Essas características são classificadas como critério de objeto, por exemplo, completude. Finalmente, existem características de qualidade que são determinadas por processo de acesso à informação. Nessa fonte, essas características são classificadas como critério de processo, por exemplo, disponibilidade e tempo de resposta.

Na literatura, existem diversas propostas de medidas de qualidade da informação. No arcabouço descrito em Stvilia et al. (2007), são propostas medidas como número de instâncias com diferentes formatos de um mesmo elemento. Wang, Pierce e Madnick (2014) observam que medir qualidade da informação não é simples. Uma vez que qualidade da informação tem natureza multidimensional, geralmente são necessárias diversas medidas para se caracterizar a qualidade da informação. Finalmente, Naumann e Rolker (2000) relacionam os seguintes aspectos a serem considerados na atribuição de valores a características de qualidade da informação: precisão, praticidade, unidade de medida e faixa de medida (*range*).

8.5.3 Características e medidas de qualidade de modelo conceitual

As características e medidas de qualidade de modelo conceitual variam entre domínios de aplicação, assim como variam entre tipos de modelos conceituais. Por exemplo, no contexto de modelos conceituais de software, são propostas diversas características e medidas de qualidade. Em Piattini et al. (2005), são propostas medidas de qualidade de completude de modelo conceitual, caso de uso (*use case*), diagrama de classe e dia-

grama de estado representados na linguagem de modelagem UML, modelos conceituais de *datawarehouse* e modelos de processos de software.

8.5.4 Características e medidas de qualidade de vocabulário controlado

Existem diversas propostas de características de qualidade de vocabulário controlado. Algumas características de qualidade dependem da percepção do usuário. Outras são inerentes à informação. Algumas características de qualidade propostas podem ser medidas independentemente do uso do vocabulário controlado. Outras dependem do contexto de uso (NEUHAUS et al., 2013). Sobre vocabulário controlado estruturado como tesouro, Kless e Milton (2010) relacionam as seguintes características de qualidade: pureza conceitual, exaustividade conceitual, exaustividade de ponto de vista, especificidade, ausência de redundância conceitual, clareza conceitual, nível de pré-coordenação, comprimento de representação de conceito, exaustividade terminológica, ausência de redundâncias terminológicas, correção sintática, correção estrutural, completeza estrutural, pureza estrutural, navegabilidade, comprimento de trajeto até conceitos, profundidade de navegação, completeza de documentação, qualidade de documentação, complexidade e consistência. Por fim, Pinto (2008) constatou a importância dos seguintes fatores na qualidade de vocabulário controlado estruturado como tesouro: arcabouço conceitual, desempenho (*performance*), formato e sistema de auxílio.

Sobre vocabulário controlado estruturado como ontologia, em Stvilia (2007), são relacionadas as seguintes características de qualidade: acurácia/validade, coesão, complexidade, consistência semântica, consistência estrutural, atualidade, redundância, naturalidade, precisão/completude, verificabilidade, volatilidade e autoridade. Apesar de destacarem que a variedade de possíveis usos de ontologias faz com que não exista uma lista única de características, Neuhaus et al. (2013) relacionam as seguintes características: inteligibilidade, fidelidade, *craftmanship*, *fitness* e implantabilidade. A essas características, associam perguntas que questionam entendimento correto da ontologia por seres humanos, acurácia da representação do domínio, boa construção da ontologia, consistência da construção em relação a desenho (*design*), atendimento da representação do domínio a requisitos de uso e atendimento da ontologia implantada a requisitos do sistema de informação do qual ela é parte. Por fim, o quadro 10 contém termos que designam características de qualidade de vocabulários controlados e referências para as fontes onde foram identificados. Com o objetivo de minimizar mal-entendidos, nesse quadro, os termos que designam as características de qualidade foram mantidos no idioma adotado nas fontes referenciadas. As características de qualidade em Duque-Ramos et al. (2013) são descritas em Breis, Duque-Ramos e Martinez (2016). O quadro 11 contém traduções de inglês para português dos termos que designam as características. Essas traduções seguem Arouck (2011) ou fontes sobre qualidade.

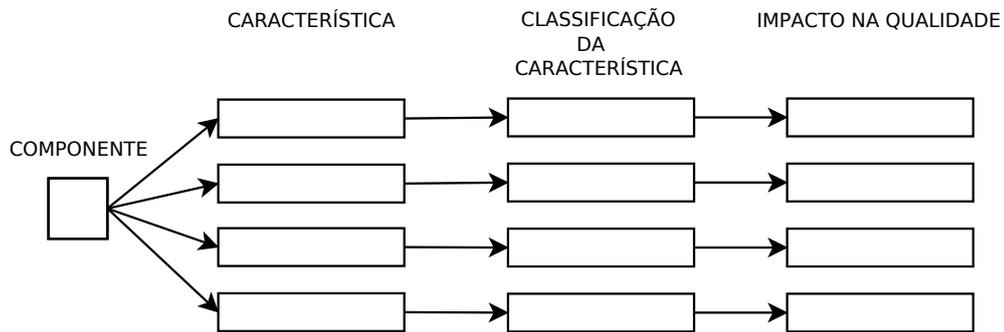
Assim como existem diversas características de qualidade de vocabulário controlado, também existem diversas propostas de medidas para caracterização da qualidade de vocabulário controlado. Em [Martínez et al. \(2011\)](#), são descritas medidas de qualidade de vocabulário controlado estruturado como tesouro. Essas medidas foram definidas levando-se em consideração normas sobre sistemas de organização do conhecimento e de teoria da classificação da informação. As medidas propostas são: percentual de termos preferidos que designam mais de um conceito, percentual de termos preferidos que carecem de relações hierárquicas, percentual de termos preferidos com dois ou mais termos genéricos, percentual de termos preferidos com um só termo específico. Sobre medidas de qualidade de vocabulário controlado estruturado como ontologia, além de diversas propostas, existem diversas ferramentas para medição da qualidade de ontologias ([LANTOW, 2016](#)).

8.6 Modelo de qualidade

Modelo de qualidade é um modelo desenvolvido com o objetivo de descrever, avaliar e/ou prever qualidade. Esses modelos podem ser desenvolvidos segundo diversas abordagens. Existem modelos de qualidade compostos por descrições de características de qualidade e por relações entre essas características. Existem também modelos de qualidade estatísticos ([WAGNER, 2013](#)). Em algumas fontes de informação, o termo “arcabouço de qualidade” (*quality framework*) é usado para designar conceito similar ao que é designado por “modelo de qualidade”. Modelos de qualidade podem ser usados em diversos contextos, por exemplo, em processo de especificação de requisitos de qualidade, em processo de implementação e em processo de avaliação da qualidade de entidades variadas. Por exemplo, produtos, processos ou recursos. Existem modelos de qualidade definidos para diversos domínios, como os modelos de qualidade de software. [Manoj \(2014\)](#) e [Al-Badareen et al. \(2011\)](#) contrastam alguns desses modelos. Finalmente, em [Manoj \(2014\)](#) é encontrada relação de atributos de qualidade que foram propostos em diversos modelos de qualidade de software.

Em [Dromey \(1996\)](#), são sugeridas as seguintes recomendações sobre processo de desenvolvimento de modelo de qualidade: modelo de qualidade deve identificar características internas de produto que tenham impacto sobre características externas de qualidade e estabelecer dependências entre essas características; inicialmente, se deve identificar características tangíveis (mensuráveis ou acessíveis), depois características em nível mais alto e menos tangíveis; regras governam o uso de componentes em um produto (regras de forma e de composição). A violação dessas regras afeta a qualidade do produto. Uso de componente incorreto, implementação incorreta de componente ou uso impróprio de componente em relação a outros componentes, violam as regras. A figura 30 apresenta o exposto.

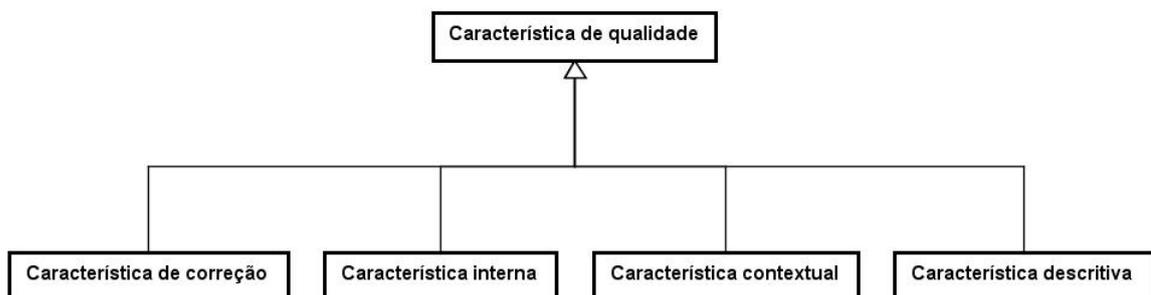
Figura 30 – Mapeamento entre componente e impacto



Fonte: Adaptado de [Dromey \(1996\)](#)

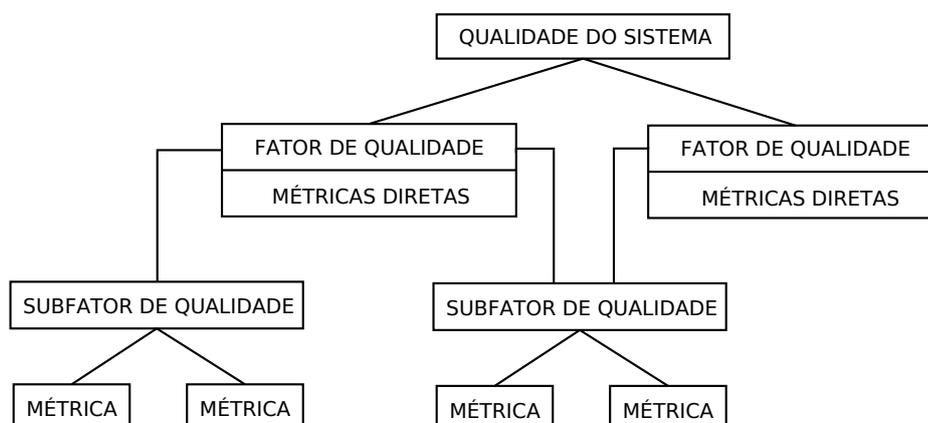
Para organizar as características de qualidade, em [Dromey \(1996\)](#), é sugerido classificá-las em estrutura hierárquica. Essa estrutura é ilustrada na figura 31. Uma estrutura hierárquica é adotada em vários modelos de qualidade. Por exemplo, em [IEEE \(2009\)](#), são sugeridos modelo e método com os seguintes objetivos: estabelecimento de requisitos de qualidade; identificação, implementação, análise e validação de medidas de qualidade. Esse modelo engloba os seguintes elementos: fator de qualidade, subfator de qualidade e medida (métrica). A figura 32 ilustra relações entre esses elementos. Por fim, o método descrito em [IEEE \(2009\)](#), é composto pelas seguintes atividades: definir requisitos de qualidade; identificar, implementar e analisar resultados; e validar medidas de qualidade.

Figura 31 – Hierarquia de características de qualidade



Fonte: Adaptado de [Dromey \(1996\)](#)

Figura 32 – Modelo de medidas (métricas) de qualidade



Fonte: Adaptado de [IEEE \(2009\)](#)

8.6.1 Modelo de qualidade da informação

[Eppler e Wittig \(2000\)](#) relacionam os seguintes propósitos de arcabouço de qualidade da informação (*information quality framework*): prover conjunto sistemático e conciso de critérios de acordo com os quais informação possa ser avaliada; prover esquema para análise e solução de problemas de qualidade da informação; prover embasamento para medição de qualidade da informação e para gerenciamento proativo; prover comunidade de pesquisa com mapa conceitual que possa ser usado para estruturar variedade de abordagens, teorias e fenômenos relacionados à qualidade da informação. Sugerem também questões para avaliação desses arcabouços de qualidade da informação. A partir dessas questões, é possível identificar os seguintes elementos como desejáveis a esses arcabouços: definições e explicações claras das características de qualidade; se características de qualidade forem agrupadas em dimensões, definições e explicações sobre essas dimensões; critérios de qualidade mutuamente exclusivos e coletivamente exaustivos; divisão do arcabouço em dimensões sistemáticas, mutuamente exclusivas e coletivamente exaustivas; concisão; exemplos que expliquem critérios de qualidade; ferramentas (questionários, guias de implementação etc.); descrição de contexto de aplicação e limites do arcabouço; e posicionamento em termos da literatura. A partir de análise de diversos arcabouços de qualidade da informação (resultados dessa análise são descritos em [Eppler e Wittig \(2000\)](#)), em [Eppler \(2001\)](#) é proposto um arcabouço de qualidade da informação composto por visões, fases e princípios. As visões são as seguintes: informação relevante (*relevant information*), informação sólida (*solid information*), processo otimizado e infraestrutura confiável. As fases adotam sequência cronológica segundo ponto de vista do usuário da informação, e são as seguintes: identificação, avaliação, alocação e aplicação. Os princípios

orientam a implementação do arcabouço de qualidade.

8.6.2 Modelo de qualidade de modelo conceitual

Segundo [Moody \(2005\)](#), existem diversas propostas de modelos de qualidade para modelos conceituais e esses modelos frequentemente apresentam deficiências, por exemplo, as seguintes: falta de testes empíricos e de adoção prática, diferentes níveis de generalidade, falta de acordo sobre conceitos e termos, falta de consistência com campos relacionados e com padrões, falta de definição sobre como medir critérios de qualidade, falta de procedimentos para avaliação da qualidade, falta de recomendações sobre como melhorar qualidade, foco em modelos estáticos, foco na qualidade de produto e não na qualidade de processo, e falta de conhecimento sobre a prática. [Lindland, Sindre e Sølvsberg \(1994\)](#) destacam que diversos modelos de qualidade de modelos conceituais são listas de propriedades sem estrutura definida para avaliação.

Segundo [Piattini et al. \(2005\)](#), em termos históricos, inicialmente foram propostas listas de propriedades não estruturadas e imprecisas; em seguida, surgiram propostas mais sistemáticas, porém sem avaliação quantitativa; finalmente, surgiram medidas propostas para características de qualidade de modelos conceituais. [Lindland, Sindre e Sølvsberg \(1994\)](#) relacionam as seguintes deficiências típicas de modelos de qualidade para modelos conceituais: definições vagas, complicadas ou em falta; lista não estruturada de propriedades; sobreposição de propriedades; propriedades que pressupõem a existência de desenho (*design*) ou de implementação; objetivos irrealistas ou impossíveis de serem alcançados. Esses autores sugerem características desejáveis a modelos de qualidade propostos para modelos conceituais: separação de objetivos de qualidade dos meios para alcançá-los; definição de objetivos realistas; propriedades diretamente relacionadas à construção da especificação, sem depender de entregas posteriores do projeto.

Sobre processo de desenvolvimento de modelo de qualidade de modelo conceitual, em [Moody \(2005\)](#) são apresentadas as seguintes sugestões: organizar características de qualidade em hierarquias e medidas, identificar característica de qualidade por palavra de fácil entendimento, definir característica de qualidade por sentença concisa, e definir medidas e procedimentos para avaliação da qualidade. [Moody \(2005\)](#) também destaca a importância de consistência dos modelos de qualidade com a norma ISO 9000, pois considera que um modelo conceitual é um tipo de produto. Sugere que modelos de qualidade de modelos conceituais sejam consistentes com a norma ISO/IEC 9126, pois considera que são frequentemente usados no desenvolvimento de sistemas de informação. Quanto a alternativas para o processo de desenvolvimento de modelo de qualidade de modelo conceitual, são relacionadas as seguintes em [Moody \(2005\)](#): embasar modelo de qualidade em teoria, embasar modelo de qualidade em experiência, embasar modelo de qualidade em consenso, desenvolver modelo de qualidade por observação, sintetizar propostas existentes,

desenvolver modelo de qualidade a partir de norma sobre qualidade, desenvolver o modelo de qualidade por meio da metodologia *Goal Question Metric* (GQM), e desenvolver o modelo de qualidade por meio da abordagem descrita em [Dromey \(1995\)](#).

8.6.3 Modelo de qualidade de vocabulário controlado

Segundo [Stvilia \(2007\)](#), sendo uma ontologia uma especificação de uma conceituação de um domínio, um modelo de avaliação de qualidade de ontologias deve focar a avaliação da qualidade de mapeamentos de conceitos em domínios, em classes, relações e vocabulários nas ontologias, e possibilitar a avaliação da qualidade da representação dos conceitos no domínio. Em [Vrandecic \(2010\)](#), é descrito arcabouço de avaliação de qualidade de ontologias no qual são definidos critérios e aspectos para avaliação de ontologias. Os nomes dos critérios definidos são: acurácia, adaptabilidade, clareza, completude, eficiência computacional, concisão, consistência e adequação organizacional (*organizational fitness*). Nesse arcabouço, são abordados vocabulário, sintaxe, estrutura, semântica, representação e contexto. [Duque-Ramos et al. \(2011\)](#) e [Duque-Ramos et al. \(2013\)](#) contêm informação sobre *OquaRE*, um arcabouço que engloba modelo de qualidade de ontologia desenvolvido a partir de padrões de qualidade de software e de medidas de qualidade. Em *OquaRE*, ontologia é avaliada como produto de processo de desenvolvimento e as definições das características de qualidade de ontologias são embasadas na norma ISO/IEC 25000:2005, a qual define padrão de qualidade conhecido como *SQuaRE*. O desenvolvimento de *OquaRE* englobou adaptação de padrão de qualidade, padronização de processo de avaliação e definição do modo como as medidas são associadas a características de qualidade. O modelo de qualidade é hierárquico. No nível mais alto da hierarquia, são encontradas as seguintes características de qualidade: estrutural, adequação funcional, confiabilidade, eficiência de desempenho, operabilidade, manutenibilidade, compatibilidade, transferibilidade e qualidade em uso. A essas características de qualidade, são associadas subcaracterísticas de qualidade. Finalmente, são associadas medidas de qualidade às subcaracterísticas propostas.

Quadro 10 – Características de qualidade de vocabulários controlados

CARACTERÍSTICAS DE QUALIDADE	FONTES DE INFORMAÇÃO	
Accuracy	Burton-Jones et al. (2005)	Stvilia (2007)
	Raad e Cruz (2015)	Lantow (2016)
	Hloman e Stacey (2014)	Elkin, Brown e Chute (2001)
Adaptability	Duque-Ramos et al. (2013)	Lantow e Sandkuhl (2015)
	Raad e Cruz (2015)	
Adequacy	Duque-Ramos et al. (2013)	
Analysability	Duque-Ramos et al. (2013)	
Appropriateness recognisability	Duque-Ramos et al. (2013)	
Authority	Burton-Jones et al. (2005)	Stvilia (2007)
Availability	Duque-Ramos et al. (2013)	
Changeability	Duque-Ramos et al. (2013)	
Clarity	Burton-Jones et al. (2005)	Gruber (1995)
	Yu, Thom e Tam (2007)	Uschold e Gruninger (1996)
	Lantow e Sandkuhl (2015)	Raad e Cruz (2015)
	Hloman e Stacey (2014)	
Compatibility	Duque-Ramos et al. (2013)	
Completeness	Gómez-Pérez (1996)	Yu, Thom e Tam (2007)
	Gómez-Pérez (2001)	Raad e Cruz (2015)
	Tibaldo et al. (2015)	Lacasta et al. (2016)
	Kless e Milton (2010)	
Complexity	Zhang, Li e Tan (2010)	Damasevicius (2009)
	Kang et al. (2004)	Stvilia (2007)
	Kless e Milton (2010)	Lacasta et al. (2016)
Comprehensiveness	Elkin, Brown e Chute (2001)	Burton-Jones et al. (2005)
Conciseness	Gómez-Pérez (1996)	Yu, Thom e Tam (2007)
	Gómez-Pérez (2001)	Raad e Cruz (2015)
	Lantow (2016)	Tibaldo et al. (2015)
	Hloman e Stacey (2014)	
Consistency	Burton-Jones et al. (2005)	Gómez-Pérez (1996)
	Yu, Thom e Tam (2007)	Gómez-Pérez (2001)
	Duque-Ramos et al. (2013)	Raad e Cruz (2015)
	Hloman e Stacey (2014)	Kless e Milton (2010)
Correctness	Yu, Thom e Tam (2007)	Kless e Milton (2010)
Currency	Stvilia (2007)	
Ease of use	Duque-Ramos et al. (2013)	
Extensibility	Uschold e Gruninger (1996)	Fox, Barbuceanu e Gruninger (1996)
Generality	Fox, Barbuceanu e Gruninger (1996)	
Helpfulness	Duque-Ramos et al. (2013)	
Interoperability	Duque-Ramos et al. (2013)	
Interpretability	Burton-Jones et al. (2005)	
Learnability	Duque-Ramos et al. (2013)	
Maintainability	Duque-Ramos et al. (2013)	
Modularity	Duque-Ramos et al. (2013)	Oh e Yeom (2012)
Operability	Duque-Ramos et al. (2013)	
Performance efficiency	Duque-Ramos et al. (2013)	
Portability	Duque-Ramos et al. (2013)	
Precision	Duque-Ramos et al. (2013)	Stvilia (2007)
Quality in use	Duque-Ramos et al. (2013)	
Recoverability	Duque-Ramos et al. (2013)	
Relevance	Burton-Jones et al. (2005)	
Reliability	Duque-Ramos et al. (2013)	
Replaceability	Duque-Ramos et al. (2013)	
Reusability	Duque-Ramos et al. (2013)	Lantow e Sandkuhl (2015)
Scalability	Fox, Barbuceanu e Gruninger (1996)	
Testability	Duque-Ramos et al. (2013)	
Understandability	Lantow e Sandkuhl (2015)	Lantow (2016)
	Coronado et al. (2009)	
Usability	Elkin, Brown e Chute (2001)	Pizzoleto e Oliveira (2016)
	Coronado et al. (2009)	
Volatility	Stvilia (2007)	

Fonte: Elaborado pelo autor

Quadro 11 – Traduções de termos

TERMOS EM INGLÊS	TERMOS EM PORTUGUÊS	TERMOS EM INGLÊS	TERMOS EM PORTUGUÊS
Accuracy	Acurácia	Interoperability	Interoperabilidade
Adaptability	Adaptabilidade	Interpretability	Interpretabilidade
Adequacy	Suficiência	Learnability	Aprendibilidade
Analysability	Analísabilidade	Maintainability	Manutenibilidade
Appropriateness recognisability	Reconhecibilidade de adequação	Modularity	Modularidade
Authority	Autoridade	Operability	Operabilidade
Availability	Disponibilidade	Performance efficiency	Eficiência de desempenho
Changeability	Modificabilidade	Portability	Portabilidade
Clarity	Clareza	Precision	Precisão
Compatibility	Compatibilidade	Quality in use	Qualidade em uso
Completeness	Compleitude	Recoverability	Recuperabilidade
Complexity	Complexidade	Relevance	Pertinência
Comprehensiveness	Abrangência	Reliability	Confiabilidade
Conciseness	Concisão	Replaceability	Substituibilidade
Consistency	Coerência	Reusability	Reusabilidade
Correctness	Correção	Scalability	Escalabilidade
Currency	Atualidade	Testability	Testabilidade
Ease of use	Facilidade de uso	Understandability	Compreensibilidade
Extensibility	Extensibilidade	Usability	Usabilidade
Generality	Generalidade	Volatility	Volatilidade
Helpfulness	Utilidade		

Fonte: Elaborado pelo autor

9 Elementos de Engenharia de Software

Este capítulo contém definição de “engenharia de software” e informação sobre: necessidades de informação de engenheiros de software, ciclo de vida de software, recursos de informação em ciclo de vida de software, processos em ciclo de vida de software, desenvolvimento de processos em ciclo de vida de software, melhoria de processos em ciclo de vida de software, modelo de maturidade, processo de desenvolvimento de software, repositório de software, recuperação da informação em repositório de software, vocabulários controlados e metadados em Engenharia de Software. Os assuntos abordados neste capítulo são relevantes pois o domínio da disciplina Engenharia de Software é aquele para o qual será desenvolvido exemplo de uso de elementos do arcabouço para ciclo de vida de projeto de vocabulário controlado proposto no capítulo 10. Além disso, contribuições do domínio da Engenharia de Software também são relevantes na proposta dos elementos desse arcabouço.

9.1 Definições de engenharia de software

O termo “engenharia de software” pode ser usado para designar disciplina, produto de disciplina ou objeto de estudo de disciplina. O termo “engenharia de software” geralmente designa estudo e aplicação de abordagem sistemática, disciplinada e quantificável ao desenvolvimento, à operação e à manutenção de software (ISO, 2010). Por sua vez, o termo “Engenharia de Software” designa disciplina de engenharia relacionada a todos os aspectos de produção de software (SOMMERVILLE, 2010). Elementos do corpo de conhecimento da disciplina Engenharia de Software são descritos em IEEE (2014c), guia desenvolvido para promover visão consistente da disciplina; caracterizar conteúdo da disciplina; especificar escopo da disciplina e posicioná-la em relação a outras disciplinas; possibilitar acesso ao corpo de conhecimento da disciplina por meio de tópicos; e prover fundamentos para desenvolvimento de currículos, materiais de licenciamento e certificação (IEEE, 2014c).

Na fonte IEEE (2014c), o corpo de conhecimento da disciplina Engenharia de Software é organizado nas seguintes áreas de conhecimento: Requisitos de Software; Desenho (*design*) de Software, Construção de Software, Teste de Software, Manutenção de Software, Gerência de Configuração de Software, Gerência de Engenharia de Software, Processo de Engenharia de Software, Métodos e Modelos de Engenharia de Software, Qualidade de Software, Prática Profissional de Engenharia de Software, Economia de Engenharia de Software, Fundamentos de Computação, Fundamentos de Matemática e Fundamentos de Engenharia. Nesse contexto, o termo desenho (*design*) designa processo de definição de arquitetura de software, componentes, módulos, interfaces e dados para que um sistema de software satisfaça requisitos especificados, ou resultado desse processo

(ISO, 2010).

O desenvolvimento de software geralmente ocorre em projetos, requer esforço coletivo e execução de atividades integrantes de diversos processos. Nesse contexto, o termo “software” designa “programas de computador, procedimentos, possivelmente documentação e dados associados, pertinentes à operação de um sistema de computador” (IEEE, 2012a). Um programa de computador é “combinação de instruções de computador e definições de dados que capacitam o hardware de um computador a realizar funções computacionais ou funções de controle”. O hardware de computador é o “equipamento físico usado para processar, armazenar ou transmitir programas de computador ou dados” (ISO, 2010). Sistema de computador é sistema com um ou mais computadores e software associado. Computador é “unidade funcional que pode realizar computações substanciais, inclusive numerosas operações aritméticas e lógicas sem a intervenção humana” (ISO, 2010). Nesse contexto, projeto é “empreendimento com início e fim definidos, realizado para criar produto ou serviço de acordo com recursos e requisitos especificados” é um “empreendimento temporário realizado para criar produto, serviço ou resultado único” (ISO, 2008a; PMI, 2013). Finalmente, processo é “série de atividades direcionada a causar um resultado final de modo que uma ou mais entradas sejam atuadas com o objetivo de criar uma ou mais saídas” (IEEE, 1990; PMI, 2013).

9.2 Necessidades de informação de engenheiros de software

Existem várias pesquisas sobre as necessidades de informação de engenheiros, particularmente sobre as necessidades de informação de engenheiros de software (ALLARD; LEVINE; TENOPIR, 2009). Em relação a engenheiros de software responsáveis por realizar manutenção de software, Seaman (2002) destaca o uso de código fonte (*source code*) e pessoas com experiência no código fonte em processo de manutenção de software. Também destaca a importância de capturar e disponibilizar informação sobre princípios de desenho (*design*) e as intenções do sistema de modo a tornar os responsáveis pela manutenção menos dependentes dos responsáveis pelo desenvolvimento. Ainda sobre necessidades de informação de engenheiros de software atuando em manutenção de software, após coletarem informação em variados estudos, Koskinen, Salminen e Paakki (2004) relacionam necessidades de informação mais frequentes, dentre elas, descrições de conceitos no domínio. Sobre as fontes de informação, relacionam código, execução de código, comentários em código e documentação de software. Em Grzywaczewski e Iqbal (2012), são reportadas necessidades de informação de desenvolvedores de software, por exemplo: documentação de código, exemplo de código, solução para problema relacionado ao código, ideia genérica sobre a abordagem a adotar, notícia sobre tecnologia e material técnico. Em Freund, Toms e Waterhouse (2005), é reportada pesquisa sobre comportamento informacional de engenheiros de software que atuam como consultores. Segundo essa pesquisa, as fontes

de informação usadas encontravam-se armazenadas em diversos sítios e variavam desde mensagens de correio eletrônico até manuais. Os recursos digitais acessados estavam em variados formatos. Quanto à importância da documentação de software na realização de tarefas por engenheiros de software, pesquisa relatada em [Lethbridge, Singer e Forward \(2003\)](#) revela que a documentação de software é particularmente relevante nos seguintes contextos: aprendizado sobre sistema de software, teste de software, trabalho com sistema de software, solução de problemas quando outros desenvolvedores não estão disponíveis para responder perguntas, obtenção de panorama geral sobre sistema de software, manutenção de sistema de software, resposta a questões da gerência ou de clientes sobre sistema de software, e pesquisa por informação aprofundada sobre sistema de software. A partir do anteriormente exposto, constata-se a variedade de necessidades de informação de engenheiros de software, assim como a diversidade de fontes de informação por eles acessadas.

9.3 Ciclo de vida de software

Ciclo de vida pode ser definido como evolução, da concepção à retirada, de sistema, produto, serviço, projeto ou entidade feita pelo ser humano. Por retirada, entende-se remoção de suporte ativo por parte da organização responsável pela manutenção e operação, substituição parcial ou total por novo sistema ou instalação de sistema atualizado ([ISO, 2008a](#)). Um ciclo de vida de software define fases na vida do produto de software ([FUGGETTA, 2000](#)). Essas fases são determinadas por necessidades dos responsáveis pelo produto. Um ciclo de vida de software geralmente inicia na percepção de necessidades de usuários (pessoas que operam o produto ou que interagem com o produto) ou clientes (pessoas que pagam pelo produto e usualmente decidem sobre requisitos) e termina quando o uso do produto é descontinuado ([IEEE, 1993](#); [ISO, 2010](#)). O ciclo de vida de software inclui processos, tais como, desenvolvimento, implantação, manutenção, suporte, evolução e descarte de produto ([IEEE, 2014c](#)).

Um “modelo de ciclo de vida” é um arcabouço de processos e atividades relacionadas a ciclo de vida, pode ser organizado em fases e servir de referência para comunicação e entendimento ([ISO, 2008a](#)). Os modelos de ciclo de vida de software geralmente englobam processos, atividades e tarefas pertinentes ao desenvolvimento, operação e manutenção de software ([IEEE, 2016](#)). Existem também modelos de ciclo de vida de desenvolvimento de software, por exemplo, linear, iterativo e ágil. No modelo linear, as fases do desenvolvimento ocorrem sequencialmente e, no modelo iterativo, o desenvolvimento ocorre em incrementos em ciclos iterativos. Finalmente, no modelo ágil, o desenvolvimento ocorre em ciclos iterativos curtos que produzem pequenos incrementos ao software ([IEEE, 2014c](#)). Se for adaptativo, passa-se por várias iterações ao longo do desenvolvimento. Nos processos que seguem o modelo iterativo e incremental, o desenvolvimento é organizado em iterações e o

software é desenvolvido de modo incremental em sucessivas iterações. Cada iteração contém fluxos de atividades é planejada e tem critérios definidos para sua avaliação. A diferença entre resultados de duas iterações subsequentes é um incremento (JACOBSON; BOOCH; RUMBAUGH, 1999). Um processo de desenvolvimento de software pode ser preditivo ou adaptativo. Se for preditivo, tenta planejar o desenvolvimento por períodos relativamente longos. Por sua vez, em processos adaptativos, geralmente não há um planejamento detalhado para todo desenvolvimento. Há planejamento de alto nível onde se estima data final do desenvolvimento e marcos relevantes e o detalhamento do planejamento ocorre a cada iteração.

9.4 Processo de software

Processo de software pode ser definido como “conjunto coerente de políticas, estruturas organizacionais, tecnologias, procedimentos e artefatos necessários à concepção, desenvolvimento, implantação e manutenção de produto de software” (FUGGETTA, 2000). Ao longo do ciclo de vida de software, ocorre desenvolvimento, operação e manutenção por meio de processos intensivos na aquisição, produção e uso de informação. As definições desses processos têm os seguintes objetivos: facilitar entendimento, comunicação e coordenação entre pessoas; auxiliar gerenciamento de projetos; medir e melhorar qualidade de software; prover suporte a melhorias; prover base para automação do suporte à execução de processos (FUGGETTA, 2000; IEEE, 2014c). Os processos em ciclo de vida de software podem ser organizados de diversos modos. A seguir, são listados nomes de classes de processos segundo a norma ISO (2008a): Processos de Acordo, Processos Organizacionais Capacitadores de Projeto, Processos de Projeto, Processos Técnicos, Processos de Implementação de Software, Processos de Suporte de Software, e Processos de Reuso de Software.

Na fonte ISO (2008a), a descrição de cada processo contém: título, propósito, resultados (*outcome*), lista de atividades e tarefas (*task*) para alcançar os resultados. No contexto desta tese, são particularmente relevantes o Processo de Gestão de Informação (*Information Management Process*) e o Processo de Gestão de Documentação (*Software Documentation Management Process*). O Processo de Gestão de Informação integra o grupo Processos de Projeto. Esse processo é responsável por gerar, coletar, transformar, reter, recuperar, disseminar e descartar informação. Visa prover informação relevante, em tempo, completa, válida e se requerido confidencial, durante e quando apropriado, após o ciclo de vida do sistema (ISO, 2008a). A seguir, são listados possíveis resultados desse processo: identificação da informação a ser gerenciada; definição de formas de representação da informação; transformação e disponibilização da informação de acordo com o requerido; registro do estado da informação; informação corrente, completa e válida; informação disponível a partes designadas. O Processo de Gestão de Informação

engloba planejamento e execução de gestão da informação. O planejamento engloba definir itens de informação que devem ser gerenciados em ciclo de vida do sistema de software e posteriormente mantidos; designar autoridades e responsabilidades relativas à origem, geração, captura, arquivamento e descarte de itens de informação; definição de direitos, obrigações e compromissos relacionados à retenção, transmissão e acesso a itens de informação; definição de conteúdo, semântica, formatos e meios para representar, reter, transmitir e recuperar informação; e definição de ações para manutenção da informação. A gestão da informação engloba obter itens identificados de informação; manter itens de informação e registros de armazenamento segundo critérios de integridade, segurança e privacidade; recuperar e distribuir informação de acordo com cronogramas ou circunstâncias definidas; prover documentação oficial de acordo com o requerido; arquivar informação de acordo com propósitos definidos; e descartar informação de acordo com políticas e requisitos definidos. O Processo de Gestão de Documentação integra o grupo Processos de Suporte de Software (*Software Support Processes*). Esse processo é especialização do Processo de Gestão de Informação e é responsável por desenvolver e prover manutenção à informação registrada sobre software. São possíveis resultados desse processo: identificação da documentação a produzir; padrões a aplicar na documentação; especificação, revisão e aprovação de conteúdo e propósito da documentação; desenvolvimento da documentação; manutenção da documentação. No contexto desta tese, são também relevantes processos no grupo Processos de Implementação de Software. A seguir, são listados nomes desses processos: Processo Análise de Requisitos de Software, Processo Desenho (*design*) Arquitetural de Software, Processo Desenho (*design*) Detalhado de Software, Processo Construção de Software, Processo Integração de Software, e Processo Teste de Qualificação de Software.

Os processos em ciclo de vida de software podem ser desenvolvidos por meio de metodologias. A norma *IEEE 1074:2006 IEEE Standard for Developing a Software Project Lifecycle Process* descreve metodologia para arquiteto de processo (pessoa ou entidade organizacional) desenvolver processo em ciclo de vida de projeto de software (*software project life cycle process*). Nesse contexto, processo em ciclo de vida de projeto de software é descrição de processo desenvolvida acrescentando-se ativos de processos organizacionais (*organizational process asset*) a um ciclo de vida de projeto de software. Por sua vez, o termo “ciclo de vida de projeto de software” (*software project life cycle*) designa porção de ciclo de vida de software aplicável a projeto específico de software (ISO, 2010). Os ativos de processos organizacionais definem ambientes organizacionais para projetos de software e esses ativos geralmente podem ser adaptados às particularidades de projetos (IEEE, 2006).

9.4.1 Processos de desenvolvimento de software

Na norma IEEE (2006), são descritas atividades de processos em ciclo de vida de projeto de software, organizadas em seções e em grupos de atividades relacionadas. Em

IEEE (2006), o termo atividade designa corpo de trabalho a ser realizado, é descrição de transformação de informação de entrada em informação de saída. Uma atividade é concluída quando toda informação de entrada é processada e toda informação de saída é gerada. A descrição de uma atividade discute ações a serem realizadas para que a transformação seja realizada. Nesse contexto, atividades não são processos, mas podem formar processos, por exemplo, ao serem mapeadas para um modelo de ciclo de vida de projeto de software. Nesse mapeamento, as atividades podem ser sequenciadas com o objetivo de formar processo. O termo “processo de desenvolvimento de software” frequentemente designa a tradução de necessidades de usuários ou clientes em produtos de software (ISO, 2010). Esses processos geralmente englobam atividades responsáveis por traduzir necessidades de usuários ou clientes em requisitos, transformar requisitos em desenho (*design*), implementar desenho (*design*) e código, testar código, instalar e verificar uso do software (ISO, 2010). A definição de um processo de desenvolvimento de software geralmente descreve e ordena atividades, descreve artefatos e atribui responsabilidades a envolvidos. Processos de desenvolvimento de software incluem atividades que resultam na criação ou na atualização de diversos artefatos. Nesse contexto, o termo artefato designa produto tangível do trabalho em desenvolvimento de software. A seguir, são descritos relevantes processos de desenvolvimento de software.

9.4.1.1 Processo Unificado de Desenvolvimento de Software

O Processo Unificado de Desenvolvimento de Software (*Unified Software Development Process*) é um processo de desenvolvimento de software iterativo e incremental que pode ser adotado em projetos de diferentes dimensões. Nesse processo, o desenvolvimento de software passa por ciclos divididos em fases e termina com a liberação de software para produção. As fases são divididas em iterações que são concluídas quando alcançados marcos definidos. Em cada iteração, vários fluxos de trabalho podem ocorrer. A diferença entre resultados de iterações subsequentes é um incremento. Cada ciclo de desenvolvimento é dividido nas seguintes fases: concepção, elaboração, construção e transição. A figura 33 apresenta relação entre ciclo, fase e iteração. As responsabilidades da fase Concepção são: avaliar viabilidade do desenvolvimento do software; identificar escopo, entidades que interagirão com o software e casos de uso; descrever casos de uso significativos; estabelecer critérios de sucesso; definir recursos necessários e realizar planejamento. Na fase Elaboração, o domínio do problema é analisado, a arquitetura do software é definida e procura-se garantir que arquitetura, requisitos e planos estejam estáveis, que os riscos sejam conhecidos, que seja possível determinar custos e cronograma. Nessa fase é construído protótipo executável que realiza casos de uso críticos e expõe riscos técnicos. Também podem ser construídos protótipos descartáveis para avaliação de aspectos específicos. Ao final dessa fase, deve ser possível avaliar se o desenvolvimento será um sucesso, caso sejam adotados a arquitetura e o planejamento propostos. Na fase Construção, os componentes do software são construídos e integrados. Avalia-se também se os requisitos foram satisfeitos.

O resultado dessa fase é o software a ser entregue aos usuários. Por fim, na fase Transição, o software migra para os usuários. Nessa fase, os usuários são treinados e lhes é provido suporte. Sugestões dos usuários podem ainda resultar em modificações. Ao final dessa fase, é avaliado se o software atingiu os seus objetivos ou se ainda é necessário novo ciclo de desenvolvimento (JACOBSON; BOOCH; RUMBAUGH, 1999; KRUCHTEN, 2003).

Figura 33 – Modelo de ciclo de vida no Processo Unificado



Fonte: Elaborado pelo autor

Existem variantes do Processo Unificado de Desenvolvimento de Software, por exemplo, o *Enterprise Unified Process* (EUP). Nesse processo, cada ciclo de desenvolvimento é dividido em Concepção, Elaboração, Construção, Transição e Descarte (*retirement*). As disciplinas são classificadas da seguinte forma: Disciplinas de Desenvolvimento, Disciplinas de Suporte e Disciplinas Empresariais (*Enterprise Disciplines*). Nas classes Disciplinas de Desenvolvimento e Disciplinas de Suporte, são encontradas as seguintes disciplinas: Modelagem de Negócios, Requisitos, Análise e Desenho (*Design*), Implementação, Teste e Implantação, Configuração e Gestão de Mudanças, Gestão de Projetos, Ambiente, Operação e Suporte. Por sua vez, na classe Disciplinas Empresariais, encontram-se: Modelagem de Negócios Empresarial (*Enterprise Business Modeling*), Gestão de Portfólio, Arquitetura Empresarial (*Enterprise Architecture*), Reuso Estratégico, Gestão de Pessoal, Administração Empresarial (*Enterprise Administration*), Melhoria de Processo de Software (AMBLER, 2013).

9.4.1.2 Processo ágil de desenvolvimento de software

O termo “processo ágil de software” se originou no Japão e identifica processos de software com as seguintes características: as atividades não são prescritas visando-se

criar um ambiente de linha de produção para manufatura de software; ciclo de desenvolvimento decomposto em iterações com curta duração; iterações usadas no planejamento do desenvolvimento do software; em cada iteração, encontram-se as atividades para se alcançar objetivos da iteração; minimização do número de atividades visando a menor quantidade de atividades para se reduzir riscos e alcançar objetivos; atividades podem ser acrescentadas ou descartadas, na medida em que riscos surgem ou são descartados; software dividido em incrementos posteriormente integrados; processo orientado às pessoas, nesse processo é atribuído poder aos desenvolvedores; enfatiza a comunicação entre os envolvidos no desenvolvimento (MILLER, 2001).

Um exemplo de processo ágil de desenvolvimento de software é denominado *Extreme Programming* (XP). As práticas nesse processo são: cliente decide prazos e escopos das entregas e participa do desenvolvimento; software colocado em produção em curto espaço de tempo; novas entregas realizadas frequentemente; simplificação do desenho (*design*); testes de unidade realizados frequentemente; desenho (*design*) evolui por meio de transformações denominadas refatorações (*refactoring*); códigos escritos por duplas de desenvolvedores; novos códigos integrados a códigos existentes em curtos intervalos de tempo; desenvolvedores trabalham em ambientes abertos; qualquer desenvolvedor pode melhorar qualquer código; e minimização de trabalho em horas extras (BECK, 1999; BECK; ANDRES, 2004).

9.4.1.3 Processo distribuído de desenvolvimento de software

Em processo de desenvolvimento de software, os participantes podem estar próximos ou distantes. No desenvolvimento distribuído de software pode ocorrer distância geográfica, temporal, cultural e/ou linguística. Devido a essas distâncias, coordenação, visibilidade, comunicação e cooperação efetivas são particularmente importantes. O processo de desenvolvimento de software pode ser globalmente distribuído. Nesse caso, ocorre por meio de fluxos de trabalho geograficamente distribuídos (LAPLANTE, 2007). Nesse processo, geralmente tecnologias de comunicação são usadas em tarefas interdependentes; os envolvidos no desenvolvimento trabalham em um contexto onde são cruzadas fronteiras temporais, geográficas e organizacionais; e a comunicação geralmente ocorre por meios eletrônicos assíncronos. Em desenvolvimento globalmente distribuído de software, a gestão da informação é importante ao sucesso do desenvolvimento (RICHARDSON et al., 2012). O processo distribuído de desenvolvimento de software geralmente ocorre, por exemplo, em projetos de desenvolvimento de software aberto (*open source software development*). A seguir, são listadas características típicas do processo de desenvolvimento nesses projetos: código de software livremente disponível para acesso e para modificação, artefatos resultantes do processo são armazenados em repositórios de software, desenvolvimento ocorre em comunidades em linha (*online*), participantes do projeto podem ser voluntários ou podem ser remunerados, comunidades geralmente não apresentam estruturas estritamente

hierárquicas, a influência de cada participante depende de suas contribuições, incentivo à participação, ênfase na comunicação assíncrona por meio de redes de computadores, uso de portais para acesso a fontes de informação, defeitos e requisitos registrados por meio do uso de sistema de informação (ANKOLEKAR; HERBSLEB; SYCARA, 2003; ANKOLEKAR, 2004; ERENKRANTZ; TAYLOR, 2003; GODFREY; TU, 2000; GUTWIN; PENNER; SCHNEIDER, 2004; MOCKUS; FIELDING; HERBSLEB, 2002; MOODY, 2001; NAKAKOJI et al., 2002; REIS; FORTES, 2002; SCACCHI et al., 2006; SCOTTO; SILLITTI; SUCCI, 2007).

9.5 Recursos de informação em ciclo de vida de software

Em um ciclo de vida de software, diversos recursos de informação são geralmente usados: código, glossário, modelo, plano, guia, *template*, especificação de requisitos, documento de metas do negócio, documento de visão do negócio, documento de arquitetura do negócio, regra de negócio, solicitação de *stakeholder*, *storyboard*, documento de arquitetura, mapa de navegação, sumário de avaliação de teste, *script* de teste, *log* de teste, caso de teste, resultado de teste, estratégia de teste, arquitetura de automação de teste, configuração do ambiente de teste, lista de materiais, notas de entrega (*release note*), material de treinamento, arte de produto, material de suporte ao usuário final, solicitação de modificação, avaliação de iteração, lista de riscos, ordem de trabalho, registro de revisão, história de usuário e descrição de interface de programação (ECLIPSE FOUNDATION, 2012; IEEE, 2006; JACOBSON; BOOCH; RUMBAUGH, 1999; OPENSTACK, 2015; SHUJA; KREBS, 2008; WELLS, 2013).

A norma ISO (2017) especifica propósitos e conteúdos de recursos de informação em ciclo de vida de software e sistema, assim como recursos de informação para gestão de serviços de tecnologia da informação. Essa norma pode ser aplicada em diversos contextos: suporte a processo de gestão da informação em ciclo de vida de sistema; identificação de requisitos e conteúdos de recursos de informação em ciclo de vida de software; determinação de quais recursos de informação são necessários à garantia da qualidade de projeto, sistema, produto ou serviço; desenho (*design*) e desenvolvimento de recursos de informação de serviços, sistemas ou software; melhoria de processos de serviços, sistemas ou software em organizações. Em ISO (2017), são definidas as seguintes classes de recursos de informação: descrição (*description*), plano (*plan*), política (*policy*), procedimento (*procedure*), relatório (*report*), solicitação (*request*) e especificação (*specification*). Para cada uma dessas classes, essa norma define propósito e sugere elementos que devem ser incluídos e recursos de informação. Para os recursos de informação, há definições, listas de normas relacionadas e elementos integrantes. Entre os recursos de informação, existem os que documentam software. Graaf (2011) observa a importância da documentação de software em projeto de software. Também destaca que a documentação de software pode explicitar motivos de

decisões de desenho (*design*), prover suporte à coordenação, prover suporte a processo de aprendizado e impactar a qualidade do software.

Classes de recursos de informação relacionados a processos de software podem ser identificadas por meio de análise das normas ISO (2008a) e ISO (2017). Por exemplo, o quadro 12 contém recursos de informação relacionados aos processos “Processo de Desenho Arquitetural de Software” e “Processo de Desenho Detalhado de Software” do grupo “Processos de Implementação de Software” descrito em ISO (2008a). Esses recursos de informação são descritos em ISO (2017). Outra alternativa para identificação de classes de recursos de informação relacionados a processos de software, é auditar repositórios de software.

Quadro 12 – Exemplos de classes de recursos de informação

CLASSE	RECURSO DE INFORMAÇÃO
Descrição	Conceito de operações Descrição de arquitetura de sistema Descrição de desenho (<i>design</i>) de base de dados Descrição de interface Descrição de arquitetura de software Descrição de desenho (<i>design</i>) de software
Plano	Plano de aceitação Plano de desenvolvimento Plano de documentação Plano de gerenciamento de projeto (cronograma mestre)
Roteiro	Procedimento de teste de unidade de software Documentação de usuário
Relatório	Relatório de avaliação
Especificação	Contrato Especificação de requisitos de software Especificação de requisitos de sistema

Fonte: Elaborado pelo autor

9.6 Recuperação da informação em ciclo de vida de software

Um repositório digital é um local onde são depositados recursos de informação e/ou metadados sobre recursos de informação (MILLER, 2011). Em processos de desenvolvimento de software, diversos recursos de informação são geralmente organizados e armazenados em repositórios digitais (IEEE, 2012a). Por exemplo, no desenvolvimento de software aberto, os repositórios de software registram conhecimento acumulado pelos participantes no desenvolvimento de software e são usados para anunciar produtos, documentar produtos, prover auxílio a usuários, registrar solicitações de modificações, registrar notificações de defeitos, prover suporte a desenvolvedores, armazenar códigos de software e registrar comunicação entre participantes (DILLON; SIMMONS, 2008; HASSAN, 2008; SIMMONS; DILLON, 2006). Existem diferentes classes de repositórios de software, onde

são armazenados códigos de software, formulários, definições de interfaces, solicitações de requisitos, notificações de defeitos, mensagens trocadas por desenvolvedores, recursos destinados a auxiliar usuários finais e a prover suporte à colaboração entre desenvolvedores. [Hassan \(2008\)](#) relaciona as seguintes classes de repositórios de software: repositório histórico, repositório *run-time* e repositório de código. Os repositórios históricos armazenam recursos sobre evoluções dos projetos. Os repositórios *run-time* armazenam recursos sobre execuções e usos de produtos de software. Os repositórios de código armazenam códigos de software. [Sun et al. \(2015\)](#) relacionam as seguintes classes de repositórios de software: repositório de controle de código fonte, repositório de defeitos (*bug*) e arquivo (*archive*) de comunicação. Os repositórios de controle de código fonte armazenam histórico de desenvolvimento do software e recursos de informação sobre mudanças feitas ao código. Os repositórios de defeitos (*bug*) registram históricos de soluções de defeitos ou solicitações de características (*feature*) por usuários ou desenvolvedores. Finalmente, são registradas discussões e comunicações sobre projetos nos repositórios da classe arquivo de comunicação.

Em organizações voltadas ao desenvolvimento de software, a quantidade de tempo gasto na recuperação da informação revela frequente interação com informação. Facilitar o acesso a recursos de informação em repositórios de software contribui para aumentar a produtividade e a qualidade em projetos de software. Para serem úteis, esses repositórios precisam prover meios para recuperação da informação. Em ciclo de vida de software, técnicas de recuperação da informação podem ser usadas na solução de problemas variados. Por exemplo, é importante a recuperação da informação nos seguintes contextos: localização de requisitos de software, rastreabilidade entre recursos de informação, reuso de software, realização de medidas sobre sistemas de software ([BINKLEY; LAWRIE, 2010](#)), recuperar informação em fóruns ([GOTTIPATI; LO; JIANG, 2011](#)), recuperar modelos de software ou fragmentos desses modelos ([BISLIMOVSKA et al., 2012](#); [ROBLES et al., 2012](#)), recuperar informação sobre componente de software ([BRAGA; WERNER; MATTOSO, 2006](#); [BRITO et al., 2009](#); [ICHII et al., 2009](#); [ILK et al., 2010](#); [MARTINS et al., 2009](#); [NIE; ZHONG, 2009](#); [SILVESTRI et al., 2006](#); [SINGH, 2013](#); [SRINIVAS; RADHAKRISHNA; RAO, 2014](#); [SUN; MIAO; CAO, 2006](#); [VANDERLEI et al., 2007](#); [YAO; ETZKORN; VIRANI, 2008](#); [ZHANG et al., 2012](#)) e recuperar informação sobre defeitos (*bug*) em software ([BORG, 2014](#); [CAVALCANTI et al., 2014](#); [CORREA; SUREKA, 2013](#); [RAO; MEDEIROS; KAK, 2013](#)).

A recuperação da informação em repositórios de software é frequentemente realizada por meio da comparação entre termos em consultas e termos em descrições de software, em códigos de software ou em comentários em códigos de software. Por exemplo, motores de busca frequentemente tratam códigos como textos onde a semântica de cada termo é desconhecida. Nesse caso, a coincidência entre o termo na consulta e o termo localizado não garante a relevância do recurso recuperado ([MCMILLAN et al., 2012](#)). [Tang, Liang e Vliet \(2011\)](#) observam que existem dificuldades na recuperação da informação em

desenvolvimento de software e sugerem que os recursos de informação sejam indexados por meio de ontologias leves e de fácil uso. Existem diversas abordagens para indexar recursos de informação em repositórios de software, por exemplo, por meio de palavras-chave extraídas dos recursos de informação, indexação facetada ou rede semântica. Na primeira abordagem, são extraídas palavras-chave dos recursos de informação e criados índices associando-se palavras-chave aos recursos de informação. As palavras-chave usadas nas consultas são comparadas a palavras-chave usadas como índices. Essa abordagem tem a vantagem da simplicidade, mas a falta de semântica associada às palavras-chave impossibilita determinar se diferentes palavras-chave representam um mesmo conceito. Na segunda abordagem, os especialistas extraem palavras-chave de recursos de informação, as organizam em facetadas integrantes de esquemas de classificação e as usam como descritores dos recursos de informação. Essa abordagem é reportada como efetiva, mas requer grande quantidade de trabalho e a avaliação das distâncias conceituais entre termos, o que depende de fatores como experiência e senso comum. Finalmente, a terceira abordagem consiste em prover representação do conhecimento por meio da definição de relações semânticas entre conceitos. Essa abordagem possibilita realizar inferências, mas geralmente apresenta as seguintes desvantagens: requer elevada quantidade de trabalho; provê suporte a domínios de aplicação restritos; e apresenta elevado nível de rigidez (OSTERTAG et al., 1992). Por fim, em projetos de software, pode ser relevante reusar componentes de software. Um componente de software é parte de sistema de software, engloba funcionalidades, interfaces e implementação. Componentes de software podem ser organizados em coleções em repositórios. Nesse contexto, para que componentes de software sejam reusados é necessário recuperá-los dos repositórios onde eles estão armazenados. Existem diversas abordagens para facilitar a recuperação de componentes de software armazenados em repositórios. Em Nidhi e Jatain (2014), são relacionadas diversas abordagens, por exemplo, classificação dos componentes segundo facetadas.

9.7 Representação da informação em ciclo de vida de software

Em Engenharia de Software, a representação da informação é importante na implementação de práticas necessárias a certos níveis de maturidade de processos; na normalização de termos usados em projetos de software; no desenvolvimento de processos em ciclo de vida de software; na educação em Engenharia de Software; na tarefa “definir conteúdo, semântica, formatos e meios para representar, reter, transmitir e recuperar informação”, definida na norma ISO/IEC 12207:2008: *Systems and software engineering - Software life cycle processes* e parte do “Processo de Gestão da Informação” (*Information Management Process*); no desenvolvimento de sistemas de recuperação da informação em ciclo de vida de software; e no reuso de software. Representação da informação é também relevante em atividades no Grupo de Atividades de Planejamento de Projeto (*Project*

Planning Activity Group) e em atividades no Grupo de Atividades de Desenvolvimento de Documentação (*Documentation Development Activity Group*) descritos em [IEEE \(2006\)](#), por exemplo, no Grupo de Atividades de Planejamento de Projeto e no Grupo de Atividades de Desenvolvimento de Documentação. O Grupo de Atividades de Planejamento de Projeto é responsável pelo planejamento na gestão de projeto. Nesse grupo, é encontrada a atividade Planejar Documentação (*Plan Documentation*), que é responsável por produzir descrições, requisitos, cronogramas e restrições a serem assimiladas na definição de abordagem consistente e disciplinada para se produzir documentação requerida. Por sua vez, o Grupo de Atividades de Desenvolvimento de Documentação é responsável por planejamento, desenho (*design*), implementação, edição, produção, distribuição e manutenção de documentos necessários a desenvolvedores e usuários. As atividades nesse grupo visam prover, em tempo hábil, documentação de software aos usuários. Atividades nesse grupo ocorrem em várias fases de processo de ciclo de vida de projeto de software. Nesse grupo, a atividade Implementar Documentação (*Implement Documentation*) é responsável por desenho (*design*), preparação e manutenção de documentação. Por sua vez, a atividade Produzir e Distribuir Documentação (*Produce and Distribute Documentation*) é responsável por prover informação necessária à audiência pretendida e engloba, por exemplo, gerir arquivos eletrônicos.

9.8 Vocabulários controlados em Engenharia de Software

No domínio da Engenharia de Software, foram desenvolvidos diversos vocabulários controlados. Esses vocabulários apresentam estruturas variadas, contêm termos que representam conceitos em diferentes áreas de conhecimento dessa disciplina e podem ser usados com objetivos variados. [Borges et al. \(2013\)](#) contém resultado de mapeamento de ontologias para suporte a desenvolvimento distribuído de software; [Ruiz e Hilera \(2006\)](#) relacionam ontologias sobre o domínio da Engenharia de Software e ontologias sobre as áreas de conhecimento Requisitos de Software, Desenho (*design*) de Software, Manutenção de Software, Qualidade de Software, Processo de Engenharia de Software e Gestão de Engenharia de Software. Ainda se destacam a norma ISO/IEC/IEEE 24765 *Systems and Software Engineering Vocabulary* (vide [ISO \(2010\)](#)) sobre vocabulário em Engenharia de Software e em Engenharia de Sistemas, e o *SEVOCAB* (vide [IEEE \(2016\)](#)), projeto desenvolvido por *IEEE Computer Society* e ISO/IEC JTC 1/SC7 que possibilita o acesso a vocabulário sobre Engenharia de Software e Engenharia de Sistemas por meio da Internet.

9.8.1 Esquemas de classificação da informação em Engenharia de Software

Segundo [Damiani, Fugini e Bellettini \(1999\)](#), a classificação de componentes de software pode contribuir para o sucesso de projetos de reuso de software, particularmente, quando o reuso engloba outros artefatos além de código e facilitar compreensão de código

em processo de engenharia reversa, de dependências e de padrões de uso. Em repositórios de software, esquemas de classificação podem ser usados para classificar recursos de informação. Prieto-Diaz (1990) relaciona as seguintes características como desejáveis a esquemas de classificação destinados à classificação de componentes de software: acomodar expansão da coleção, possibilitar localização de componentes similares, possibilitar localização de componentes funcionalmente equivalentes entre domínios, precisão, elevado poder descritivo, facilidade de manutenção, facilidade de uso e poder ser automatizado. Segundo Prieto-Diaz (1990), usar vocabulários controlados nesse contexto pode ser interessante pelos seguintes motivos: códigos de componentes de software geralmente têm pouca quantidade de texto; significados das palavras-chave em componentes de software geralmente decorrem de convenções ou preferências de desenvolvedores; nem sempre é óbvio aquilo que os componentes de software fazem, ou como fazem; intervenção humana é geralmente necessária para extrair índices significativos para componentes de software (PRIETO-DIAZ, 1990). Prieto-Diaz (1990) e Albrechtsen (1992) descrevem esquemas de classificação de software e Albrechtsen (1992) compara esquemas. Apesar do exposto, vale destacar que existem questionamentos se o uso de vocabulários controlados melhora o desempenho da recuperação de componentes de software armazenados em repositórios. Mili et al. (1997) destacam que a busca por componentes de software ocorre em duas fases. Na primeira fase, a busca é exploratória e o uso de vocabulário controlado pode ser um fator restritivo. Na segunda fase, são selecionados componentes dentre os potencialmente úteis. Nessa fase, as descrições dos componentes precisam ser mais detalhadas do que as descrições geralmente providas por meio de vocabulários controlados.

9.8.2 Taxonomias e ontologias em Engenharia de Software

Existem diversas taxonomias no domínio da Engenharia de Software. Essas taxonomias são compostas por termos que representam conceitos em áreas de conhecimento da Engenharia de Software como Requisitos de Software, Desenho (*design*) de Software, Construção de Software, Manutenção de Software, Gestão de Engenharia de Software, Processo de Engenharia de Software, e Qualidade de Software. Algumas delas abordam conceitos em diversas áreas de conhecimento. No domínio da disciplina Engenharia de Software, existem ontologias desenvolvidas com o objetivo de modelar subdomínios e ontologias desenvolvidas com o objetivo de modelar todo o domínio dessa disciplina (CESARE et al., 2010; HILERA; FERNÁNDEZ-SANZ, 2010). Hilera e Fernández-Sanz (2010) destacam dificuldades na integração de ontologias que modelam subdomínios da disciplina Engenharia de Software, destacam que projetos de desenvolvimento de ontologias que visam modelar todo domínio dessa disciplina são complexos e tendem a consumir muitos recursos. Existem diversas abordagens que envolvem o uso de ontologias em Engenharia de Software, tais como: desenvolvimento guiado por ontologias (*ontology-driven development*), que consiste em usar ontologia em tempo de desenvolvimento; desenvolvimento capacitado por

ontologia (*ontology-enabled development*), que consiste em usar ontologia com o objetivo de prover suporte aos desenvolvedores nas suas tarefas; arquitetura baseada em ontologia (*ontology-based architecture*), que consiste em usar ontologia como artefato em tempo de execução (as ontologias integram lógicas de aplicações); e arquitetura capacitada por ontologia (*ontology-enabled architecture*), que consiste em usar ontologia com o objetivo de prover suporte de infraestrutura em tempo de execução (AHMED, 2008; HAPPEL; SEEDORF, 2006).

No domínio da Engenharia de Software, ontologias são usadas para organização da informação, compartilhamento de conhecimento, localização de recursos, entendimento de relações entre conceitos, comunicação, consistência, interoperabilidade e reuso; especificação, representação e gestão de requisitos de software; descrição e modelagem de domínios; descrição de regras de negócio; descrição de serviços; descrição de relações e dependências entre artefatos; descrição de componentes de software e de dependências entre os mesmos; mapeamento de modelos em códigos; documentação de interfaces de programação de aplicação (*application programming interface*); documentação de código; descrição de relações entre defeitos (*bug*) e códigos; descrição de domínio em formato processável por máquina para geração de teste; suporte à transformação entre modelos; representação de padrões (*pattern*) de software; implementação de software; estruturação e classificação de recursos de informação em desenvolvimento de software; promoção de terminologia comum entre interessados (*stakeholder*); harmonização de vocabulários usados por grupos responsáveis por desenvolver normas em Engenharia de Software, tais como grupos de trabalho do comitê técnico ISO/IEC JTC 1/SC 7 (AHMED, 2008; AMBROSIO et al., 2004; GASEVIC; KAVIANI; MILANOVIC, 2009; HAPPEL; SEEDORF, 2006; MENDES; ABRAN, 2004; RUIZ; HILERA, 2006).

Existem diversos esquemas de classificação de ontologias em Engenharia de Software. Em Wongthongtham et al. (2005), é sugerido que ontologias nesse domínio podem ser genéricas ou específicas. As genéricas englobam conceitos em Engenharia de Software, enquanto as específicas englobam conceitos em projetos específicos de software. Em Ruiz e Hilera (2006), são sugeridas as classes “ontologia de domínio” e “ontologia de artefato de software”. As ontologias de domínio visam descrever o domínio ou subdomínios da Engenharia de Software. Entre as fontes usadas no desenvolvimento dessas ontologias, são encontrados glossários, guias e livros sobre o corpo de conhecimento da Engenharia de Software. Em Zhao, Dong e Peng (2009), é sugerida a seguinte classificação: ontologia de processo de software, ontologia de domínio de aplicação, ontologia de modelo de características de domínio de aplicação, ontologia de comportamento de sistema, ontologia de arquitetura de software, ontologia de lógica de aplicação, ontologia de desenho (*design*) orientado a objetos, ontologia de padrões (*pattern*), ontologia de artefatos de software, ontologia de códigos fonte orientados a objetos, ontologia de versão, ontologia de configuração de sistema, ontologia de documentação, ontologia de qualidade, ontologia de teste,

ontologia de processo de manutenção de software e ontologia de tecnologia. Ontologia de processo de software relaciona atividades, artefatos, fases de processo, modelo de processo etc. Ontologia de domínio de aplicação engloba conceitos sobre domínio de aplicação e informação relevante a desenvolvimento de software em certo domínio. Ontologia de modelo de características de domínio de aplicação engloba características de software em certo domínio de aplicação. Ontologia de comportamento de sistema modela comportamento de sistema e ações do sistema em certos cenários. Ontologia de arquitetura de software engloba conceitos em estilos de arquitetura, componentes em estilos de arquitetura, interações entre componentes. Ontologia de lógica de aplicação engloba conceitos na modelagem de lógica de software. Ontologia de desenho (*design*) orientado a objetos engloba conceitos no desenho (*design*) de software orientado a objetos. Ontologia de padrões (*pattern*) provê catálogo de padrões. Ontologia de artefatos de software engloba conceitos que possibilitam classificar artefatos em função, por exemplo, de formatos e estruturas internas. Ontologia de código fonte orientado a objetos engloba conceitos em programação orientada a objetos. Ontologia de versões modela relações entre arquivos, *releases* e revisões de projetos de software. Ontologia de configuração de sistema modela restrições de componentes e versões. Ontologia de documentação engloba conceitos sobre documentos de software. Ontologia de qualidade engloba características e medidas de qualidade. Ontologia de teste engloba conceitos sobre teste de software. Ontologia de defeitos engloba conceitos sobre defeitos em software. Ontologia de processo de manutenção de software engloba conceitos sobre processo de manutenção de software. Finalmente, ontologia de tecnologia engloba conceitos sobre tecnologias, ambientes, plataformas e ferramentas (ZHAO; DONG; PENG, 2009).

9.9 Metadados em Engenharia de Software

Existem vários contextos nos quais metadados podem ser úteis na Engenharia de Software, por exemplo, quando do desenvolvimento de software baseado em tecnologia de componentes (*component based development*), e quando é necessário descobrir artefatos que possam ser reusados e que atendam a requisitos definidos. Para facilitar a descoberta de artefatos em ciclo de vida de projeto de software, metadados podem, por exemplo, ser armazenados em repositórios de software (ELIAS et al., 2006). A seguir, são relacionadas fontes com informação sobre uso de metadados no domínio da Engenharia de Software: em Vegi, Lisboa-Filho e Cromptvoets (2012), é encontrado perfil de aplicação *Dublin Core* desenvolvido para descrição de padrões de análise (*analysis pattern*), que capturam abstrações em modelagem de software; em Baghdadi (2009) são encontrados metadados para arquitetura de serviços *Web*, desenvolvidos para representar serviços *Web* nas perspectivas especificação de serviço, implantação de serviço, serviço como invólucro de serviços legados e serviço como unidade de composição de software que implementa processo de negócio; Rodriguez et al. (2009) contém proposta de uso de metadados *Dublin*

Core para classificar publicações em Engenharia de Software; em [Ma, Wang e Li \(2007\)](#), são definidos metadados para descrever componentes de software; em [Cazzola et al. \(2007\)](#), são definidos metadados para destacar atividades de refatoração (*refactoring*) em códigos; em [Orso et al. \(2007\)](#), é descrito uso de metadados de componentes de software em testes de regressão; em [González e Meer \(2004\)](#), o *Dublin Core Metadata Element Set* é adaptado para descrever componentes de software, para facilitar a recuperação de componentes armazenados em repositórios; [OMG \(2005\)](#) contém *Reusable Asset Specification* (RAS), uma especificação para descrição de ativos reusáveis de software, em que um ativo pode ser composto por múltiplos artefatos; [Elias et al. \(2006\)](#) contém modelo de metadados para descrição de ativos em repositórios distribuídos e compartilhados de componentes de software denominado X-ARM; [Hoff, Partovi e Thai \(1997\)](#) contém modelo denominado *The Open Software Description Format* (OSD) para descrever pacotes de software e relações entre esses pacotes.

9.10 Melhoria da qualidade de software

Na garantia da qualidade, destacam-se os modelos de qualidade. Segundo [ISO \(2010\)](#), modelo de qualidade é “conjunto definido de características e de relações entre elas, que provê a base para especificação de requisitos de qualidade e avaliação da qualidade”. No contexto da Engenharia de Software, [ABNT \(2003\)](#) contém proposta de modelo de qualidade de produto. [Moody \(2005\)](#) relaciona as seguintes características desse modelo de qualidade: estrutura hierárquica, termos usados para identificar características e subcaracterísticas são termos também usados na prática, características e subcaracterísticas apresentam definições concisas, medidas são definidas para as subcaracterísticas, e norma separada define procedimentos para realização de avaliações de produtos. Em [ABNT \(2003\)](#), são descritas características de qualidade em uso, características de qualidade externa e características de qualidade interna. Qualidade em uso está relacionada às necessidades de qualidade dos usuários. É a qualidade do ponto de vista do usuário. Qualidade externa está associada à qualidade requerida do ponto de vista externo. É a totalidade das características do produto do ponto de vista externo. Qualidade interna está associada à qualidade requerida do ponto de vista interno do produto. É a totalidade das características do produto do ponto de vista interno.

9.10.1 Modelos de qualidade de processo de software

Um meio de melhorar produtividade e qualidade em projeto é melhorar a qualidade de processos usados nos projetos. Com o objetivo de auxiliar na melhoria da qualidade de processos, têm sido desenvolvidos modelos de referência, modelos de avaliação e modelos de capacidade. Um modelo de referência de processo é geralmente composto por declaração do domínio do modelo; descrições de processos; descrições de objetivos e resultados de

processos; descrição de relação entre modelo e contexto de uso; descrições de relações entre processos, caracterização e especificação da comunidade de interesse e ações realizadas para alcançar consenso na comunidade (GALIN, 2003; ISO, 2008a; ISO, 2010; CMMI PRODUCT TEAM, 2010). Modelos de avaliação de processos têm o objetivo de avaliar capacidades de processos levando em consideração modelos de referência (IEEE, 2016). Em modelos de referência, modelos de avaliação e modelos de capacidade, a descrição de cada processo geralmente inclui propósitos e resultados dos processos. O conjunto de resultados do processo deve ser necessário e suficiente aos objetivos do processo. Um processo é considerado capaz se satisfaz objetivos de qualidade de produto, de qualidade de serviço e de qualidade de desempenho (IEEE, 2016; ISO, 2010).

9.10.1.1 Modelos de maturidade de capacidade

Um modelo de maturidade de capacidade (*capability maturity model*) é um modelo composto por elementos considerados essenciais para que processos sejam considerados efetivos e que descreve caminho para melhoria de processos em uma ou mais disciplinas (ISO, 2010). Em desenvolvimento de software, um relevante modelo de maturidade de capacidade é o *Capability Maturity Model Integration* (CMMI). Esse modelo é composto por melhores práticas para auxiliar organizações a melhorarem seus processos. No CMMI, os caminhos para melhoria de processos são descritos por definições de níveis. Para alcançar um determinado nível, a organização precisa satisfazer metas de área de processo ou de conjunto de áreas de processo. A melhoria e a avaliação de processos podem ser abordadas por meio de representação contínua (*continuous representation*) ou por meio de representação por estágio (*staged representation*). A representação contínua é destinada a organização que procura melhorar processos em determinadas áreas ou grupo de áreas. A representação por estágio é destinada a organização que procura melhorar conjuntos de processos relacionados por meio da melhoria de conjuntos sucessivos de áreas. O *Capability Maturity Model Integration for Development* (CMMI-DEV) é um modelo de referência que engloba práticas de desenvolvimento de produtos e serviços. Essas práticas abrangem o ciclo de vida de produtos. O CMMI-DEV descreve metas (*goal*), práticas e áreas de processo. As metas descrevem características que devem estar presentes para satisfazer áreas de processos. As práticas descrevem atividades importantes para alcançar metas. O CMMI-DEV engloba 22 áreas de processo. Essas áreas de processo abordam conceitos associados à melhoria de processos, e englobam práticas que, ao serem implementadas, satisfazem metas para melhoria na área. As áreas de processos são organizadas nos seguintes grupos: Gerenciamento de Processo, Gerenciamento de Projeto, Engenharia e Suporte (CMMI PRODUCT TEAM, 2010).

Em CMMI-DEV, dado é definido como “informação registrada” e gestão de dados é definida como “processos disciplinados e sistemas que planejam, obtêm e cuidam de dados técnicos e de negócio, de modo consistente com requisitos de dados, ao longo do ciclo de

vida dos dados”. Entre as metas da área de processo “Planejamento de Projeto”, se encontra “Desenvolver Plano de Projeto”. Entre as metas da área de processo “Monitoramento e Controle de Projeto”, se encontra “Monitorar o Projeto em Relação ao Plano”. Associada à meta “Desenvolver Plano de Projeto”, se encontra a prática “Planejar Gestão de Dados”. Por sua vez, associada à meta “Monitorar o Projeto em Relação ao Plano”, se encontra a prática “Monitorar a Gestão de Dados”. Em relação à prática “Planejar Gestão de Dados”, vale destacar a subprática “Estabelecer um mecanismo para arquivar dados e acessar dados arquivados”. Entre os produtos de “Planejar Gestão de Dados”, vale destacar “Descrição de conteúdo e formato de dado” e “Mecanismos para recuperação, reprodução e distribuição de dados” (CMMI PRODUCT TEAM, 2010).

O Modelo de Referência para Melhoria do Processo de Software Brasileiro é outro modelo de referência de processo. Esse modelo leva em consideração normas e modelos internacionalmente reconhecidos, boas práticas da Engenharia de Software e necessidades de negócio da indústria de software no Brasil. Os modelos MPS são descritos em documentos no formato de guias. SOFTEX (2016b) contém descrição geral do Modelo MPS e detalhamento do Modelo de Referência MPS para Software (MR-MPS-SW). O MR-MPS-SW define níveis de maturidade, relações entre processos e suas capacidades. A capacidade do processo, é representada por um conjunto de atributos de processo, é definida como “a caracterização da habilidade do processo para alcançar os objetivos de negócio, atuais e futuros; estando relacionada com o atendimento aos atributos de processo associados aos processos de cada nível de maturidade”. O nível de capacidade de processo é descrito por atributos de processo (AP). Entre os processos descritos em SOFTEX (2016b), se encontra o processo Gerência de Projetos (GPR), cujo propósito é “estabelecer e manter planos que definem as atividades, recursos e responsabilidades do projeto, bem como prover informações sobre o andamento do projeto que permitam a realização de correções quando houver desvios significativos no desempenho do projeto”. Entre os resultados esperados do processo, se encontra o seguinte: “os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição, um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança”. Em SOFTEX (2016a), dado de projeto são definidos como formas de documentação exigidas para sua execução. Esses dados podem estar em qualquer formato e meio. SOFTEX (2016a) também contém orientações para implementar o nível G nas organizações de acordo com o MR-MPS-SW. Nesse nível, se encontra o processo GPR e o resultado esperado anteriormente descrito. Para alcançar esse resultado, é destacada a importância da identificação de dados relevantes para que sejam coletados, armazenados e distribuídos de modo controlado.

Parte III

Resultados

Prólogo

Esta parte da tese contém capítulos onde são descritos elementos de arcabouço de arquitetura da informação propostos e exemplo de uso de alguns desses elementos. O capítulo 10, onde são propostos elementos de arcabouço de arquitetura da informação para ciclo de vida de projeto de vocabulário controlado, contém informação sobre processo de desenvolvimento, contexto de aplicação, limites, diretrizes, arquitetura de referência, modelo de domínio, modelo de qualidade e lista de atividades. Os elementos propostos são embasados em fontes identificadas na pesquisa bibliográfica realizada. O capítulo 11, onde é exemplificado o uso de elementos do arcabouço proposto nesta tese, contém informação sobre ciclo de vida de projeto, definição de domínio, definição de necessidades de informação, definição de idioma, definição de garantia semântica, definição de fontes de informação, construção de lista de termos candidatos, definição de facetas e relações, definição de conceitos, controle de qualidade e implementação de protótipo de vocabulário controlado.

10 Proposta de elementos de arcabouço

Neste capítulo, são propostos elementos de arcabouço de arquitetura da informação para ciclo de vida de projeto de vocabulário controlado. Este capítulo contém considerações sobre a pesquisa bibliográfica, definição do termo “arcabouço de arquitetura da informação”, proposta de elementos de arcabouço, contexto de aplicação e limites, proposta de arquitetura de referência, modelo de domínio, modelo de qualidade e lista de atividades.

10.1 Considerações sobre a pesquisa bibliográfica

Na pesquisa bibliográfica realizada, não foi constatado consenso sobre abordagem para desenvolvimento e avaliação de vocabulários controlados, embora alguns arcabouços, processos e métodos propostos para essas finalidades frequentemente apresentem elementos em comum. A pesquisa bibliográfica também possibilitou as seguintes constatações: vocabulário controlado pode ser desenvolvido e avaliado por meio de processos definidos; qualidade de processo influencia qualidade de produto; qualidade de vocabulário controlado pode ser caracterizada por grau de conformidade entre características do vocabulário controlado e requisitos de interessados (*stakeholder*) no vocabulário; característica de qualidade têm relação com ponto de vista (*viewpoint*) de interessado (*stakeholder*). Por fim, foi possível constatar que alguns arcabouços, processos e métodos para desenvolvimento e avaliação de vocabulários controlados, apresentam características indesejáveis como: uso de termos para os quais não são propostas definições; falta de acordo, com outros arcabouços, processos e/ou métodos, sobre uso de termos e definições; critérios de qualidade não mutuamente exclusivos; conjunto de critérios de qualidade não exaustivo; falta de propostas de ferramentas para implementação do arcabouço, processo ou método; falta de informação sobre contexto e/ou limite de uso do arcabouço, processo ou método; dependência quanto a contexto de uso; falta de referências à literatura correlata, de coerência com campos correlatos e/ou com normas; falta de exemplos de uso; falta de alinhamento com ferramentas semânticas existentes.

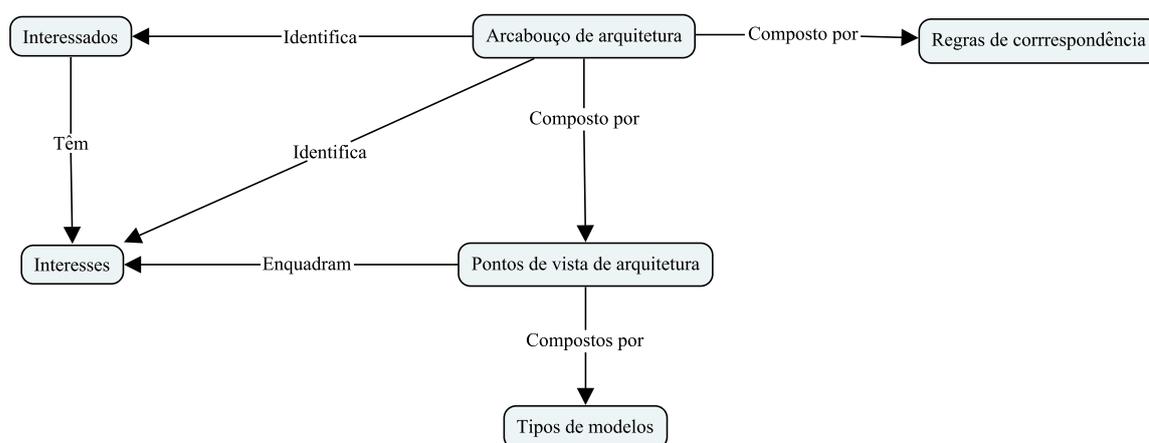
10.2 Definição de arcabouço de arquitetura da informação

Nesta tese, é proposta a seguinte definição para o termo “arcabouço de arquitetura da informação”: é arcabouço de arquitetura que enfoca a “atenção à arquitetura da informação”. Também é proposto que a “atenção à arquitetura da informação” seja caracterizada pela presença, no arcabouço de arquitetura da informação, de elementos mapeáveis para camadas do modelo descrito em [Albuquerque e Lima-Marques \(2011\)](#). A

definição anteriormente proposta é embasada em definição de “arcabouço de arquitetura” (*architectural framework*).

Nesse contexto, arcabouço de arquitetura designa estrutura conceitual ou estruturas conceituais para desenvolvimento, implementação e manutenção de arquiteturas. Um arcabouço de arquitetura aborda pontos de vista relevantes a interessados (*stakeholder*), estabelece convenções, princípios e práticas para descrição de arquiteturas em domínio e/ou comunidade, e pode restringir relações entre elementos da arquitetura. Pode também relacionar e/ou descrever domínios, escopos, vocabulários, estratégias, modelos, métodos, processos, ferramentas, arquiteturas de referência, recomendações, pontos de vista, artefatos e resultados (GREEFHORST; KONING; VLIET, 2006; ISO, 2011e; LANKHORST, 2012; MALLOY et al., 2010; The Open Group, 2011; BERNARD, 2012; PMI, 2013). Finalmente, a figura 34 é um mapa de conceitos sobre arcabouço de arquitetura, construído, pelo autor desta tese, a partir de modelo proposto em ISO (2011e).

Figura 34 – Elementos de arcabouço de arquitetura



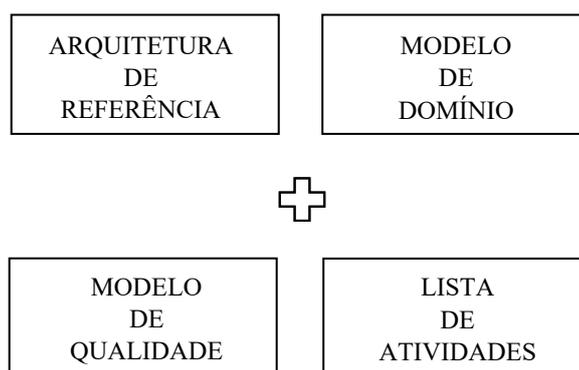
Fonte: Elaborado pelo autor a partir de ISO (2011e)

10.3 Elementos de arcabouço

Considerando os objetivos desta pesquisa, constatações decorrentes da pesquisa bibliográfica e benefícios decorrentes da existência de arcabouços de arquitetura em outros domínios (vide ISO (2011e) e Malloy et al. (2010)), são propostos elementos de arcabouço de arquitetura da informação para ciclo de vida de projeto de vocabulário controlado, em particular, para suporte a processo de desenvolvimento e a processo de avaliação em ciclo de vida de projeto de vocabulário controlado. No desenvolvimento desses elementos, foi adotado um processo de desenvolvimento definido e observadas fontes de informação

referenciadas nesta tese. Os elementos propostos são os seguintes: (a) arquitetura de referência, (b) modelo de domínio, (c) modelo de qualidade e (d) lista de atividades. Esses elementos foram desenvolvidos por meio de processo descrito na seção 2.3. Nesse arcabouço, a atenção à arquitetura da informação é caracterizada pela presença de elementos mapeáveis para camadas do modelo descrito por [Albuquerque e Lima-Marques \(2011\)](#). A figura 35 ilustra os elementos de arcabouço propostos nesta tese.

Figura 35 – Elementos de arcabouço de arquitetura da informação



Fonte: Elaborado pelo autor

10.4 Contexto de aplicação e limites

Os elementos de arcabouço de arquitetura da informação propostos enfocam a qualidade de esquema de vocabulário controlado. Nesse contexto, qualidade de esquema de vocabulário controlado é aqui definida como qualidade de termos, relações e descrições de conceitos em vocabulário controlado. Os elementos propostos enfocam processo de avaliação no qual especialista ou grupo de especialistas avalia a qualidade do vocabulário controlado. Particularmente, os elementos enfocam avaliação onde o vocabulário controlado é avaliado por meio de listas compostas por critérios previamente definidos. Quanto ao objeto do processo de análise, é o vocabulário controlado. Por fim, quanto ao momento da avaliação do vocabulário controlado, pode ocorrer durante o desenvolvimento ou posteriormente.

Com o objetivo de delimitar o escopo da pesquisa, foram identificadas apenas evidências de qualidade relacionadas à característica de qualidade “compatibilidade”, particularmente, compatibilidade de desenho (*design*) de vocabulário controlado, que é aqui definida como “característica de concordar o desenho (*design*) do vocabulário controlado com referência”. Nesse contexto, a referência com a qual o desenho (*design*) do vocabulário controlado deve concordar é definida em normas, guias de melhores práticas, livros e artigos sobre vocabulário controlado. Também com o objetivo de delimitar o

escopo da pesquisa, as características de qualidade que integram o modelo de qualidade são apenas as características de qualidade identificadas simultaneamente no quadro 10 e em fontes referenciadas na seção 8.5.2; as medidas de qualidade propostos foram identificadas em fontes de informação; não foram descritos requisitos para processos em ciclo de vida de projeto de vocabulário controlado e nem modelos de ciclo de vida de projeto, apesar desses elementos integrarem a arquitetura de referência proposta. Por fim, é importante destacar que os conjuntos de pontos de vista, de características de qualidade, de evidências de qualidade e de medidas de qualidade propostos, não são completos; e que os elementos de arcabouço propostos podem não abranger todas as classes existentes de vocabulários controlados.

10.5 Proposta de arquitetura de referência

É proposta uma arquitetura de referência para definição de processos em ciclo de vida de projeto de vocabulário controlado embasada em IEEE (2006) e composta pelos seguintes elementos: descrições de requisitos para processos em ciclo de vida de projeto de vocabulário controlado, modelos de ciclo de vida de projeto de vocabulário controlado, descrições de atividades para modelo de ciclo de vida de projeto de vocabulário controlado, e recursos que podem ser usados como ativos organizacionais em projeto de vocabulário controlado.

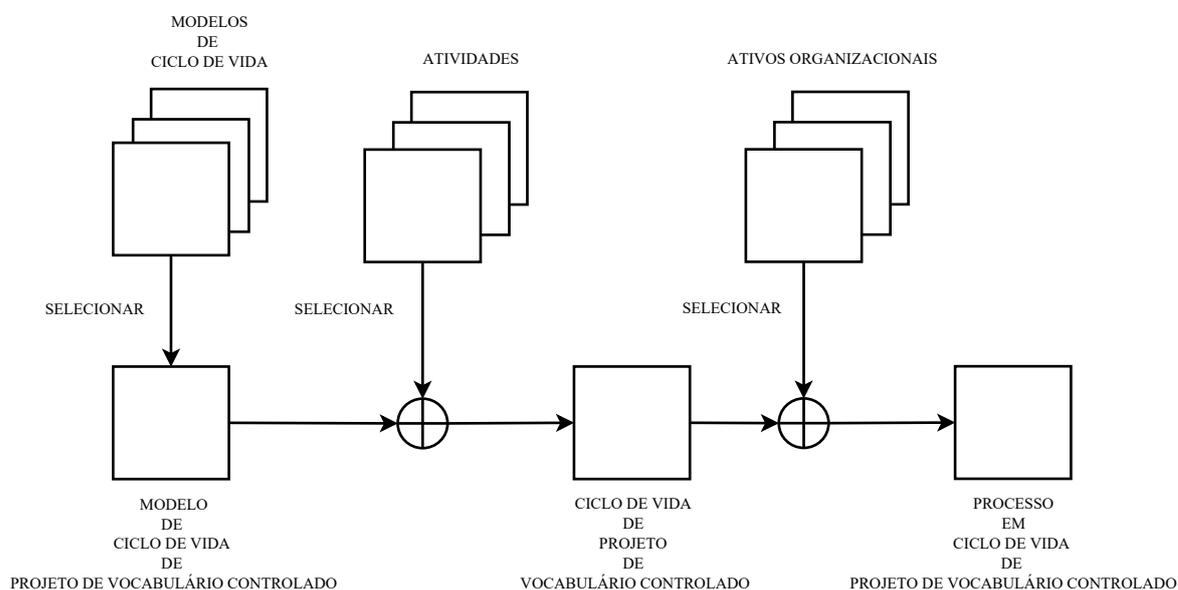
Nessa arquitetura de referência, cada atividade pode integrar processos variados em ciclo de vida de projeto de vocabulário controlado. Assim como em IEEE (2006), é sugerido que a definição de processo em ciclo de vida de projeto de vocabulário controlado ocorra por meio das seguintes atividades: definir requisitos de processo, selecionar modelo de ciclo de vida de projeto, definir ciclo de vida de projeto por mapeamento de atividades para modelo de ciclo de vida de projeto selecionado, incorporar ativos organizacionais a ciclo de vida de projeto e validar processo em ciclo de vida de projeto. A figura 36 ilustra elementos na definição de processo.

10.6 Proposta de modelo de domínio

O modelo de domínio proposto é composto por elementos em ciclo de vida de projeto de vocabulário controlado. A figura 37 representa o modelo de domínio. Nesse modelo, conceito é definido como conjunto de proposições verdadeiras sobre características de entidade. Essa definição é embasada na *Referent-oriented, Analytical Concept Theory* proposta por Ingetratur Dahlberg. Elementos dessa teoria foram anteriormente descritos na seção 3.5.1.1 desta tese.

No desenvolvimento do modelo de domínio, foram analisadas fontes de informação

Figura 36 – Definição de processo



Fonte: Elaborado pelo autor a partir de [IEEE \(2006\)](#)

referenciadas nesta tese, por exemplo, [ISO \(2000\)](#), [ISO \(2010\)](#), [Princeton University \(2010\)](#) e [European Union \(2017\)](#); e foram executadas atividades para alinhar o modelo de domínio a conteúdo da base de dados léxica *WordNet* (vide o quadro 13). As definições dos termos usados no modelo de domínio são embasadas ou originadas, principalmente, na *WordNet* (vide [Princeton University \(2010\)](#)). A seguir, são apresentadas definições para termos que designam elementos do modelo de domínio.

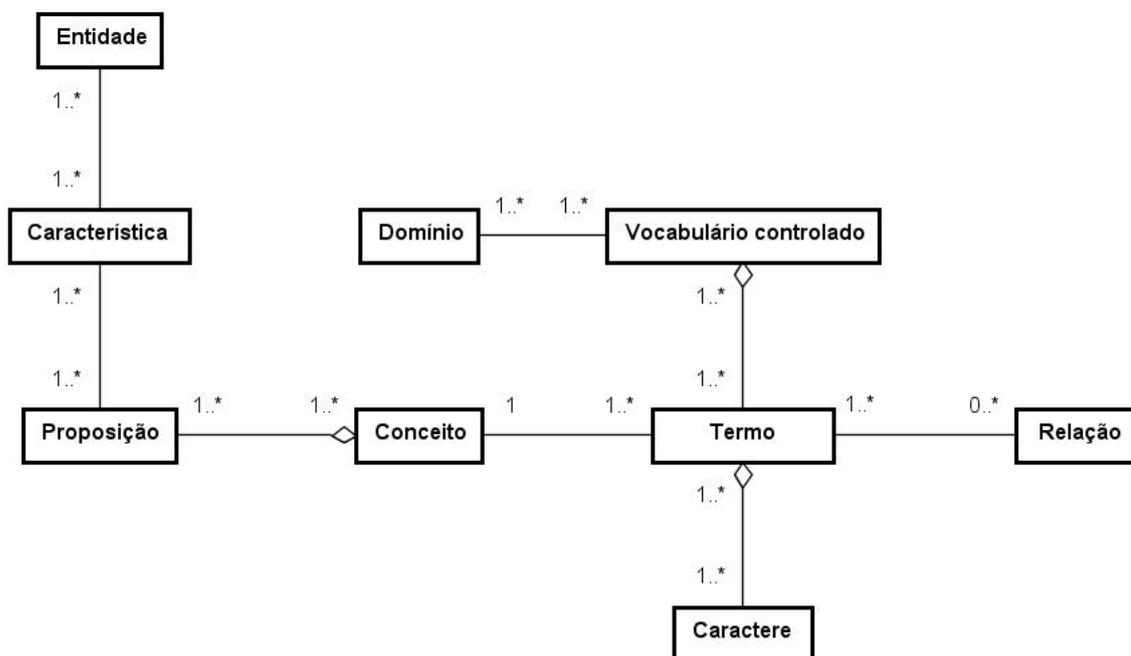
Quadro 13 – Relações entre termos no modelo de domínio e na *WordNet*

TERMOS NO MODELO DE DOMÍNIO	TERMOS HIPERÔNIMOS NA WORDNET
caractere	symbol
característica	property, attribute, dimension
conceito	set
domínio	content, cognitive content, mental object
entidade	
proposição	statement
relação	abstraction, abstract entity
termo	sequence
vocabulário controlado	system

Fonte: Elaborado pelo autor

- (a) **Caractere** é símbolo usado na escrita;
- (b) **Característica** é propriedade proeminente ou aspecto de algo;
- (c) **Conceito** é conjunto de proposições verdadeiras sobre características de entidade;

Figura 37 – Modelo de domínio



Fonte: Elaborado pelo autor

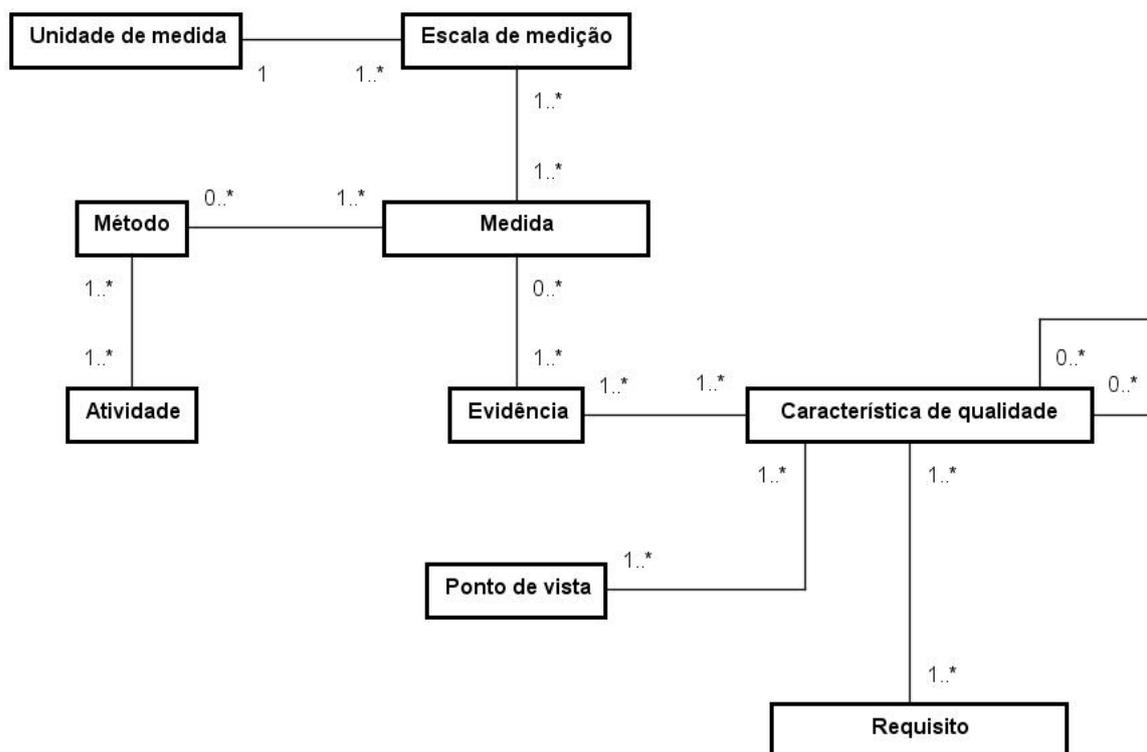
- (d) **Domínio** é conteúdo de campo de conhecimento;
- (e) **Entidade** é aquilo que é percebível ou concebível;
- (f) **Proposição** é enunciado que afirma ou nega algo;
- (g) **Relação** é abstração comum a duas entidades;
- (h) **Termo** é sequência de caracteres em linguagem natural ou artificial;
- (i) **Vocabulário controlado** é sistema de termos, relações e conceitos.

10.7 Proposta de modelo de qualidade

O modelo de qualidade proposto visa prover suporte à garantia da qualidade em ciclo de vida de projeto de vocabulário controlado. A figura 38 representa o modelo de qualidade. Nesse contexto, embasado em PMI (2013) e Wagner (2013), qualidade de vocabulário controlado é grau com o qual um conjunto de características do vocabulário controlado satisfaz requisitos, enquanto “modelo de qualidade” é definido como modelo para descrever, avaliar e/ou prever qualidade. No desenvolvimento do modelo de qualidade, foram analisadas fontes de informação referenciadas nesta tese, por exemplo, ABNT (2003),

Dromey (1996), Duque-Ramos et al. (2013), Eppler e Wittig (2000), Eppler (2001), IEEE (2009), ISO (2010), Stvilia et al. (2007) e Princeton University (2010); e foram executadas atividades para alinhar o modelo de qualidade a conteúdo da base de dados léxica *WordNet* (vide o quadro 14). As definições dos termos usados no modelo de qualidade são embasadas ou originadas, principalmente, na *WordNet* (vide Princeton University (2010)). A seguir, são apresentadas definições para termos que designam elementos do modelo de qualidade.

Figura 38 – Modelo de qualidade – Parte 1



Fonte: Elaborado pelo autor

Quadro 14 – Relações entre termos no modelo de qualidade e na *WordNet*

TERMOS NO MODELO DE QUALIDADE	TERMOS HIPERÔNIMOS NA WORDNET
atividade	event
característica de qualidade	attribute
escala de medição	standard, criterion, measure, touchstone
evidência	information
medida	abstraction, abstract entity
método	know-how
ponto de vista	position, stance, posture
requisito	thing
unidade de medida	definite quantity

Fonte: Elaborado pelo autor

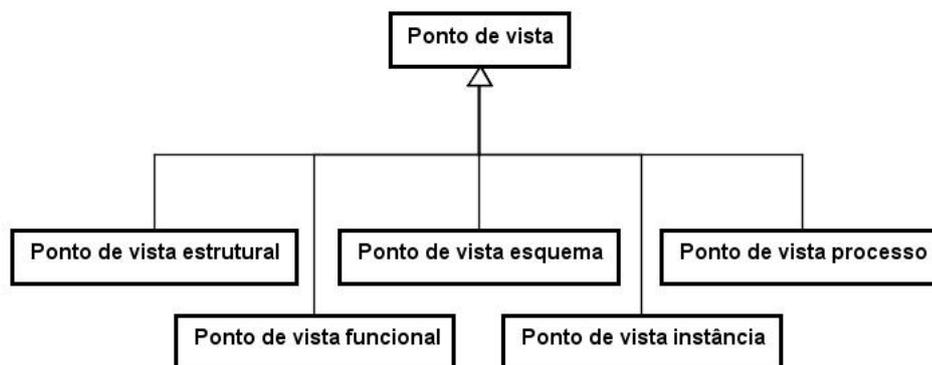
(a) **Atividade** é evento causado por pessoa;

- (b) **Característica de qualidade** é atributo distinguidor de algo;
- (c) **Escala de medição** é padrão ordenado de referência;
- (d) **Evidência** é informação que justifica crença;
- (e) **Medida** é quanto de algo que pode ser quantificado;
- (f) **Método** é modo de fazer algo;
- (g) **Ponto de vista** é posição a partir da qual se concebe ou analisa algo;
- (h) **Requisito** é algo demandado;
- (i) **Unidade de medida** é quantidade aceita como padrão de medição.

10.7.1 Pontos de vista

No modelo de qualidade proposto, características estão associadas a pontos de vista (*viewpoint*) de qualidade. Como em outros domínios do conhecimento (vide Rozanski e Woods (2011)), são considerados diferentes pontos de vista de interessados (*stakeholder*). Os pontos de vista foram identificados nas seguintes fontes de informação: Aleman-Meza et al. (2005), Gangemi et al. (2006) e Stvilia (2007). As definições propostas são originadas ou embasadas nessas fontes. A figura 39 representa hierarquia composta pelos pontos de vista que integram o modelo de qualidade. A seguir, são apresentadas definições para esses pontos de vista.

Figura 39 – Modelo de qualidade – Parte 2



Fonte: Elaborado pelo autor

- (a) **Ponto de vista estrutural** é ponto de vista que enfoca qualidade de estrutura;

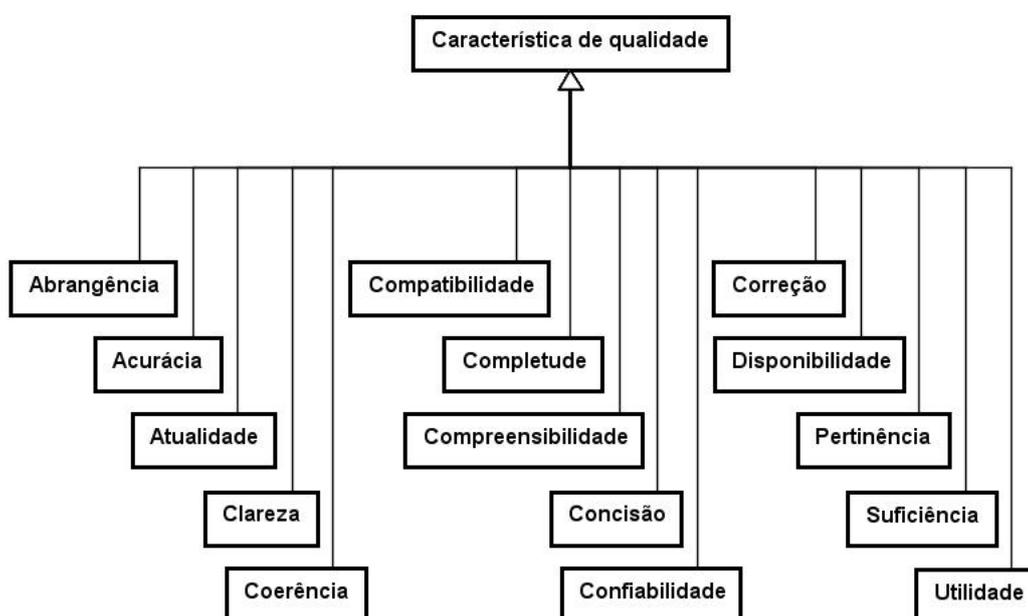
- (b) **Ponto de vista funcional** é ponto de vista que enfoca qualidade em uso;
- (c) **Ponto de vista esquema** é ponto de vista que enfoca qualidade de modelagem de domínio por esquema;
- (d) **Ponto de vista instância** é ponto de vista que enfoca qualidade de base de conhecimento;
- (e) **Ponto de vista processo** é ponto de vista que enfoca qualidade de processo.

10.7.2 Características de qualidade

As características de qualidade que integram o modelo de qualidade, são características identificadas no quadro 10 e em fontes referenciadas na seção 8.5.2. Algumas dessas características dependem da percepção do usuário, outras são inerentes à informação. As definições propostas para as características são originadas ou embasadas em fontes de informação referenciadas nesta tese, por exemplo, Arouck (2011), Burton-Jones et al. (2005), Damasevicius (2009), Duque-Ramos et al. (2013), Fox, Barbuceanu e Gruninger (1996), Gómez-Pérez, Fernández e Vicente (1996), Gómez-Pérez (1996), Hloman e Stacey (2014), ISO (2000), ISO (2010), IQ International (2017), ISTQB (2017), Kang et al. (2004), Oh e Yeom (2012), Pizzoleto e Oliveira (2016), Raad e Cruz (2015), Stvilia (2007), Tibaldo et al. (2015), Princeton University (2010), Uschold e Gruninger (1996), European Union (2017) e Zhang, Li e Tan (2010). A figura 40 representa hierarquia de características de qualidade que integram o modelo de qualidade. A seguir, são apresentadas definições para essas características que qualidade.

- (a) **Abrangência** é característica de qualidade de compreender vasto escopo;
- (b) **Acurácia** é característica de qualidade de estar próximo da verdade;
- (c) **Atualidade** é característica de qualidade de ser contemporâneo a determinado tempo;
- (d) **Clareza** é característica de qualidade de ser de fácil entendimento;
- (e) **Coerência** é característica de qualidade de não ter contradição;
- (f) **Compatibilidade** é característica de qualidade de concordar com valor de referência;
- (g) **Compleitude** é característica de qualidade de se apresentar sem faltas de elementos;
- (h) **Compreensibilidade** é característica de qualidade de comunicar o significado pretendido;
- (i) **Concisão** é característica de qualidade de se ater ao essencial;

Figura 40 – Modelo de qualidade – Parte 3



Fonte: Elaborado pelo autor

- (j) **Confiabilidade** é característica de qualidade de se poder acreditar ser acurado;
- (k) **Correção** é característica de qualidade de se apresentar sem defeitos;
- (l) **Disponibilidade** é característica de qualidade de ser disponível quando necessário;
- (m) **Pertinência** é característica de qualidade de ser de interesse;
- (n) **Suficiência** é característica de qualidade de ser adequado ao propósito;
- (o) **Utilidade** é característica de qualidade de ser de serventia.

10.7.3 Evidências

As evidências que integram o modelo de qualidade foram identificadas nas seguintes fontes de informação: Aitchison, Bawden e Gilchrist (2000), DLESE (2005), Hedden (2010), NISO (2005), Stock (2010), Breis, Duque-Ramos e Martinez (2016), Burton-Jones et al. (2005), Sicilia et al. (2012), Zhang, Li e Tan (2010), Duque-Ramos et al. (2013), Yao, Orme e Etkorn (2005), Lozano-Tello e Gómez-Pérez (2004), Strasunskas e Tomassen (2008), Gómez-Pérez (2001), Orme, Tao e Etkorn (2006), Stvilia (2007), Gavrilova, Gorovoy e Bolotnikova (2010), Zhang, Li e Tan (2010), Neuhaus et al. (2013), Gangemi et al. (2006), Aleman-Meza et al. (2005), Tibaldo et al. (2015) e Chen e Matthews (2008). As evidências

de qualidade relativas às relações associativas, foram identificadas, principalmente, em Hedden (2010).

As evidências identificadas estão relacionadas nos quadros 15, 16 e 17. Nesses quadros, linguagem natural é linguagem cujas regras são baseadas no uso, em vez de estabelecidas antes do uso da linguagem, enquanto linguagem formal é linguagem cujas regras foram explicitamente estabelecidas antes do uso da linguagem (ISO, 2010). A importância de cada evidência de qualidade depende de aspectos como a classe de vocabulário controlado.

Quadro 15 – Evidências - Parte 1

ELEMENTO	EVIDÊNCIAS
Termo	Termo designa apenas um conceito.
	Termo de acordo com grafia amplamente aceita.
	Termo de acordo com grafia em fonte de referência (dicionário, enciclopédia, glossário, etc.).
	Termo de acordo com grafia adotada no vocabulário controlado.
	Termo de acordo com sintaxe aceita.
	Termo de acordo com convenção de nome padronizada.
	Termo originado em literatura no domínio.
	Termo é encontrado em outro vocabulário controlado.
	Termo de acordo com a audiência.
	Termo de acordo com o escopo do vocabulário controlado.
	Termo de acordo com padrão (acadêmico, cultural, político, comercial, etc.).
	Termo de acordo com estilo de escrita adotado no vocabulário controlado.
	Termo é composto por poucas palavras.
	Termo é usado por pessoas familiares com o domínio do vocabulário controlado.
	Termo no idioma esperado pelos usuários do vocabulário controlado.
	Termo designa nome próprio segundo prática de catalogação.
	Termo inclui apenas caracteres necessários.
	Termo inclui diacríticos necessários.
	Termo comunica escopo pretendido a usuários do vocabulário controlado.
	Termo é único no vocabulário controlado.
	Termo é neutro.
	Termo tem definição.
	Termo e sua definição são substituíveis em sentença.
	Termo é substantivo.
	Termo designa atividade por substantivo ou por verbo no gerúndio.
	Termo designa conceito abstrato no singular.
	Termo designa entidade única no singular.
	Termo composto não designa dois princípios de divisão.
	Qualificador é usado para desambiguar termo que designa mais de um conceito.
	Qualificador é usado para desambiguar termo em vocabulário usado em diferentes domínios.
	Qualificador não é usado para representar termo composto.
	Qualificador é composto por poucas palavras.
	Qualificador não é homógrafo.
Qualificador é padronizado no vocabulário controlado.	

Fonte: Elaborado pelo autor

10.7.4 Medidas

Em vez de sugerir novas medidas de qualidade, medidas foram identificadas em fontes de informação e adequadas ao arcabouço proposto. Nesse processo, as seguintes

Quadro 16 – Evidências - Parte 2

ELEMENTO	EVIDÊNCIAS
Relação	Relação é representada de modo padronizado.
	Relação é identificada de modo padronizado.
	Relação não é redundante.
	Relação tem relação recíproca.
	Relação ocorre entre conceitos apropriados.
	Relação tem aridade especificada.
	Relação resulta em estrutura não circular.
	Relação resulta em hierarquia com herança simples.
	Relação em hierarquia faz parte de estrutura em árvore.
	Em relação hierárquica entre gênero e espécie, instâncias da espécie são do gênero.
	Em relação hierárquica entre todo e parte, a parte só pertence ao todo.
	Em relação hierárquica entre todo e parte, o todo é um sistema.
	Relação de equivalência ocorre entre abreviação e forma completa de termo.
	Relação de equivalência ocorre entre acrônimo e forma completa de termo.
	Relação de equivalência ocorre entre termo não preferido e termo preferido.
	Relação de equivalência ocorre entre termos que designam o mesmo conceito.
	Relação de equivalência ocorre entre grafias variantes aceitas.
	Relação associativa ocorre entre processo e agente.
	Relação associativa ocorre entre processo e contra-agente.
	Relação associativa ocorre entre ação e propriedade.
	Relação associativa ocorre entre ação e produto.
	Relação associativa ocorre entre ação e alvo.
	Relação associativa ocorre entre ação e paciente.
	Relação associativa ocorre entre causa e efeito.
	Relação associativa ocorre entre objeto e propriedade.
	Relação associativa ocorre entre objeto e origem.
Relação associativa ocorre entre material e produto.	
Relação associativa ocorre entre disciplina e praticante.	
Relação associativa ocorre entre disciplina e objeto.	
Relação associativa ocorre entre disciplina e fenômeno.	
Relação associativa ocorre entre parte e todo.	

Fonte: Elaborado pelo autor

fontes de informação foram analisadas: Breis, Duque-Ramos e Martinez (2016), Burton-Jones et al. (2005), Hloman e Stacey (2014), Hobel e Revenko (2016), Kang et al. (2004), Kless e Milton (2010), Martínez et al. (2011), Navarro, García-Peñalvo e Therón (2010), Orme, Tao e Etkorn (2006), Urdiciain (1998) e Yao, Orme e Etkorn (2005). O quadro 18 contém as medidas identificadas. Por meio do processo descrito, outras podem ser incorporadas ao modelo de qualidade.

Quando do uso de medida de qualidade em ciclo de vida de projeto de vocabulário controlado, a informação sobre a medida pode ser registrada em formulário padronizado. Adotando IEEE (2009) como referência, esse formulário pode conter nome da medida, custo de uso da medida, benefícios de uso da medida, impacto da medida sobre o projeto, valor da medida para alcançar o requisito de qualidade, características de qualidade relacionadas à medida, identificação de ferramentas usadas no cálculo da medida, descrição

Quadro 17 – Evidências - Parte 3

ELEMENTO	EVIDÊNCIAS
Conceito	Definição tem só um significado.
	Definição em linguagem formal coincide com significado de definição em linguagem natural.
	Definição não contém contradição.
	Definição não contradiz outra definição no vocabulário controlado.
	Definição não contradiz o mundo real.
	Definição não contradiz inferência.
	Definição é completa (condições necessárias e suficientes).
	Definição de acordo com fonte de referência (dicionário, enciclopédia, etc.).
	Definição independe de contexto computacional.
	Definição independe de contexto social.
	Definição não usa o próprio conceito sendo definido.
	Definição não usa conceito desconhecido.
	Definição não usa negação.
	Definição não expõe propriedade supérflua.
	Definição não provoca reação emocional.
	Definição não é redundante a outra definição no vocabulário controlado.
	Definição é necessária.
Definição é composta por apenas uma sentença.	
Definição é composta por proposições.	

Fonte: Elaborado pelo autor

de como usar a medida, descrição da área de aplicação da medida, identificação de dados necessários para cálculo da medida, descrição dos passos necessários para cálculo da medida, interpretação dos resultados do cálculo da medida, consideração sobre a propriedade da medida, identificação de treinamento requerido para implementação ou uso da medida, descrição de exemplo de uso da medida, identificação de projetos onde a medida foi usada, identificação de critérios de validade satisfeitos pela medida, referências para entendimento da medida, e referências relevantes à implementação da medida. Para selecionar medidas e planejar medições em ciclo de vida de projeto, pode ser usado método, por exemplo, *Goal Question Metrics* (GQM) (descrito na seção 8.5.1).

10.8 Proposta de lista de atividades

Foram listadas atividades relevantes a processo de desenvolvimento e/ou a processo de avaliação de vocabulário controlado. Essas atividades foram identificadas por meio de análise de conteúdos das seguintes fontes de informação: Abbas (2010), Aitchison, Bawden e Gilchrist (2000), Arp, Smith e Spear (2015), Baonza (2010), Denton (2009), Fernández, Gómez-Pérez e Juristo (1997), Glushko (2013), Ontology Engineering Group (2015), Harpring (2010), Hedden (2010), Kwasnik (1999), Barre (2010), NISO (2005), Noy e McGuinness (2001), Rowley e Hartley (2008), Staab et al. (2001), Sure, Staab e Studer (2004), Svenonius (2000) e Svenonius (2003). Tal como em IEEE (2006), as atividades foram organizadas em grupos. Para cada grupo, foi proposta uma lista de atividades. As atividades foram listadas na ordem que pareceu ser a mais lógica. A ordem escolhida não

Quadro 18 – Medidas

ELEMENTO	MEDIDA
Vocabulário controlado	Número de acessos. Número de atualizações.
Termo	Número de termos. Número de termos distintos. Número de termos que designam mais de um conceito. Número de termos com erros de ortografia. Número de termos com erros de sintaxe. Número de termos sem relação hierárquica. Número de termos com mais de um termo genérico direto. Número de termos listados em fontes de referência. Número de termos com significados definidos em fontes de referência.
Relação	Número de relações. Número de relações hierárquicas. Número de relações de associação (associativa). Número de relações entre vocabulários controlados. Número de ancestrais. Número de ancestrais diretos. Número médio de ancestrais. Número de descendentes. Número de descendentes diretos. Número médio de descendentes. Número de hierarquias. Número de hierarquias com herança múltipla. Profundidade média de hierarquia. Profundidade máxima de hierarquia. Largura média de hierarquia. Largura máxima de hierarquia. Número de círculos.
Conceito	Número de conceitos. Número de conceitos redundantes. Número de conceitos fora do escopo do vocabulário controlado. Número de proposições. Número de proposições falsas. Número de proposições em conflito. Número de proposições em linguagem natural.

Fonte: Elaborado pelo autor

presume e nem dita a estrutura organizacional onde as atividades serão executadas e nem dita como sequenciar as atividades. A necessidade de determinada atividade e a ordem na qual será executada podem variar entre métodos ou processos. Cada atividade pode ser decomposta em atividades mais específicas, pode integrar um ou mais processos e pode ser executada mais de uma vez em método ou processo. Cada método ou processo pode ser composto por atividades originadas em diferentes grupos. Os nomes dos grupos de atividades são os seguintes:

- (a) Grupo de atividades de requisitos;
- (b) Grupo de atividades de desenho (*design*);
- (c) Grupo de atividades de implementação;

- (d) Grupo de atividades de avaliação;
- (e) Grupo de atividades de desenvolvimento de documentação;
- (f) Grupo de atividades de importação;
- (g) Grupo de atividades de instalação;
- (h) Grupo de atividades de manutenção.

As descrições das atividades listadas nos grupos podem ser encontradas nesta tese e/ou em fontes referenciadas. Para facilitar a integração das atividades a métodos ou a processos, uma padronização pode ser adotada para descrição das atividades. Adotando [IEEE \(2006\)](#) como uma referência, cada descrição de atividade pode, por exemplo, ser composta por informação de entrada e fontes dessa informação; ações para transformar informação de entrada em informação de saída; informação de saída e destinos dessa informação.

10.8.1 Grupo de atividades de requisitos

O grupo de atividades de requisitos é composto por atividades responsáveis por definir requisitos de vocabulário controlado. Nesse contexto, definir requisito engloba especificar e analisar requisito. A seguir, são listadas atividades que integram esse grupo.

- (a) Definir domínio;
- (b) Definir escopo;
- (c) Definir contexto;
- (d) Definir propósito;
- (e) Definir questão de competência;
- (f) Definir nível de formalização;
- (g) Definir nível de aderência a norma;
- (h) Definir nível de aderência a guia de melhores práticas;
- (i) Definir idioma;
- (j) Definir requisito sobre fonte de informação;
- (k) Definir perfil de usuário;
- (l) Definir quantidade de usuários;

- (m) Analisar necessidade de informação de usuário;
- (n) Analisar comportamento informacional de usuário;
- (o) Definir garantia;
- (p) Definir característica de qualidade;
- (q) Definir evidência de qualidade;
- (r) Definir medida de qualidade;
- (s) Definir estrutura de vocabulário controlado.

10.8.2 Grupo de atividades de desenho (*design*)

O grupo de atividades de desenho (*design*) é composto por atividades responsáveis por desenhar (*design*) o vocabulário controlado a partir de requisitos definidos para o vocabulário controlado. A seguir, são listadas atividades que integram esse grupo.

- (a) Selecionar fonte de informação;
- (b) Analisar fonte de informação;
- (c) Selecionar termo;
- (d) Incluir termo em lista de termos;
- (e) Desenhar (*design*) termo;
- (f) Desenhar (*design*) faceta;
- (g) Desenhar (*design*) relação;
- (h) Desenhar (*design*) conceito;
- (i) Desenhar (*design*) armazenamento de vocabulário controlado;
- (j) Desenhar (*design*) apresentação de vocabulário controlado;
- (k) Alinhar conteúdo de vocabulário controlado a conteúdo de fonte de referência.

10.8.3 Grupo de atividades de implementação

O grupo de atividades de implementação é composto por atividades responsáveis por implementar vocabulário controlado a partir de desenho (*design*) e adequar vocabulário controlado a requisito de produto. A seguir, são listadas atividades que integram esse grupo.

- (a) Selecionar linguagem de implementação;
- (b) Implementar vocabulário controlado;
- (c) Adequar vocabulário controlado a requisito de produto.

10.8.4 Grupo de atividades de avaliação

O grupo de atividades de avaliação é composto por atividades responsáveis por identificar defeito em vocabulário controlado ou em processo integrante de ciclo de vida de projeto de vocabulário controlado. A seguir, são listadas atividades que integram esse grupo.

- (a) Desenvolver procedimento de teste;
- (b) Desenvolver caso de teste;
- (c) Testar vocabulário controlado;
- (d) Relatar resultado de teste de vocabulário controlado.

10.8.5 Grupo de atividades de desenvolvimento de documentação

O grupo de atividades de desenvolvimento de documentação é composto por atividades que são responsáveis por desenvolver, por produzir e por distribuir documentação sobre vocabulário controlado. A seguir, são listadas atividades que integram esse grupo.

- (a) Desenvolver documentação;
- (b) Produzir documentação;
- (c) Distribuir documentação.

10.8.6 Grupo de atividades de importação de vocabulário controlado

O grupo de atividades de importação é composto por atividades responsáveis por selecionar e importar vocabulário controlado. A seguir, são listadas atividades que integram esse grupo.

- (a) Selecionar fonte de vocabulário controlado;
- (b) Analisar vocabulário controlado;
- (c) Importar vocabulário controlado.

10.8.7 Grupo de atividades de instalação

O grupo de atividades de instalação é composto por atividades responsáveis por instalar vocabulário controlado em ambiente de operação. A seguir, são listadas atividades que integram esse grupo.

- (a) Distribuir vocabulário controlado;
- (b) Instalar vocabulário controlado em ambiente de operação;
- (c) Aceitar vocabulário controlado em ambiente de operação.

10.8.8 Grupo de atividades de manutenção

O grupo de atividades de manutenção é composto por atividades responsáveis por identificar necessidade de manutenção, relatar necessidade de manutenção e efetuar manutenção de vocabulário controlado. A seguir, são listadas atividades que integram esse grupo.

- (a) Identificar necessidade de manutenção;
- (b) Relatar necessidade de manutenção;
- (c) Efetuar manutenção.

11 Exemplo de uso de elementos propostos

Este capítulo descreve um exemplo de uso de elementos do arcabouço de arquitetura da informação para ciclo de vida de projeto de vocabulário controlado proposto nesta tese. Por meio desse exemplo, se pretende ilustrar o uso de elementos do arcabouço de arquitetura da informação anteriormente proposto. Este capítulo engloba informação sobre definição de processos, seleção de modelo de ciclo de vida de projeto, mapeamento de atividades, seleção de ferramentas, incorporação de ativos organizacionais e resultados de atividades executadas. O exemplo de uso de elementos do arcabouço, descrito neste capítulo, resultou em um protótipo de vocabulário controlado no domínio da Engenharia de Software. Ao final deste capítulo, são apresentadas sugestões para avaliação e desenvolvimento futuro do protótipo de vocabulário controlado.

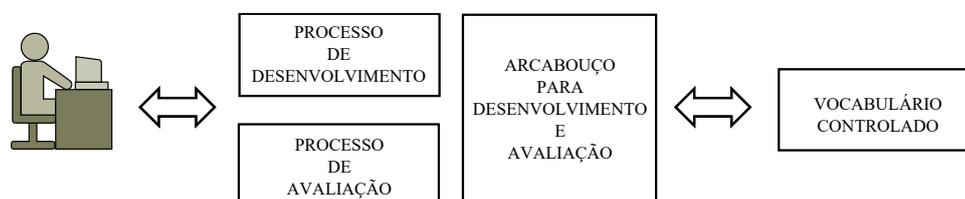
11.1 Contexto de aplicação e limites

Com o objetivo de delimitar o escopo da pesquisa, foram enfocadas as áreas de conhecimento Requisitos de Software, Desenho (*design*) de Software, Construção de Software e Teste de Software (nomes segundo [IEEE \(2014c\)](#)). Também com o objetivo de delimitar o escopo da pesquisa, parte da lista de termos candidatos foi alocada às facetas, parte dos termos do vocabulário controlado foi definida e atividades para uso de vocabulário controlado em metadados foram apenas listadas. O protótipo de vocabulário controlado foi estruturado como tesauro. Nesse contexto, o termo tesauro designa vocabulário controlado composto por termos, relações hierárquicas, relações de equivalência, relações de associação e descrições de conceitos. São justificativas para escolha dessa classe de vocabulário controlado: tesauro pode ser usado para melhorar a recuperação da informação e o entendimento de áreas de conhecimento ([AITCHISON; BAWDEN; GILCHRIST, 2000](#); [MORVILLE; ROSENFELD, 2006](#)); o desenvolvimento de vocabulário controlado estruturado como tesauro é uma abordagem para análise de domínio ([HJØRLAND, 2002](#)). A figura 41 representa elementos no exemplo de uso.

11.2 Definição de processos

A definição dos processos usados no ciclo de vida do projeto que resultou no protótipo de vocabulário controlado no domínio da Engenharia de Software ocorreu por meio das seguintes atividades: definir requisitos para processo em ciclo de vida de projeto, selecionar modelo de ciclo de vida de projeto, definir ciclo de vida de projeto por meio de mapeamento de atividades para modelo de ciclo de vida de projeto selecionado, incorporar

Figura 41 – Elementos no exemplo de uso do arcabouço

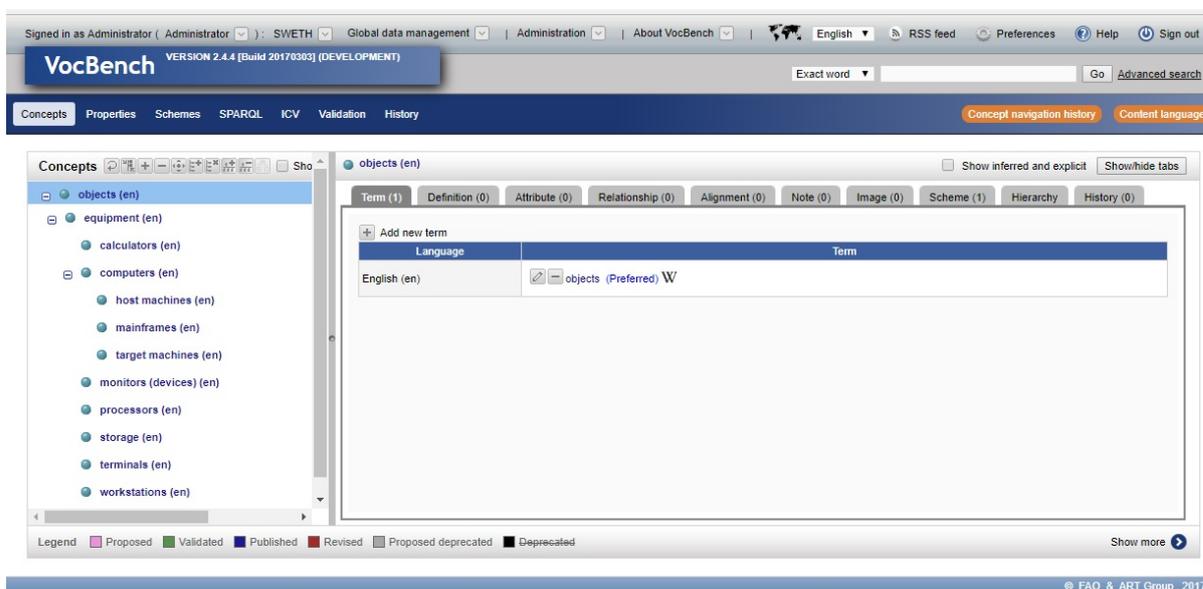


Fonte: Elaborado pelo autor

ativos organizacionais a ciclo de vida de projeto e validar processo em ciclo de vida de projeto. Os processos orientados a produto tiveram as seguintes responsabilidades: desenvolvimento de vocabulário controlado, implementação de vocabulário controlado e avaliação de vocabulário controlado. O modelo de ciclo de vida do projeto, foi definido a partir de modelo de ciclo de vida genérico descrito em [PMI \(2013\)](#). Cada fase foi decomposta em iterações. Com o objetivo de promover a melhoria da qualidade, algumas iterações foram estruturadas segundo o ciclo *Plan-Do-Check-Act* (PDCA) (descrito na seção [8.1.1](#)). Para os processos, foram mapeadas atividades da lista de atividades do arcabouço proposto nesta tese. Algumas atividades foram executadas mais de uma vez. Quando da definição dos processos, foram incorporados os modelos do arcabouço proposto nesta tese. Esses modelos foram usados em atividades nos referidos processos. Para suporte aos processos, foram selecionadas as seguintes ferramentas: *OpenOffice Calc*, *VocBench* e *SKOS Play*. As ferramentas *VocBench* e *SKOS Play* foram descritas na seção [5.7.4](#). A título de ilustração, a figura [42](#) é uma tela da ferramenta *VocBench* e a figura [43](#) apresenta a sequência de uso dessas ferramentas.

11.3 Resultados de atividades

As atividades mapeadas para os processos enfocaram definição de domínio, definição de idioma, definição de garantia, definição de necessidade de informação, seleção de fonte de informação, construção de lista de termos, definição de faceta, definição de relação, definição de conceito, desenvolvimento de procedimento de teste, desenvolvimento de caso de teste, teste de vocabulário controlado, relato de resultado de teste de vocabulário controlado e implementação de vocabulário controlado. A seguir, são descritos resultados das atividades.

Figura 42 – Tela da ferramenta *VocBench*

Fonte: Elaborado pelo autor

11.3.1 Definição de domínio

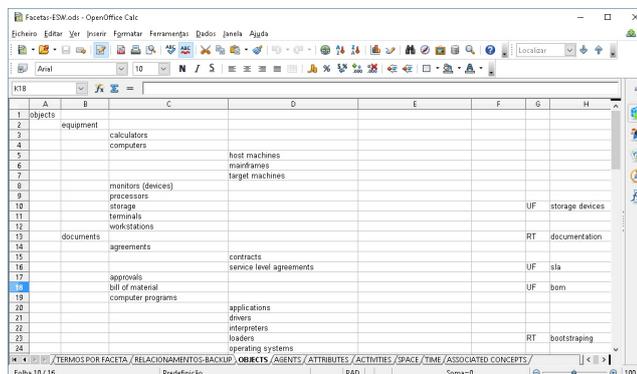
Na definição de domínio, foram observados eixos sugeridos em [Tennis \(2003\)](#). Esses eixos são denominados área de modulação e grau de especialização. O eixo área de modulação especifica nome e extensão do domínio. A extensão do domínio consiste no seu escopo. Esse eixo responde questões sobre como o domínio é chamado e o que abrange, designa o que é incluído e o que não é incluído. O eixo grau de especialização detalha como o domínio é organizado, qualifica e define a extensão do domínio. No projeto que resultou no exemplo de uso descrito neste capítulo, o domínio da disciplina Engenharia de Software é o domínio sobre o qual foi desenvolvido o vocabulário controlado. Nesse domínio, foram enfocadas as áreas de conhecimento denominadas Requisitos de Software, Desenho (*design*) de Software, Construção de Software e Teste de Software (nomes segundo [IEEE \(2014c\)](#)).

11.3.2 Definição de idioma e garantia semântica

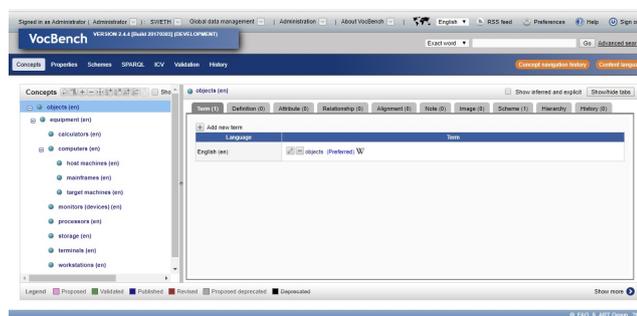
O inglês foi o idioma escolhido para o vocabulário controlado desenvolvido. Essa escolha decorreu da popularidade desse idioma entre engenheiros de software e o fato de diversos projetos de software serem realizados por equipes globalmente distribuídas. A garantia semântica escolhida para desenvolvimento do vocabulário controlado foi a garantia literária.

Figura 43 – Sequência de uso de ferramentas

Vocabulário representado na ferramenta OpenOffice Calc



Vocabulário representado na ferramenta VocBench



Vocabulário publicado pela ferramenta SKOS Play

- objects
- calculators
 - BT: equipment
 - TT: objects
 - computers
 - BT: equipment
 - TT: objects
 - NT: target machines
 - NT: host machines
 - NT: mainframes
 - equipment
 - BT: objects
 - TT: objects
 - NT: calculators
 - NT: monitors (devices)
 - NT: workstations
 - NT: computers
 - NT: terminals
 - NT: storage
 - NT: processors
 - host machines
 - BT: computers
 - monitors (devices)
 - BT: equipment
 - TT: objects
 - objects
 - NT: equipment
 - processors
 - BT: equipment
 - TT: objects
 - storage
 - UF: storage devices
 - BT: equipment
 - TT: objects
 - storage-devices
 - USE: storage
 - target machines
 - BT: computers
 - TT: objects
 - terminals
 - BT: equipment
 - TT: objects

Fonte: Elaborado pelo autor

11.3.3 Definição de necessidades de informação

Para definição de necessidades de informação, foram identificados potenciais interessados (*stakeholder*) nos resultados do projeto do vocabulário controlado. Foram enfocadas as necessidades de informação de engenheiro de software que, nesse contexto, é pessoa que aplica engenharia de software em ciclo de vida de software. Necessidades de informação de engenheiros de software foram abordadas na seção 9.2 desta tese. Foram enfocadas necessidades de informação de engenheiros de software relacionadas a desenho (*design*) de software.

11.3.4 Seleção de fontes de informação

Para escolha das fontes de informação, foram consideradas vantagens e desvantagens descritas em Kang et al. (1990). O quadro 19 apresenta vantagens e desvantagens das seguintes classes de fontes: livro texto, padrão, aplicação existente e especialista no domínio. Considerando a garantia semântica escolhida, necessidades de informação, vantagens e desvantagens em Kang et al. (1990), as fontes de informação selecionadas foram normas sobre padrões em Engenharia de Software e artefatos de projetos de software. As normas foram selecionadas de modo que o vocabulário controlado fosse embasado em especificações endossadas por organização renomada e tida como referência em Engenharia de Software. Por sua vez, os artefatos resultaram de projetos de software relevantes. Como fonte de normas, foi selecionada a organização IEEE, uma importante organização na área de tecnologia (IEEE, 2017). Levando-se em consideração as necessidades de informação identificadas, foram selecionadas normas publicadas pelo IEEE relevantes às seguintes áreas de conhecimento: Requisitos de Software, Desenho (*design*) de Software, Construção de Software e Teste de Software. A identificação das normas relevantes nessas áreas ocorreu por meio de análise de IEEE (2014c). Como fonte de informação, também foi usada a norma ISO/IEC/IEEE 24765 - *Systems and Software Engineering Vocabulary*, identificada na pesquisa bibliográfica. O quadro 20 contém os identificadores das normas selecionadas para análise.

Quadro 19 – Fontes de informação na análise de domínio

FONTES	VANTAGENS	DESVANTAGENS
Livros texto	Consiste de boa fonte de conhecimento no domínio, teorias, métodos, técnicas e modelos.	Reflete apenas visões específicas dos autores. Pode usar modelos idealizados ou tendenciosos.
Padrões	Representa modelo de referência padrão para o domínio.	Modelo pode não estar atualizado com nova tecnologia.
Aplicações existentes	Mais importante fonte de conhecimento acerca do domínio. Pode ser usado diretamente de modo a determinar características visíveis ao usuário. Documentos de requisitos disponíveis para modelo de domínio. Projeto detalhado e código fonte revelam arquiteturas.	Custo de analisar muitos sistemas é elevado.
Especialistas no domínio	Pode prover informação de contexto e razão não disponíveis em outras fontes. Pode ser consultado durante análise de domínio e posteriormente ser validador de produtos.	Especialistas têm diferentes áreas de especialidade; diversos especialistas podem ser necessários.

Fonte: Adaptado de Kang et al. (1990)

Quadro 20 – Normas selecionadas

ÁREAS DE CONHECIMENTO	NORMAS SELECIONADAS
Requisitos de Software	IEEE 29148:2011
Desenho (<i>design</i>) de Software	IEEE Std. 26514-2010 IEEE 42010:2011 IEEE Std. 1016-2009
Construção de Software	ISO/IEC TR 24772:2013
Teste de Software	IEEE Std. 26513-2010 IEEE Std. 829-2008 IEEE Std. 1008-1987 IEEE 29119-1 IEEE 29119-2 IEEE 29119-3 IEEE 29119-4

Fonte: Elaborado pelo autor

11.3.5 Construção de lista de termos

Uma vez selecionadas, as fontes de informação foram analisadas para se construir uma lista de termos candidatos. Na construção dessa lista de termos, inicialmente foram coletados os termos relacionados na norma ISO/IEC/IEEE 24765. Essa norma relaciona termos originados, principalmente, em publicações das seguintes organizações: ISO, IEEE e PMI (ISO, 2010). A partir da norma ISO/IEC/IEEE 24765, foi construída uma lista com 3.327 termos. Essa lista de termos foi complementada com termos para os quais foram encontradas definições em outras normas. Esse processo resultou em uma lista com 3.877 termos. Em seguida, artefatos de projetos de software foram analisados com o objetivo de identificar quais dos termos na lista de termos candidatos eram usados nesses artefatos. Esse processo teve o objetivo de identificar quais termos estavam presentes tanto em normas analisadas, quanto em artefatos resultantes de projetos de software. Para alcançar esse objetivo, inicialmente foram definidas classes de projetos de software. Cada classe foi caracterizada pela classe do processo de desenvolvimento de software adotado. As classes de processos de desenvolvimento de software consideradas foram as seguintes: Processo Unificado de Desenvolvimento de Software, Processo distribuído de desenvolvimento de software e Processo ágil de desenvolvimento de software. A classe escolhida foi Processo distribuído de desenvolvimento de software, devido a diversos projetos de software serem globalmente distribuídos. Em seguida, foram escolhidos projetos que adotam processo distribuído de desenvolvimento de software aberto (*open source*). Essa escolha decorreu da importância dessa classe de projeto de software e da possibilidade de acesso a artefatos de projetos dessa classe.

Após escolhida a classe de projetos de software, foram selecionados projetos de software. Para isso, foi analisado conteúdo do sítio denominado *Black Duck Open Hub* (vide [Open Hub \(2017\)](#)). Esse sítio disponibiliza dados sobre projetos de software aberto, por

exemplo: número de linhas de código (*lines of code*), linguagens de programação usadas, avaliações (*ratings*) de usuários etc. Por meio de análise de dados disponíveis em [Open Hub \(2017\)](#), foram selecionados os seguintes projetos de software aberto: *Apache HTTP Server*, *Apache Subversion*, *PHP* e *OpenStack*. Esses projetos de software aberto foram selecionados considerando-se popularidade, tamanho e atividade. O quadro 21 contém alguns dados, obtidos a partir de [Open Hub \(2017\)](#) em 13 de julho de 2017, sobre esses projetos de software aberto. Uma vez selecionados os projetos de software, artefatos das seguintes classes foram transferidos a partir de sítios desses projetos: artigo, glossário, guia, lista de perguntas frequentes, livro, manual, nota e tutorial. Uma vez transferidos os artefatos, a ferramenta *AntConc*, uma ferramenta para análise de concordância e texto, foi usada para analisar os artefatos ([ANTHONY, 2014](#)). Essa análise foi realizada com o objetivo de identificar quais termos na lista de termos candidatos se encontravam nesses artefatos. Esse processo resultou em uma lista composta por 1.016 termos candidatos. Nesse processo, não foram analisados significados dos termos, apenas a presença deles. A título de ilustração, o quadro 22 contém os cinquenta primeiros termos na lista construída.

Quadro 21 – Dados sobre projetos selecionados

PROJETOS	LINHAS DE CÓDIGO	COMMITTS	CONTRIBUIDORES	HOME PAGES
Apache HTTP Server	1.883.545	140.043	122	http://httpd.apache.org/
Apache Subversion	660.711	12.491	85	http://subversion.apache.org/
PHP	3.886.491	230.466	1.022	http://php.net/
OpenStack	3.837.759	394.498	6.712	http://www.openstack.org/

Fonte: Elaborado pelo autor

Quadro 22 – Trecho de lista de termos

abort	accessibility	address space	alphanumeric	application frameworks
abstract class	accident	ae	analysis	approval
abstraction	accuracy	affordable	ancestor	arc
acceptable	action	agent	annotate	architect
acceptance criteria	activation	agreement	annotation	architectural design
acceptance testing	activity	algorithm	announcement	architecture
access	actor	alias	ansi	architecture model
access method	actual argument	alignment	ap	argument
access object	adapter	allocation	api	array
access type	address	allocator	application	arrow

Fonte: Elaborado pelo autor

11.3.6 Definição de facetas

No processo de desenvolvimento do tesauro, a partir da lista de termos, foi inicialmente desenvolvido um esquema de classificação. A Teoria das Facetas foi adotada no processo de desenvolvimento desse esquema de classificação. Entre os motivos da escolha dessa teoria, é possível destacar vantagens dela na classificação da informação em domínios de engenharia (ALBRECHTSEN, 1992; GIESS; WILD; MCMAHON, 2008; PRIETO-DIAZ; FREEMAN, 1987). No desenvolvimento do esquema de classificação, foram observados princípios descritos em Spiteri (1998). As facetas do esquema de classificação foram definidas por meio de análise de fontes referenciadas na seção 3.7.2.2, em particular, por meio de análise de facetas propostas pelo CRG. Essa escolha decorreu dessas facetas serem consideradas adequadas a assuntos nos domínios da tecnologia e manufatura (BROUGHTON, 2006a).

Os nomes das facetas propostas são os seguintes: *Objects*, *Agents*, *Attributes*, *Activities*, *Space* e *Time*. Os nomes estão em inglês, por ser esse o idioma do vocabulário controlado. A faceta *Objects* engloba entidades que são tangíveis e visíveis. A faceta *Agents* engloba entidades que executam atividades. A faceta *Attributes* engloba entidades que designam características. A faceta *Activities* engloba entidades que designam atividades, seqüências de atividades, ocorrências discretas e métodos. A faceta *Space* engloba entidades que designam pontos ou extensões no espaço. Finalmente, a faceta *Time* engloba entidades que designam períodos de tempo.

11.3.7 Atribuição de termos

Uma vez definidas as facetas, termos da lista de termos candidatos foram atribuídos às facetas. Considerando o tamanho da lista de termos candidatos e com o objetivo de delimitar essa fase da pesquisa, após análise da lista de 1.016 termos candidatos foram atribuídos 375 termos às facetas. É importante destacar que as facetas propostas não necessariamente abrangem todos os conceitos designados na lista de termos candidatos. Podem existir termos nessa lista que não são atribuíveis às facetas propostas. No processo de atribuição de termos às facetas, foram consultadas fontes de informação sobre Engenharia de Software e domínios correlatos, por exemplo, ACM (2017), IEEE (2014a) e IEEE (2016). Além dessas fontes de informação, também foram consultados Princeton University (2010), Dictionary.com (2017), European Union (2017), The Getty Research Institute (2015) e FAO (2015a).

11.3.8 Definição de relações

Após a atribuição dos termos às facetas, foram definidas relações entre termos. Além das fontes de informação consultadas na atribuição de termos às facetas, foram também

consultadas fontes de informação sobre vocabulários controlados no domínio da Engenharia de Software. Essas fontes de informação foram identificadas por processo de mapeamento sistemático de literatura realizado pelo autor desta tese, seguindo processo similar ao descrito em [Petersen et al. \(2008\)](#). Resultado desse mapeamento sistemático de literatura é encontrado no apêndice A desta tese. Os vocabulários controlados identificados foram classificados considerando-se estrutura e área de conhecimento da Engenharia de Software. Quanto à estrutura, cada vocabulário controlado foi classificado de acordo com a definição dada, na fonte de informação, pelo seu autor. Alguns vocabulários foram classificados como taxonomias, outros como ontologias. A classificação de cada vocabulário controlado em área de conhecimento foi realizada por meio de análise de fonte de informação sobre o vocabulário controlado. As áreas de conhecimento da Engenharia de Software consideradas foram as descritas em [IEEE \(2014c\)](#). Para cada área de conhecimento, a tabela 1 contém o percentual de taxonomias localizadas, e a tabela 2 contém o percentual de ontologias localizadas.

Tabela 1 – Percentual de taxonomias por área

ÁREA DE CONHECIMENTO	PERCENTAGEM
Diversas	19.5%
Requisitos de Software	9.8%
Design de Software	12.2%
Construção de Software	12.2%
Manutenção de Software	14.6%
Gestão de Engenharia de Software	7.3%
Processo de Engenharia de Software	4.9%
Qualidade de Software	19.5%

Fonte: Elaborado pelo autor

Tabela 2 – Percentual de ontologias por área

ÁREA DE CONHECIMENTO	PERCENTAGEM
Diversas	17.0%
Requisitos de Software	4.3%
Design de Software	17.0%
Construção de Software	8.5%
Teste de Software	4.3%
Manutenção de Software	2.1%
Gestão de Engenharia de Software	10.6%
Processo de Engenharia de Software	27.7%
Modelos e Métodos de Engenharia de Software	2.1%
Qualidade de Software	6.4%

Fonte: Elaborado pelo autor

Considerando o número de fontes de informação sobre vocabulários controlados no domínio da Engenharia de Software identificadas no processo de mapeamento sistemático de literatura realizado pelo autor desta tese, apenas parte dessas fontes foi considerada na definição de relações. É importante destacar que ISO (2010) e IEEE (2016) contêm informação sobre relações entre termos em Engenharia de Software. Por exemplo, sobre o termo *software*, IEEE (2016) informa a existência das seguintes relações: *Syn: SW* e *See Also: application software*. As fontes ISO (2010) e IEEE (2016) foram analisadas quando da definição das relações entre termos nas facetas.

A título de ilustração, os quadros 23, 24, 25, 26, 27 e 28 contêm termos e relações em trechos iniciais de cada faceta do protótipo de vocabulário controlado desenvolvido. Em cada quadro, níveis hierárquicos são identificados por colunas. Termos no primeiro nível hierárquico (nível hierárquico mais alto) estão na primeira coluna, termos no segundo nível hierárquico estão na segunda coluna, e assim por diante. Por exemplo, no quadro 23, *objects* está no primeiro nível hierárquico, *equipment* está no segundo nível hierárquico e *calculators* está no terceiro nível hierárquico. Nesses quadros, há também relações designadas pelos termos UF e RT. O termo UF designa *use for*, enquanto RT designa *related term*. Por exemplo, no quadro 23, entre *storage* e *storage devices*, existe relação designada pelo termo UF.

Quadro 23 – Elementos iniciais em *Objects*

objects					
	equipment				
		calculators			
		computers			
			host machines		
			mainframes		
			target machines		
		monitors (equipment)			
		processors			
		storage		UF	storage devices

Fonte: Elaborado pelo autor

11.3.9 Definição de conceitos

Conceitos designados por termos do protótipo de vocabulário controlado desenvolvido foram definidos seguindo-se diretriz do arcabouço proposto nesta tese. Portanto, cada definição é composta por uma ou mais proposições. Um conceito é definido pelo conjunto de proposições. Os conceitos foram definidos em linguagem natural (em inglês por ser esse o idioma do protótipo de vocabulário controlado). No processo de definição dos conceitos, foram consultadas fontes de informação sobre Engenharia de Software e

Quadro 24 – Elementos iniciais em *Agents*

agents			
	people		
		architects	
		assessors	
		clients	
		consumers	
		customers	
		developers	
			committers
		evaluators	

Fonte: Elaborado pelo autor

Quadro 25 – Elementos iniciais em *Attributes*

attributes	
	accessibility
	accuracy
	authority
	availability
	compatibility
	complexity
	concurrency
	conformance
	connectivity

Fonte: Elaborado pelo autor

domínios correlatos, por exemplo, [ACM \(2017\)](#), [IEEE \(2014a\)](#), [IEEE \(2016\)](#) e [IEEE \(2016\)](#). Além dessas fontes de informação, também foi consultada a base de dados léxica [Princeton University \(2010\)](#) e os tesouros [Dictionary.com \(2017\)](#), [European Union \(2017\)](#), [The Getty Research Institute \(2015\)](#) e [FAO \(2015a\)](#). Em cada definição, as fontes que embasam a definição são referenciadas. A título de exemplo, a seguir são apresentadas as definições dos termos *compiler* e *terminal*.

- (a) **Compiler:** *is a computer program that translates programs expressed in a high-order language into their machine language equivalents (ISO, 2010).*
- (b) **Terminal:** *is an electronic equipment that provides access to a computer, and is an electronic equipment that has a keyboard, and is an electronic equipment that has a display (PRINCETON UNIVERSITY, 2010).*

Quadro 26 – Elementos iniciais em *Activities*

activities						
	aborting		RT	exception		
	addressing					
	assembling		RT	disassembling	compiling	interpreting
			UF	assemble		
	assessing		UF	assessment		
		estimating	UF	estimate		
	assigning		RT	binding		
			UF	assignment		
	auditing					

Fonte: Elaborado pelo autor

Quadro 27 – Elementos iniciais em *Space*

locations					
	areas				
		storage areas			
			buffers (storage areas)		
	environments				
		background		RT	foreground
		contexts			
		domains			
			security domains		
		foreground		RT	background

Fonte: Elaborado pelo autor

Quadro 28 – Elementos iniciais em *Time*

time periods					
	code freeze				
	commit window				
	cycle				
		life cycle	UF	lifecycle	life-cycle
		software development cycle			
	dates				
		early finish date	UF	ef	
		early start date	UF	es	
		late finish date	UF	lf	

Fonte: Elaborado pelo autor

11.3.10 Controle de qualidade

O controle de qualidade do vocabulário controlado desenvolvido englobou atividades com os seguintes objetivos: desenvolvimento de listas de verificação (*checklist*), teste de vocabulário controlado e relato de resultado de teste de vocabulário controlado. No controle de qualidade, ocorreu avaliação de evidências de qualidade anteriormente identificadas nesta tese sobre a característica de qualidade “compatibilidade”, em especial, compatibilidade de desenho (*design*) do vocabulário controlado. A qualidade foi avaliada por meio de listas de verificação (*checklist*) embasadas nas evidências de qualidade anteriormente mencionadas. A título de ilustração, a figura 44 apresenta trecho de uma lista de verificação usada na avaliação da qualidade do vocabulário. Esse trecho consiste de parte de uma das listas de verificação.

Figura 44 – Trecho de uma lista de verificação

LISTA DE VERIFICAÇÃO DE TERMOS

REVISOR : Fernando Albuquerque

DATA : 28 / 07 / 2017

PROJETO : Vocabulário controlado em Engenharia de Software

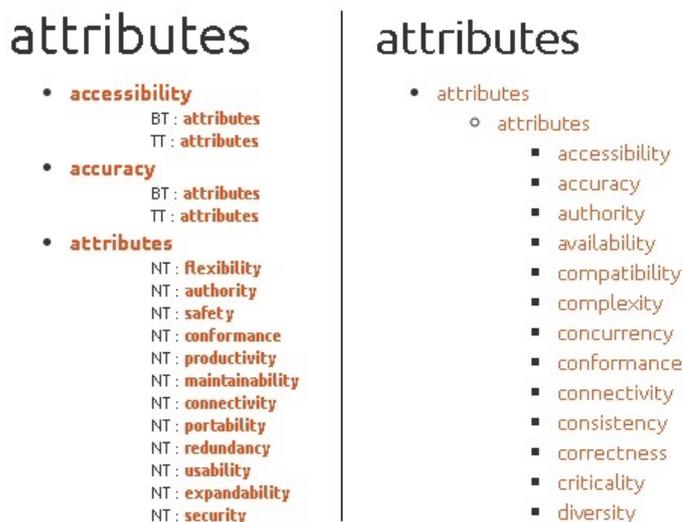
Termo designa apenas um conceito.	[x]
Termo de acordo com grafia amplamente aceita.	[x]
Termo de acordo com grafia em fonte de referência (dicionário, enciclopédia, glossário, etc.).	[x]
Termo de acordo com grafia adotada no vocabulário controlado.	[x]
Termo de acordo com sintaxe aceita.	[x]
Termo de acordo com convenção de nome padronizada.	[x]
Termo originado em literatura no domínio.	[x]
Termo é encontrado em outro vocabulário controlado.	[x]
Termo de acordo com a audiência.	[x]
Termo de acordo com o escopo do vocabulário controlado.	[x]
Termo de acordo com padrão (acadêmico, cultural, político, comercial, etc.).	[x]
Termo de acordo com estilo de escrita adotado no vocabulário controlado.	[x]
Termo é composto por poucas palavras.	[x]

Fonte: Elaborado pelo autor

11.3.11 Publicação do vocabulário controlado

Para publicação do protótipo de vocabulário controlado, foi usada a ferramenta *SKOS Play*. Essa ferramenta é descrita na seção 2.4. O vocabulário controlado foi publicado em formatos PDF e HTML, a partir de código SKOS gerado pela ferramenta *VocBench*. Para ilustrar o exposto, a figura 45 contém trechos de uma das páginas do protótipo de vocabulário controlado.

Figura 45 – Trechos de uma das páginas do protótipo de vocabulário controlado



Fonte: Elaborado pelo autor

11.4 Sugestões para avaliação e desenvolvimento futuro

Com o intuito de delimitar o escopo da pesquisa realizada, o protótipo de vocabulário controlado desenvolvido não foi avaliado em uso. Considerando a importância de metadados no contexto da Arquitetura da Informação, e que um importante uso de vocabulários controlados é no desenvolvimento de metadados, a seguir são listadas atividades que podem integrar um processo de uso do protótipo de vocabulário controlado desenvolvido. As atividades estão listadas na ordem que parece ser a mais lógica. Entretanto, vale destacar que essa ordem não dita como sequenciar as atividades. A necessidade de determinada atividade e a ordem na qual será executada podem variar. Cada atividade pode ser decomposta em atividades mais específicas, e pode ser executada mais de uma vez. As atividades foram identificadas por meio de análise, principalmente, das seguintes fontes de informação: [Taylor e Joudrey \(2009\)](#), [Miller \(2011\)](#), [CIESIN \(1998\)](#), [ISO \(2009a\)](#) e [ISO \(2009b\)](#).

- (a) Definir coleção de recursos de informação;
- (b) Definir foco da coleção de recursos de informação;
- (c) Definir classe de recurso de informação;
- (d) Definir quantidade de recursos de informação;
- (e) Analisar recurso de informação;
- (f) Analisar necessidade de informação de usuário;

- (g) Analisar comportamento informacional de usuário;
- (h) Analisar metadado existente para o recurso de informação;
- (i) Definir elemento de metadado;
- (j) Estruturar elemento de metadado;
- (k) Definir restrição associada a elemento de metadado;
- (l) Definir guia para conteúdo de elemento de metadado;
- (m) Definir padrão de codificação para elemento de metadado;
- (n) Selecionar esquema de metadados;
- (o) Mapear elemento de metadado para esquema de metadados;
- (p) Definir procedimento para uso de esquema de metadados;
- (q) Implementar esquema de metadados;
- (r) Selecionar vocabulário controlado;
- (s) Representar recurso de informação;
- (t) Definir critério para avaliação da representação da informação;
- (u) Avaliar representação da informação.

Por fim, aos interessados em prosseguir no desenvolvimento do protótipo de vocabulário controlado, é sugerida a adoção de um processo de desenvolvimento que enfatize a colaboração na distribuição de trabalho e no alcance de consenso. Nesse contexto, é útil a adoção de ferramentas que facilitem um processo de desenvolvimento colaborativo onde os participantes possam estar distribuídos geograficamente e/ou temporalmente. Por exemplo, a ferramenta *VocBench* provê suporte a um fluxo de trabalho editorial (*editorial workflow*) onde as contribuições sugeridas pelos desenvolvedores podem ser armazenadas, monitoradas e avaliadas. Nesse fluxo de trabalho editorial, os elementos criados e editados pelos desenvolvedores passam pelos seguintes diferentes estados: proposto, revisado, publicado, proposto descontinuado (*proposed deprecated*) e descontinuado (*deprecated*). Os participantes do processo de desenvolvimento podem assumir diferentes responsabilidades (STELLATO et al., 2015). As responsabilidades são caracterizadas pelo papel assumido. Por exemplo: editor de termo, editor de ontologia, validador, editor (*publisher*) e administrador. Por meio da ferramenta, os desenvolvedores podem, por exemplo: acrescentar, editar e excluir termos; acrescentar, editar e excluir relações entre termos (CARACCILO et al., 2015).

Parte IV

Considerações finais

Prólogo

Esta parte da tese contém apenas um capítulo. O capítulo 12, onde são apresentadas considerações finais sobre a pesquisa que resultou nesta tese, contém informação sobre alcance dos objetivos de pesquisa, potenciais contribuições resultantes da pesquisa, limitações da pesquisa e sugestões de possíveis pesquisas futuras.

12 Considerações finais

Nesta tese foram apresentados resultados de pesquisa bibliográfica na qual foram acessadas diversas fontes de informação, foi proposto arcabouço de arquitetura da informação para ciclo de vida de projeto de vocabulário controlado e foi descrito exemplo uso desse arcabouço no desenvolvimento de um protótipo de vocabulário controlado no domínio da disciplina Engenharia de Software. O arcabouço proposto é composto por arquitetura de referência, modelo de domínio, modelo de qualidade e lista de atividades. Esse arcabouço resultou de processo em que foram analisados resultados da pesquisa bibliográfica realizada. O exemplo de uso ocorreu por meio de projeto onde foi desenvolvido e avaliado um protótipo de vocabulário controlado no domínio da disciplina Engenharia de Software.

12.1 Alcance dos objetivos

É possível afirmar que os objetivos da pesquisa que resultou nesta tese foram alcançados. O objetivo que consistia na identificação e descrição de conceitos, métodos, atividades, processos, teorias, normas, guias de melhores práticas, ferramentas e modelos de qualidade sobre vocabulário controlado foi alcançado nos capítulos 3 a 9. O objetivo que consistia da proposição de elementos de um arcabouço conceitual para ciclo de vida de projeto de vocabulário controlado foi alcançado no capítulo 10. O objetivo que consistia na escolha de domínio de aplicação onde ocorresse exemplo de uso do arcabouço proposto e descrição de conceitos relevantes ao entendimento desse exemplo foi alcançado nos capítulos 9 e 11. Por fim, o objetivo que consistia no desenvolvimento de um exemplo de uso de elementos do arcabouço conceitual proposto por meio da construção de um protótipo de vocabulário controlado sobre o domínio de aplicação escolhido foi alcançado no capítulo 11.

12.2 Contribuições da pesquisa

As potenciais contribuições desta pesquisa são: (a) identificação de temas de pesquisa no domínio da Arquitetura da Informação; (b) organização de conceitos dispersos em diversas fontes sobre arquitetura, recuperação, organização e representação da informação; (c) normalização de termos usados para designar conceitos sobre arquitetura, recuperação, organização e representação da informação; (d) elaboração de texto que pode ser usado na educação e pesquisa em Ciência da Informação e em outras disciplinas; (e) definições de conceitos embasadas em normas e em ferramentas semânticas; (f) proposta de arcabouço composto por elementos que podem ser usados em processo de desenvolvimento e em

processo de avaliação de vocabulário controlado; (g) descrição de processo de desenvolvimento de modelos embasados em ferramentas semânticas; (h) descrição de processo de desenvolvimento de vocabulário controlado embasado no uso de ferramentas semânticas; (i) protótipo de vocabulário controlado no domínio da Engenharia de Software; (j) relação de vocabulários controlados no domínio da Engenharia de Software; (k) lista de referências que pode ser usada em educação e em pesquisa sobre arquitetura, recuperação, organização e representação da informação.

12.3 Limitações da pesquisa

Entre as limitações presentes na pesquisa que resultou nesta tese é possível destacar: (a) listas incompletas de pontos de vista, de características de qualidade, de evidências de qualidade, de medidas de qualidade e de atividades; (b) não foram propostas evidências de qualidade para todas as características de qualidade; (c) não foram estabelecidas todas as relações entre evidências e características de qualidade propostas; (d) não foram estabelecidas relações entre medidas e evidências de qualidade; (e) o arcabouço proposto não foi usado em mais de um domínio; (f) o arcabouço proposto não foi usado em projeto realizado por equipe com diversos integrantes; (g) no exemplo de uso do arcabouço proposto, não foram executadas todas as atividades na lista de atividades desse arcabouço; (h) o protótipo de vocabulário controlado desenvolvido foi avaliado apenas pelo seu autor; (i) não foram analisados, classificados e definidos todos os termos integrantes da lista de termos candidatos construída; (j) não foi avaliado impacto decorrente do uso do protótipo de vocabulário controlado desenvolvido na recuperação da informação contida em repositório de software.

12.4 Sugestões para pesquisas futuras

Como sugestões de pesquisas futuras relacionadas a esta tese: (a) ampliar o arcabouço proposto para ciclo de vida de projeto de vocabulário controlado, por exemplo, por meio da inclusão de outras características de qualidade, evidências de qualidade, medidas de qualidade e atividades relevantes a diversos processos em ciclo de vida de projeto de vocabulário controlado; (b) testar o arcabouço proposto por meio do desenvolvimento e avaliação de vocabulários controlados em diversos domínios; (c) testar o arcabouço proposto em projeto realizado por equipe com diversos integrantes; (d) descrever as atividades listadas em formato que facilite a integração delas a processos em ciclo de vida de projeto de vocabulário controlado; (e) construir vocabulário controlado no domínio da Engenharia de Software a partir do protótipo de vocabulário controlado desenvolvido; (f) avaliar o impacto decorrente do uso de vocabulário controlado construído a partir do protótipo desenvolvido, na recuperação da informação contida em repositório de software;

(g) avaliar o uso de vocabulário controlado desenvolvido a partir do protótipo desenvolvido, na educação em Engenharia de Software; (h) avaliar a importância do uso de vocabulários controlados no desenvolvimento de modelos de software.

Referências

- ABBAS, J. *Structures for Organizing Knowledge: Exploring taxonomies, ontologies and other schemas*. 1st. ed. New York: Neal-Schuman Publishers, 2010. 249 p. Citado 20 vezes nas páginas 23, 24, 73, 81, 82, 85, 89, 90, 101, 102, 104, 117, 119, 129, 133, 187, 259, 261, 262 e 269.
- ABNT. *NBR ISO/IEC 9126-1 Engenharia de software - Qualidade de produto Parte 1: Modelo de qualidade*. 2003. Citado 3 vezes nas páginas 170, 180 e 261.
- ABNT. *Normalização*. 2014. Acessado em: 11 de março de 2016. Disponível em: <<http://www.abnt.org.br/normalizacao/o-que-e/o-que-e>>. Citado 2 vezes nas páginas 47 e 265.
- ACM. *2012 ACM Computing Classification System*. 2017. Acessado em 29 de abril de 2017. Disponível em: <<https://www.acm.org/publications/class-2012>>. Citado 2 vezes nas páginas 200 e 203.
- AGUIAR, A. C. Informação e atividades de desenvolvimento científico, tecnológico e industrial: tipologia proposta com base em análise funcional. *Ciência da Informação*, v. 20, n. 1, p. 7–15, 1991. Citado 2 vezes nas páginas 46 e 263.
- AHMED, E. Use of ontologies in software engineering. In: *Proceedings 17th International Conference on Software Engineering and Data Engineering, SEDE 2008*. Los Angeles, California, USA: [s.n.], 2008. p. 145–150. Citado na página 168.
- AITCHISON, J. The thesaurifacet: A multipurpose retrieval language tool. *Journal of Documentation*, v. 26, n. 3, September 1970. Citado na página 104.
- AITCHISON, J.; BAWDEN, D.; GILCHRIST, A. *Thesaurus Construction and Use: A Practical Manual*. 4th. ed. Chicago: Fitzroy Dearborn Publishers, 2000. 230 p. Citado 20 vezes nas páginas 56, 64, 67, 69, 70, 71, 74, 82, 83, 85, 86, 93, 94, 103, 104, 140, 184, 187, 193 e 266.
- AKERMAN, A.; TYREE, J. Using ontology to support development of software architectures. *IBM Systems Journal*, v. 45, n. 4, p. 813–825, 2006. Citado na página 277.
- AL-BADAREEN, A. et al. Software quality models: A comparative study. In: *Communications in Computer and Information Science*. [S.l.: s.n.], 2011. Citado na página 147.
- AL-SHORAFAT, W. Security in software engineering requirement. In: *2013 8th International Conference for Internet Technology and Secured Transactions, ICITST 2013*. Yangzhou, Jiangsu, China: [s.n.], 2013. p. 666–673. Citado na página 274.
- ALBRECHTSEN, H. *Domain Analysis for Classification of Software*. Dissertação (Mestrado) — The Royal School of Librarianship, Copenhagen, 1992. Citado 8 vezes nas páginas 66, 67, 71, 134, 167, 200, 259 e 261.

- ALBUQUERQUE, A. *Discurso sobre fundamentos de Arquitetura da Informação*. Tese (Doutorado) — Faculdade de Ciência da Informação - Universidade de Brasília, 2010. Disponível em: <<http://repositorio.unb.br/handle/10482/7110>>. Citado na página 76.
- ALBUQUERQUE, A.; LIMA-MARQUES, M. Sobre os fundamentos da arquitetura da informação. *Perspectivas em Gestão do Conhecimento*, v. 1, n. Número Especial, p. 60–72, Outubro 2011. Citado 7 vezes nas páginas 24, 28, 76, 78, 79, 175 e 177.
- ALBUQUERQUE, F. *TCP/IP Internet Protocolos & Tecnologias*. 3º edição. ed. [S.l.]: Axcel Books, 2001. Citado 2 vezes nas páginas 263 e 267.
- ALBUQUERQUE, F. *Modelo de Framework de Arquitetura da Informação Baseado em Roteiros*. Dissertação (Mestrado) — Faculdade de Ciência da Informação, Universidade de Brasília, 2014. Disponível em: <<http://repositorio.unb.br/handle/10482/16070>>. Citado 3 vezes nas páginas 35, 36 e 132.
- ALEMAN-MEZA, B. et al. Ontoqa: Metric-based ontology quality analysis. In: *IEEE ICDM 2005 Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*. Houston, TX: [s.n.], 2005. Citado 2 vezes nas páginas 182 e 184.
- ALI, N.; HOSKING, J.; GRUNDY, J. A taxonomy and mapping of computer-based critiquing tools. *IEEE Transactions on Software Engineering*, v. 39, n. 11, p. 1494–1520, Nov 2013. ISSN 0098-5589. Citado na página 275.
- ALLARD, S.; LEVINE, K.; TENOPIR, C. Design engineers and technical professionals at work: Observing information usage in the workplace. *Journal of the American Society for Information Science and Te*, v. 60, n. 3, p. 443–454, March 2009. Citado na página 155.
- ALLEMANG, D.; HENDLER, J. *Semantic Web for the Working Ontologist*. 2nd. ed. [S.l.]: Morgan Kaufmann, 2007. 384 p. Citado na página 116.
- ALLEN, T. J. *Managing the flow of technology: technology transfer and the dissemination of technological information within the R & D organization*. Cambridge, Massachusetts: MIT Press, 1984. 320 p. Citado na página 47.
- AMBLER, S. *Enterprise Unified Process (EUP): Strategies for Enterprise Agile*. 2013. Acessado em: 20 de agosto de 2014. Disponível em: <<http://enterpriseunifiedprocess.com/>>. Citado na página 160.
- AMBROSIO, A. et al. Software engineering documentation: an ontology-based approach. In: *Proceedings WebMedia and LA-Web, 2004*. [S.l.]: IEEE, 2004. p. 38–40. Citado na página 168.
- AMERI, F.; URBANOVSKY, C.; MCARTHUR, C. A systematic approach to developing ontologies for manufacturing service modeling. In: *7th International Conference on Formal Ontology in Information Systems (FOIS 2012)*. Graz, Austria: [s.n.], 2012. Citado na página 117.
- ANKOLEKAR, A. Supporting online problem-solving communities with the semantic web. In: *Proceedings of the 15th International Conference on World Wide Web*. New York, NY, USA: ACM, 2004. p. 575–584. Citado na página 162.

- ANKOLEKAR, A.; HERBSLEB, J.; SYCARA, K. Addressing challenges to open source collaboration with the semantic web. In: *Taking Stock of the Bazaar: The 3rd Workshop on Open Source Software Engineering, the 25th International Conference on Software Engineering (ICSE)*. Portland, OR, USA: [s.n.], 2003. p. 9–13. Citado 2 vezes nas páginas 33 e 162.
- ANQUETIL, N.; OLIVEIRA, K.; DIAS, M. Ontologies for software engineering and software technology. In: _____. New York, USA: Springer, 2006. cap. Software Maintenance Ontology, p. 153–173. Citado na página 278.
- ANTHONY, L. *AntConc Homepage*. 2014. Acessado em 23 de fevereiro de 2017. Disponível em: <<http://www.laurenceanthony.net/software/antconc/>>. Citado 2 vezes nas páginas 41 e 199.
- ANTONIOU, G. et al. *A Semantic Web Primer*. 3rd. ed. Cambridge, MA, USA: The MIT Press, 2012. 288 p. (Cooperative Information Systems). Citado na página 52.
- ARCHER, C.; STINSON, M. *Object-Oriented Software Measures*. [S.l.], 1995. Citado 3 vezes nas páginas 142, 143 e 263.
- AROUCK, O. *Atributos de qualidade da informação*. Dissertação (Mestrado) — Faculdade de Ciência da Informação da Universidade de Brasília, 2011. Citado 3 vezes nas páginas 144, 146 e 183.
- ARP, R.; SMITH, B.; SPEAR, A. *Building Ontologies with Basic Formal Ontology*. Cambridge, MA, USA: The MIT Press, 2015. 248 p. Citado 7 vezes nas páginas 95, 96, 97, 98, 105, 187 e 265.
- ARUNA, T.; SARANYA, K.; BHANDARI, C. A survey on ontology evaluation tools. In: *2011 International Conference on Process Automation, Control and Computing*. [S.l.: s.n.], 2011. p. 1–5. Citado na página 111.
- ASQ. *Quality Glossary*. 2017. Acessado em 27 de abril de 2017. Disponível em: <<https://asq.org/quality-resources/quality-glossary>>. Citado 6 vezes nas páginas 137, 138, 142, 263, 264 e 266.
- AUER, S. Linked open data - creating knowledge out of interlinked data. In: _____. New York, NY, USA: Springer, 2014. (Lecture Notes in Computer Science / Information Systems and Applications), cap. Introduction to LOD2, p. 1–17. Citado na página 115.
- BABU, L. et al. Archvoc: Towards an ontology for software architecture. In: *Proceedings of the Second Workshop on SHaring and Reusing architectural Knowledge Architecture, Rationale, and Design Intent, SHARK/ADI '07: ICSE Workshops 2007*. Washington, DC, USA: IEEE Computer Society, 2007. p. 5. Citado na página 277.
- BAEZA-YATES, R.; RIBEIRO-NETO, B. *Modern Information Retrieval*. 2nd. ed. New York, NY, USA: Addison Wesley, 2011. 944 p. Citado 7 vezes nas páginas 49, 50, 51, 55, 57, 58 e 74.
- BAGHDADI, Y. A metadata for web services architecture: A framework for service-oriented software development. In: *2009 5th IEEE GCC Conference & Exhibition*. Kuwait City, Kuwait: [s.n.], 2009. p. 1 – 6. Citado na página 169.

- BAGIAMPOU, M.; KAMEAS, A. A use case diagrams ontology that can be used as common reference for software engineering education. In: *6th IEEE International Conference Intelligent Systems (IS)*. Sofia: IEEE, 2012. p. 35–40. Citado na página 279.
- BANI-SALAMEH, H.; JEFFERY, C.; AL-GHARAIBEH, J. A social collaborative virtual environment for software development. In: *Proceedings of the 2010 International Symposium on the Collaborative Technologies and Systems*. Lombard, IL, USA: IEEE, 2010. p. 46–55. Citado na página 33.
- BAONZA, M. *NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse*. Tese (Doutorado) — Universidad Politécnica de Madrid, June 2010. Acessado em: 17 de abril de 2015. Disponível em: <<http://oa.upm.es/3879/>>. Citado 6 vezes nas páginas 105, 109, 110, 112, 187 e 265.
- BARBOSA, E.; NAKAGAWA, E.; MALDONADO, J. Towards the establishment of an ontology of software testing. In: *Proceedings of the Eighteenth International Conference on Software Engineering & Knowledge Engineering (SEKE'2006)*. San Francisco Bay, CA, USA: [s.n.], 2006. p. 522–525. Citado na página 277.
- BARCELLOS, M.; FALBO, R. A software measurement task ontology. In: *28th Annual ACM Symposium on Applied Computing, SAC 2013*. Coimbra, Portugal: [s.n.], 2013. p. 311–318. Citado na página 279.
- BARRE, K. L. Facet analysis. *Annual Review of Information Science and Technology*, Wiley Subscription Services, Inc., A Wiley Company, v. 44, n. 1, p. 243–284, 2010. Citado 6 vezes nas páginas 67, 68, 70, 71, 104 e 187.
- BASEL UNIVERSITY LIBRARY. *Basel Register of Thesauri, Ontologies & Classifications (BARTOC)*. 2017. Acessado em 6 de maio de 2017. Disponível em: <<https://bartoc.org/en>>. Citado na página 112.
- BASILI, V.; CALDIERA, G.; ROMBACH, H. Encyclopedia of software engineering - 2 volume set. In: _____. [S.l.]: John Wiley & Sons, 1994. cap. Goal Question Metric Paradigm, p. 528–532. Citado 3 vezes nas páginas 142, 143 e 144.
- BATES, M. Information and knowledge: an evolutionary framework for information science. *Information Research*, v. 10, n. 4, July 2005. Acessado em: 2 de fevereiro de 2015. Disponível em: <<http://www.informationr.net/ir/10-4/paper239.html>>. Citado na página 46.
- BAUER, F.; KALTENBÖCK, M. *Linked Open Data: The Essentials: A Quick Start Guide for Decision Makers*. Vienna, Austria: Edition mono/monochrom, 2012. 59 p. Acessado em 2 de maio de 2015. Disponível em: <<http://www.semantic-web.at/LOD-TheEssentials.pdf>>. Citado na página 115.
- BCA. *The Bliss Bibliographic Classification : history & description*. 2015. Acessado em 22 de novembro de 2016. Disponível em: <<http://www.blissclassification.org.uk/bchist.shtml>>. Citado na página 69.
- BECK, K. Embracing change with extreme programming. *Computer*, v. 32, n. 10, p. 70–77, October 1999. Citado na página 161.

BECK, K.; ANDRES, C. *Extreme Programming Explained: Embrace Change*. 2nd. ed. Boston, MA, USA: Addison-Wesley, 2004. Citado na página 161.

BEGHTOL, C. Semantic validity: Concepts of warrant in bibliographic classification systems. *Library Resources and Technical Services*, v. 30, n. 2, p. 109–125, 1986. Citado 2 vezes nas páginas 90 e 91.

BELKIN, N. Information concepts for information science. *Journal of Documentation*, v. 34, n. 1, p. 55–85, 1978. Citado na página 46.

BELNAP, N. On rigorous definitions. *Philosophical Studies: An International Journal for Philosophy in the Analytic Tradition*, v. 72, n. 2/3, p. 115–146, 1993. Citado na página 40.

BERGAMASCHI, S.; MARTOGLIA, R.; SORRENTINO, S. Exploiting semantics for filtering and searching knowledge in a software development context. *Knowledge and Information Systems*, v. 45, n. 2, p. 295–318, 2014. ISSN 0219-3116. Citado na página 33.

BERMEJO, C.; RUBIO, A.; ROJO, A. Desarrollo de lenguajes documentales formalizados en lengua española: II evaluación de los tesauros disponibles en lengua española. *Revista española de Documentación Científica*, v. 12, n. 3, p. 283–305, 1989. Citado na página 141.

BERNARD, S. *An Introduction to Enterprise Architecture*. 3rd. ed. Bloomington, IN, USA: AuthorHouse, 2012. 340 p. Citado na página 176.

BERNERS-LEE, T. *Linked Data*. 2006. Acessado em 1 de maio de 2015. Disponível em: <<http://www.w3.org/DesignIssues/LinkedData.html>>. Citado na página 115.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. The semantic web. *Scientific American*, p. 29–37, May 2001. Citado 2 vezes nas páginas 95 e 265.

BERTO, M.; VALLECILLO, A.; GARCÍA, F. Ontologies for software engineering and software technology. In: _____. [S.l.]: Springer, 2006. cap. An Ontology for Software Measurement, p. 175–196. Citado na página 278.

BINKLEY, D.; LAWRIE, D. Information retrieval applications in software development. In: LAPLANTE, P. (Ed.). *Encyclopedia of Software Engineering*. Boca Raton, FL, USA: Auerbach Publications, 2010. v. 1. Citado 4 vezes nas páginas 33, 56, 83 e 164.

BISLIMOVSKA, B. et al. Search upon UML repositories with text matching techniques. In: *2012 ICSE Workshop on Search-Driven Development - Users, Infrastructure, Tools and Evaluation (SUITE)*. Zurich: IEEE, 2012. p. 9–12. Citado 2 vezes nas páginas 32 e 164.

BIZER, C. *Quality-Driven Information Filtering in the Context of Web-Based Information Systems*. Tese (Doutorado) — Freie Universität Berlin, 2007. Citado 3 vezes nas páginas 137, 138 e 139.

BLUMAUER, A. *Taxonomy Quality Assessment: Tools & Techniques*. 2016. Acessado em 6 de maio de 2017. Disponível em: <<https://pt.slideshare.net/semwebcompany/taxonomy-quality-assessment>>. Citado 2 vezes nas páginas 111 e 140.

BORG, M. Embrace your issues: Compassing the software engineering landscape using bug reports. In: *Proceedings of the 29th ACM/IEEE International Conference on Automated Software Engineering*. New York, NY, USA: ACM, 2014. (ASE '14), p. 891–894. ISBN 978-1-4503-3013-8. Citado na página 164.

- BORGES, A. et al. Ontologies supporting the distributed software development: a systematic mapping study. In: *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*. New York, NY, USA: ACM, 2013. p. 153–164. Citado na página 166.
- BORST, W. *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. Tese (Doutorado) — Centre for Telematics and Information Technology, The Netherlands, 1997. Citado 5 vezes nas páginas 96, 97, 132, 260 e 265.
- BOSU, M.; MACDONELL, S. A taxonomy of data quality challenges in empirical software engineering. In: *2013 22nd Australasian Conference on Software Engineering, ASWEC 2013*. Melbourne, VIC, Australia: [s.n.], 2013. p. 97–106. Citado na página 274.
- BOTZENHARDT, A.; MAEDCHE, A.; WIESNER, J. Developing a domain ontology for software product management. In: *2011 5th International Workshop on Software Product Management, IWSPM 2011*. Trento, Italy: IEEE, 2011. p. 7–16. Citado na página 278.
- BRAGA, R.; WERNER, C.; MATTOSO, M. Odyssey-search: A multi-agent system for component information search and retrieval. *Journal of Systems and Software*, v. 79, n. 2, p. 204 – 215, February 2006. ISSN 0164-1212. Citado na página 164.
- BRAMAN, S. Defining information: An approach for policymakers. *Telecommunications Policy*, v. 13, n. 3, p. 233–242, September 1989. Citado na página 46.
- BRANK, J.; GROBELNIK, M.; MLADENIĆ, D. A survey of ontology evaluation techniques. In: *Proc. of 8th Int. multi-conf. Information Society*. [S.l.: s.n.], 2005. Citado 2 vezes nas páginas 141 e 142.
- BRASCHER, M.; CAFÉ, L. Organização da informação ou organização do conhecimento? In: ANCIB. *IX ENANCIB*. São Paulo, Brasil, 2008. Citado na página 58.
- BRASETHVIK, T. *Conceptual modelling for domain specific document description and retrieval - An approach to semantic document modelling*. Tese (Doutorado) — Norwegian University of Science and Technology, 2004. Citado 3 vezes nas páginas 50, 62 e 123.
- BREIS, J.; DUQUE-RAMOS, A.; MARTINEZ, M. *Oquare Wiki*. 2016. Acessado em 9 de junho de 2017. Disponível em: <http://miuras.inf.um.es/oquarewiki/index.php5/Main_Page>. Citado 3 vezes nas páginas 146, 184 e 186.
- BRITO, T. et al. A search service for software components based on a semi-structured data representation model. In: *Information Technology: New Generations, 2009. ITNG '09. Sixth International Conference on*. Las Vegas, NV, USA: [s.n.], 2009. p. 1479–1484. Citado na página 164.
- BROUGHTON, V. *Essential thesaurus construction*. London: Facet Publishing, 2006. 296 p. Citado 3 vezes nas páginas 71, 72 e 200.
- BROUGHTON, V. The need for a faceted classification as the basis of all methods of information retrieval. *Aslib Proceedings: New Information Perspectives*, v. 58, n. 1/2, p. 49–72, 2006. Citado 4 vezes nas páginas 67, 68, 69 e 71.
- BROUGHTON, V.; SLAVIC, A. Building a faceted classification for the humanities: principles and procedures. *Journal of Documentation*, v. 63, n. 5, p. 727–754, 2007. Citado na página 71.

- BSI. *The benefits of using standards*. 2015. Acessado em 5 de março de 2015. Disponível em: <<http://www.bsigroup.com/en-GB/standards/benefits-of-using-standards/>>. Citado na página 117.
- BUCKLAND, M. Information as thing. *Journal of the American Society for Information Science*, v. 42, n. 5, p. 351–360, 1991. Citado na página 46.
- BUCKLAND, M. What is a document ? *Journal of the American Society for I*, v. 48, n. 9, p. 804–809, 1997. Citado na página 48.
- BUCKLEY, J. et al. Towards a taxonomy of software change. *Journal of Software Maintenance and Evolution: Research and Practice*, v. 17, n. 5, p. 309–332, 2005. Citado na página 273.
- BURTON-JONES, A. et al. A semiotic metrics suite for assessing the quality of ontologies. *Data & Knowledge Engineering*, v. 55, n. 1, p. 84–102, 2005. Citado 4 vezes nas páginas 152, 183, 184 e 186.
- CALAZANS, A. Qualidade da informação: conceitos e aplicações. *TransInformação*, v. 20, n. 1, p. 29–45, 2008. Citado na página 139.
- CALERO, C.; RUIZ, F.; PIATTIN, M. *Ontologies for Software Engineering and Software Technology*. New York: Springer, 2006. 339 p. Citado 2 vezes nas páginas 33 e 96.
- CAPLAN, P. *Metadata Fundamentals for All Librarians*. 1st. ed. Chicago: American Library Association, 2003. 192 p. Citado 11 vezes nas páginas 64, 66, 81, 84, 88, 113, 120, 123, 124, 263 e 269.
- CAPURRO, R.; HJØRLAND, B. The concept of information. *Annual Review of Information Science and Technology*, v. 37, n. 1, p. 343–411, 2003. Citado na página 46.
- CARACCILO, C. et al. *VocBench v2.3 User Manual*. [S.l.], 2015. Citado na página 207.
- CARACCILO, C. et al. Thesaurus maintenance, alignment and publication as linked data: the agrovoc use case. *Journal of Metadata, Semantics and Ontologies*, v. 7, n. 1, p. 65–75, 2012. Citado na página 117.
- CARR, M. et al. *Taxonomy-based risk identification*. Pittsburgh, PA, USA, 1993. Acessado em 25 de março de 2015. Disponível em: <<http://www.sei.cmu.edu/reports/93tr006.pdf>>. Citado 3 vezes nas páginas 273, 274 e 275.
- CARVALLO, J. et al. Characterization of a taxonomy for business applications and the relationships among them. In: *In Procs. 3 rd ICCBSS, LNCS*. [S.l.: s.n.], 2004. p. 1–4. Citado na página 275.
- CAVALCANTI, Y. et al. Challenges and opportunities for software change request repositories: a systematic mapping study. *Journal of Software: Evolution and Process*, v. 26, n. 7, p. 620–653, July 2014. ISSN 2047-7481. Citado na página 164.
- CAWKELL, T. Information architecture. In: FEATHER J. & STURGES, P. (Ed.). *International encyclopedia of information and library science*. 2nd. ed. London, UK: Routledge, 2003. Citado na página 76.

- CAZZOLA, W. et al. Co-evolving application code and design models by exploiting meta-data. In: *Proceedings of the 2007 ACM symposium on Applied computing*. New York, NY, USA: ACM, 2007. p. 1275–1279. Citado na página 170.
- CESARE, S. et al. Ontology-driven software engineering 2010. In: *Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion*. New York, NY, USA: ACM, 2010. (SPLASH '10), p. 279–280. ISBN 978-1-4503-0240-1. Citado na página 167.
- CHAVES, A. et al. Ontodisenv1: an ontology to support global software development. *CLEI Electronic Journal*, v. 14, n. 2, 2011. Citado na página 279.
- CHEN, C.; MATTHEWS, M. Metrics for evaluating the semantic implications of changes in evolving ontologies. In: *SWWS*. [S.l.: s.n.], 2008. Citado na página 184.
- CHEN, M.; CHU, H.; CHEN, Y. Developing a semantic-enable information retrieval mechanism. *Expert Systems with Applications*, v. 37, p. 322–340, 2010. Citado na página 52.
- CHEN, Y.; CHEN, S.; LIN, S. A metadata lifecycle model for digital libraries: methodology and application for an evidence-based approach to library research. In: *World Library and Information Congress: 69th IFLA General Conference and Council*. Berlin: [s.n.], 2003. Citado 2 vezes nas páginas 121 e 125.
- CHOPEY, M. Metadata: A cataloger's primer. In: _____. [S.l.]: Routledge, 2005. cap. Planning and Implementing a Metadata-Driven Digital Repository, p. 255 – 287. Citado 2 vezes nas páginas 128 e 129.
- CHU, H. *Information Representation and Retrieval in the Digital Age*. 2nd. ed. Medfor, NJ: Information Today, Inc., 2010. 264 p. (ASIST Monograph Series). Citado 16 vezes nas páginas 50, 51, 52, 55, 57, 66, 72, 73, 74, 83, 84, 89, 90, 114, 120 e 264.
- CIESIN. *CIESIN Metadata Guidelines*. 1998. Acessado em 12 de dezembro de 2016. Disponível em: <<http://www.ciesin.org/metadata/documentation/guidelines/>>. Citado 2 vezes nas páginas 127 e 206.
- CMMI PRODUCT TEAM. *CMMI for Development, Version 1.3*. [S.l.], 2010. Acessado em 12 de maio de 2015. Disponível em: <http://resources.sei.cmu.edu/asset_files/TechnicalReport/2010_005_001_15287.pdf>. Citado 3 vezes nas páginas 33, 171 e 172.
- COAD, P.; YOURDON, E. *Object Oriented Analysis*. [S.l.]: Prentice Hall PTR, 1990. (Yourdon Press Computing Series). Citado na página 135.
- CORCHO, O.; FERNÁNDEZ-LÓPEZ, M.; GÓMEZ-PÉREZ, A. Methodologies, tools and languages for building ontologies: Where is the meeting point? *Data & Knowledge Engineering*, v. 46, p. 41 – 64, 2003. Citado 2 vezes nas páginas 106 e 111.
- CORCHO, O.; GÓMEZ-PÉREZ, A. *Evaluating Knowledge Representation and Reasoning Capabilities of Ontology Specification Languages*. 2000. ECAI'00 Workshop on Applications of Ontologies and Problem Solving Methods. Citado na página 111.

- CORCHO, O.; POVEDA-VILLALÓN, M.; GÓMEZ-PÉREZ, A. *Ontology Engineering in the Era of Linked Data*. 2015. Bulletin of the American Society for Information Science and Technology. Acessado em 18 de junho de 2015. Disponível em: <https://www.asis.org/Bulletin/Apr-15/AprMay15_Corcho_EtAl.html>. Citado 7 vezes nas páginas 97, 98, 99, 104, 116, 132 e 260.
- CORONADO, S. et al. The nci thesaurus quality assurance life cycle. *Journal of Biomedical Informatics*, v. 42, p. 530–539, 2009. Citado na página 152.
- CORREA, D.; SUREKA, A. Integrating issue tracking systems with community-based question and answering websites. In: *Software Engineering Conference (ASWEC), 2013 22nd Australian*. Melbourne, VIC, Australia: IEEE, 2013. p. 88–96. ISSN 1530-0803. Citado na página 164.
- COSTA, I. *Um método para Arquitetura da Informação: Fenomenologia como base para o desenvolvimento de arquiteturas da informação aplicadas*. Dissertação (Mestrado) — Faculdade de Ciência da Informação, Universidade de Brasília, 2009. Citado na página 36.
- COYLE, K. *Guidelines for Dublin Core Application Profiles*. 2009. Acessado em 25 de abril de 2015. Disponível em: <<http://dublincore.org/documents/profile-guidelines/>>. Citado 2 vezes nas páginas 125 e 130.
- CRANFIELD, S.; PURVIS, M. Uml as an ontology modelling language. In: *Proceedings of the 1999 International Conference on Intelligent Information Integration - Volume 23*. Aachen, Germany, Germany: CEUR-WS.org, 1999. (III'99), p. 44–51. Citado na página 40.
- CRUSE, A. *Meaning in Language: An introduction to semantics and pragmatics*. 3rd. ed. [S.l.]: Oxford University Press, 2011. 512 p. (Oxford Textbooks in Linguistics). Citado 3 vezes nas páginas 60, 88 e 263.
- DAHLBERG, I. *Ontical Structures and Universal Classification*. [S.l.]: Sarada Ranganathan Endowment for Library Science, 1978. Citado 2 vezes nas páginas 60 e 61.
- DAHLBERG, I. A referent-oriented, analytical concept theory of interconcept. *International Classification*, v. 5, n. 3, p. 142–151, 1978. Citado 2 vezes nas páginas 60 e 61.
- DAHLBERG, I. Teoria do conceito. *Ci. Inf.*, v. 7, n. 2, p. 101–107, 1978. Citado 3 vezes nas páginas 60, 61 e 63.
- DAHLBERG, I. Current trends in knowledge organization. In: MARCO, F. J. G. (Ed.). *Actas del I Encuentro de ISKO-España, 1993, Madrid*. [S.l.: s.n.], 1993. Organización del conocimiento en sistemas de información y documentación 1, p. 7–25. Citado 2 vezes nas páginas 60 e 61.
- DAHLBERG, I. Concepts and terms - isko's major challenge. *Knowledge Organization*, v. 36, n. 2/3, p. 169–177, 2009. Citado 2 vezes nas páginas 60 e 61.
- DAHLBERG, I. How to improve isko's standing: ten desiderata for knowledge organization. *Knowledge Organization*, v. 38, n. 1, p. 68–74, 2011. Citado 2 vezes nas páginas 60 e 61.

- DAMA. *The DAMA Guide to the Data Management Body of Knowledge*. 1st ed.. ed. Bradley Beach, New Jersey, USA: Technics Publications, 2010. 430 p. Citado 4 vezes nas páginas 75, 119, 125 e 259.
- DAMASEVICIUS, R. 7do: a model for ontology complexity evaluation. *Computing and Information Systems*, v. 13, n. 1, 2009. Citado 2 vezes nas páginas 152 e 183.
- DAMASEVICIUS, R.; STUIKYS, V. Taxonomy of the fundamental concepts of metaprogramming. *Information Technology and Control*, v. 37, n. 2, p. 124–132, 2008. Citado na página 272.
- DAMIANI, E.; FUGINI, M.; BELLETTINI, C. A hierarchy-aware approach to faceted classification of object-oriented components. *ACM Transactions on Software Engineering and Methodology*, v. 8, n. 3, p. 215–262, 1999. Citado na página 166.
- DAVIES, J. Theory and applications of ontology: Computer applications. In: _____. [S.l.]: Springer, 2010. cap. Lightweight Ontologies, p. 197–229. Citado 9 vezes nas páginas 52, 73, 93, 94, 95, 99, 123, 128 e 265.
- DAY, M. *DCC Digital Curation Manual Instalment on Metadata*. [S.l.], 2005. Acessado em: 26 de fevereiro de 2015. Disponível em: <<http://www.dcc.ac.uk/sites/default/files/documents/resource/curation-manual/chapters/metadata/metadata.pdf>>. Citado 3 vezes nas páginas 120, 121 e 123.
- DCMI. *DCMI Metadata Terms*. Acessado em 24 de abril de 2015., 2012. Disponível em: <<http://dublincore.org/documents/2012/06/14/dcmi-terms/>>. Citado na página 131.
- DCMI. *Dublin Core Metadata Element Set, Version 1.1*. Acessado em 24 de abril de 2015., 2012. Disponível em: <<http://dublincore.org/documents/dces/>>. Citado na página 130.
- DCMI. *Metadata Basics*. Acessado em 24 de abril de 2015., 2015. Disponível em: <<http://dublincore.org/metadata-basics/>>. Citado 3 vezes nas páginas 120, 121 e 130.
- DELGADO, N.; GATES, A.; ROACH, S. A taxonomy and catalog of runtime software-fault monitoring tools. *IEEE Transactions on Software Engineering*, v. 30, n. 12, p. 859 – 872, 2004. Citado na página 275.
- DENTON, W. *How to Make a Faceted Classification and Put It On the Web*. 2009. Acessado em 28 de novembro de 2016. Disponível em: <<https://www.miskatonic.org/library/facet-web-howto.html>>. Citado 2 vezes nas páginas 69 e 187.
- DIA. *Dia Diagram Editor*. 2015. Acessado em 17 de setembro de 2016. Disponível em: <<http://sourceforge.net/projects/dia-installer/>>. Citado na página 41.
- DICTIONARY.COM. *Thesaurus.com*. 2017. Acessado em 14 de setembro de 2017. Disponível em: <<http://www.thesaurus.com/>>. Citado 2 vezes nas páginas 200 e 203.
- DILLON, A. Information architecture in jasist: Just where did we come from ? *Journal of the American Society for Information Science and Technology*, v. 53, n. 10, p. 821–823, 2002. Citado 3 vezes nas páginas 28, 77 e 78.
- DILLON, T.; SIMMONS, G. Semantic web support for open-source software development. In: *2008 IEEE International Conference on Signal Image Technology and Internet Based Systems*. [S.l.: s.n.], 2008. p. 606 – 613. Citado 2 vezes nas páginas 33 e 163.

DING, W.; LIN, X. *Information Architecture: The Design and Integration of Information Spaces*. [S.l.]: Morgan and Claypool Publishers, 2009. 172 p. (Synthesis Lectures on Information Concepts, Retrieval, and Services). Citado na página 28.

DLESE. *Metadata, collection building and cataloging*. 2005. Acessado em 15 de maio de 2015. Disponível em: <<http://www.dlese.org/Metadata/>>. Citado 9 vezes nas páginas 48, 49, 83, 88, 100, 120, 121, 184 e 263.

DODEBEI, V. *Tesouro: linguagem de representação da memória documentária*. 1a.. ed. Rio de Janeiro: Intertexto, 2002. 120 p. Citado 7 vezes nas páginas 62, 64, 83, 84, 85, 86 e 264.

DOERR, M. *DCC Digital Curation Manual Instalment on Ontologies*. [S.l.], 2008. Acessado em: 26 de fevereiro de 2015. Disponível em: <<http://www.dcc.ac.uk/sites/default/files/documents/resource/curation-manual/chapters/ontologies/ontologies.pdf>>. Citado na página 97.

DONG, H.; HUSSAIN, F.; CHANG, E. A survey in semantic search technologies. In: *Second IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2008)*. Phitsanulok, Thailand: IEEE, 2008. p. 403–408. Citado na página 54.

DROMEY, R. A model for software product quality. *IEEE Transactions on Software Engineering*, v. 21, n. 2, p. 146–162, 1995. Citado na página 151.

DROMEY, R. Cornering the chimera. *IEEE Software*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 13, n. 1, p. 33–43, jan. 1996. ISSN 0740-7459. Citado 3 vezes nas páginas 147, 148 e 181.

DUARTE, J. *Uma Arquitetura Ágil da Informação Organizacional*. Tese (Doutorado) — Universidade de Brasília, Faculdade de Ciência da Informação, Brasília, 2011. Citado na página 28.

DUARTE, K.; FALBO, R. Uma ontologia de qualidade de software. In: *Anais do VII Workshop de Qualidade de Software*. [S.l.: s.n.], 2000. Citado na página 280.

DUQUE-RAMOS, A. et al. Evaluation of the oquare framework for ontology quality. *Expert Systems with Applications*, v. 40, p. 2703, 01/2013 2013. Citado 6 vezes nas páginas 146, 151, 152, 181, 183 e 184.

DUQUE-RAMOS, A. et al. Oquare: A square-based approach for evaluating the quality of ontologies. *Journal of Research and Practice in Information Technology*, v. 43, n. 2, p. 159–176, 2011. Citado 3 vezes nas páginas 141, 142 e 151.

DUVAL, E. et al. Metadata principles and practicalities. *D-Lib Magazine*, v. 8, n. 4, 2002. Citado 5 vezes nas páginas 121, 122, 123, 124 e 125.

ECLIPSE FOUNDATION. *OpenUp*. 2012. Acessado em 8 de maio de 2015. Disponível em: <<http://epf.eclipse.org/wikis/openup/>>. Citado na página 162.

ELIAS, G. et al. X-arm: an asset representation model for component repository systems. In: *SAC '06 Proceedings of the 2006 ACM symposium on Applied computing*. New York, NY, USA: ACM, 2006. p. 1690–1694. Citado 2 vezes nas páginas 169 e 170.

- ELKIN, P.; BROWN, S.; CHUTE, C. Guideline for health informatics: controlled health vocabularies—vocabulary structure and high-level indicators. In: _____. *Medinfo. MEDINFO*. Pt 1. [S.l.: s.n.], 2001. v. 10, p. 191–195. Citado na página 152.
- ENGLISH, L. *Information Quality Applied: Best Practices for Improving Business Information, Processes and Systems*. [S.l.]: Wiley, 2009. Citado na página 138.
- EPPLER, M. A generic framework for information quality in knowledge-intensive processes. In: *Proceedings of the MIT Information Quality 2001 Conference*. [S.l.: s.n.], 2001. Citado 2 vezes nas páginas 149 e 181.
- EPPLER, M.; WITTIG, D. Conceptualizing information quality: A review of information quality frameworks from the last ten years. In: *Proceedings of the 2000 Conference on Information Quality*. [S.l.: s.n.], 2000. Citado 4 vezes nas páginas 139, 144, 149 e 181.
- ERENKRANTZ, J.; TAYLOR, R. Supporting distributed and decentralized projects: drawing lessons from the open source community. In: *Proceedings of the 1st Workshop on Open Source in an Industrial Context*. Anaheim, California, USA: [s.n.], 2003. Citado na página 162.
- EUROPEAN UNION. *EuroVoc Thesaurus*. 2017. Acessado em 7 de junho de 2017. Disponível em: <<http://eurovoc.europa.eu/>>. Citado 4 vezes nas páginas 179, 183, 200 e 203.
- FALBO, R. Experiences in using a method for building domain ontologies. *Banff. Proceedings of the 16th International Conference on Software Engineering and Knowledge Engineering*, p. 474–477, 2004. Citado na página 278.
- FALBO, R.; BERTOLLO, G. Establishing a common vocabulary for helping organizations to understand software processes. In: *Proceedings of the 1st VORTE*. Enschede, The Netherlands: [s.n.], 2005. Citado 2 vezes nas páginas 33 e 279.
- FALBO, R.; MENEZES, C.; ROCHA, A. A systematic approach for building ontologies. In: *Proceedings of the 6th Ibero-American Conference on AI: Progress in Artificial Intelligence*. London, UK, UK: Springer-Verlag, 1998. (IBERAMIA '98), p. 349–360. Citado na página 279.
- FAO. *AGROVOC Multilingual agricultural thesaurus*. 2015. Acessado em 2 de setembro de 2016. Disponível em: <<http://aims.fao.org/vest-registry/vocabularies/agrovoc-multilingual-agricultural-thesaurus>>. Citado 5 vezes nas páginas 41, 94, 111, 200 e 203.
- FAO. *VocBench 2.0*. 2015. Acessado em 19 de setembro de 2015. Disponível em: <<http://vocbench.uniroma2.it/>>. Citado 2 vezes nas páginas 41 e 111.
- FEATHER, J.; STURGES, P. *International Encyclopedia of Information and Library Science*. 2nd. ed. London, UK: Routledge, 2003. 688 p. Citado 3 vezes nas páginas 65, 66 e 260.
- FELDERER, M.; SCHIEFERDECKER, I. A taxonomy of risk-based testing. *International Journal on Software Tools for Technology Transfer*, v. 16, n. 5, p. 559–568, October 2014. Citado na página 275.

- FERNÁNDEZ-LÓPEZ, M.; GÓMEZ-PÉREZ, A. Overview and analysis of methodologies for building ontologies. *The Knowledge Engineering Review*, Cambridge University Press, New York, NY, USA, v. 17, n. 2, p. 129–156, 2002. ISSN 0269-8889. Citado 2 vezes nas páginas 38 e 106.
- FERNÁNDEZ, M.; GÓMEZ-PÉREZ, A.; JURISTO, N. Methontology: From ontological art towards ontological engineering. In: *Proceedings of the Ontological Engineering AAAI-97 Spring Symposium Series*. Stanford University, USA: [s.n.], 1997. Citado 2 vezes nas páginas 107 e 187.
- FERRÉ, X.; VEGAS, S. An evaluation of domain analysis methods. In: *4th CAiSE/IFIP8.1 International Workshop in Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD99)*. [S.l.: s.n.], 1999. Citado 2 vezes nas páginas 134 e 135.
- FERREYRA, D. *TemaTres*. 2016. Acessado em 2 de fevereiro de 2016. Disponível em: <<http://www.vocabularyserver.com/>>. Citado na página 111.
- FIDEL, R. Who needs controlled vocabulary? *Special Libraries*, v. 83, n. 1, p. 1–9, 1992. Citado na página 83.
- FIDEL, R.; PEJTERSEN, A. From information behavior research to the design of information systems: the cognitive work analysis. *Information Research*, v. 10, n. 1, 2004. Acessado em 22 de maio de 2015. Disponível em: <<http://www.informationr.net/ir/10-1/paper210.html>>. Citado na página 103.
- FLURI, B.; GALL, H. Classifying change types for qualifying change couplings. In: *Proceeding of the 14th IEEE International Conference on Program Comprehension (ICPC 2006)*. Athens, Greece: [s.n.], 2006. p. 35–45. Citado na página 273.
- FORWARD, A.; LETHBRIDGE, T. A taxonomy of software types to facilitate search and evidence-based software engineering. In: *Proceedings of the 2008 Conference of the Center for Advanced Studies, CASCON'08*. New York, NY, USA: ACM, 2008. (CASCON '08), p. 14:179–14:191. Citado na página 275.
- FOX, M.; BARBUCEANU, M.; GRUNINGER, M. An organisation ontology for enterprise modeling: Preliminary concepts for linking structure and behaviour. *Computers in Industry*, v. 29, p. 123–134, 1996. Citado 2 vezes nas páginas 152 e 183.
- FRANCART, T. *SKOS Play !* 2017. Acessado em 25 de julho de 2017. Disponível em: <<http://labs.sparna.fr/skos-play/>>. Citado 2 vezes nas páginas 41 e 111.
- FREUND, L.; TOMS, E.; WATERHOUSE, J. Modeling the information behaviour of software engineers using a work-task framework. *Proceedings of the Annual Meeting of the American Society for Information Science and Technology*, v. 42, n. 1, p. 187–214, 2005. Citado 2 vezes nas páginas 33 e 155.
- FUGGETTA, A. Software process: a roadmap. In: *Proceedings of the Conference on The Future of Software Engineering*. New York, NY, USA: ACM, 2000. (ICSE '00), p. 25–34. Citado 4 vezes nas páginas 33, 156, 157 e 266.
- GACEK, C.; ARIEF, B. The many meanings of open source. *IEEE Software*, v. 21, n. 1, p. 34–40, Jan 2004. ISSN 0740-7459. Citado na página 273.

GALIN, D. *Software Quality Assurance: From Theory to Implementation*. 1st. ed. [S.l.]: Pearson, 2003. 616 p. Citado 2 vezes nas páginas 33 e 171.

GALSTER, M.; BUCHERER, E. A taxonomy for identifying and specifying non-functional requirements in service-oriented development. In: *2008 IEEE Congress on Services, SERVICES 2008*. Honolulu, HI, United States: [s.n.], 2008. PART 1, p. 345–352. Citado na página 272.

GANGEMI, A. et al. Modelling ontology evaluation and validation. In: _____. *The Semantic Web: Research and Applications: 3rd European Semantic Web Conference, ESWC 2006 Budva, Montenegro, June 11-14, 2006 Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 140–154. Citado 2 vezes nas páginas 182 e 184.

GANTTPROJECT TEAM. *GanttProject Free project scheduling and management app for Windows, OSX and Linux*. 2015. Acessado em 19 de setembro de 2015. Disponível em: <<http://www.ganttproject.biz/>>. Citado na página 41.

GARCÍA, R. et al. Towards a consistent terminology for software measurement. *Information and Software Technology*, v. 48, n. 8, p. 631–644, August 2006. Citado na página 278.

GARSHOL, L. Metadata? thesauri? taxonomies? topic maps! making sense of it all. *Journal of Information Science*, v. 30, n. 4, p. 378–391, 2004. Citado 8 vezes nas páginas 67, 83, 89, 92, 120, 121, 264 e 269.

GARZAS, J.; PIATTINI, M. An ontology for microarchitectural design knowledge. *IEEE Software*, v. 22, n. 2, p. 28–33, 2005. Citado 2 vezes nas páginas 97 e 276.

GARZAS, J.; PIATTINI, M. An ontology for understanding and applying object-oriented design knowledge. *International Journal of Software Engineering and Knowledge Engineering*, v. 17, n. 3, p. 407–421, 2007. Citado na página 276.

GASEVIC, D.; KAVIANI, N.; MILANOVIC, M. Handbook on ontologies. In: _____. [S.l.]: Springer Berlin Heidelberg, 2009. cap. Ontologies and Software Engineering, p. 593–615. Citado na página 168.

GAVRILOVA, T.; GOROVOY, V.; BOLOTNIKOVA, E. Evaluation of the cognitive ergonomics of ontologies on the basis of graph analysis. *Scientific and Technical Information Processing*, v. 37, n. 6, p. 398–406, 2010. Citado 3 vezes nas páginas 141, 142 e 184.

GEORGETOWN UNIVERSITY MEDICAL CENTER. *PIR Protein Information Resource*. 2017. Acessado em 29 de abril de 2017. Disponível em: <<http://pir.georgetown.edu/pro/>>. Citado na página 99.

GIESS, M.; WILD, P.; MCMAHON, C. The generation of faceted classification schemes for use in the organisation of engineering design documents. *International Journal of Information Management*, v. 28, p. 379–390, 2008. Citado 12 vezes nas páginas 32, 46, 52, 66, 67, 68, 69, 71, 84, 89, 90 e 200.

GIL, A. *Como Elaborar Projetos de Pesquisa*. Quinta edição. São Paulo: Editora Atlas, 2010. 184 p. Citado 3 vezes nas páginas 29, 38 e 39.

- GILCHRIST, A. Thesauri, taxonomies and ontologies - an etymological note. *Journal of Documentation*, v. 59, n. 1, p. 7–18, 2003. Citado na página 92.
- GILL, T. Introduction to metadata. In: _____. Los Angeles, CA, USA: Getty Research Institute, 2008. cap. Metadata and the Web, p. 20–38. Citado 2 vezes nas páginas 120 e 263.
- GILLILAND, A. Introduction to metadata. In: _____. Second edition. Los Angeles, CA, USA: Getty Research Institute, 2008. cap. Setting the Stage, p. 1–19. Citado 5 vezes nas páginas 54, 120, 121, 123 e 125.
- GIUNCHIGLIA, F.; DUTTA, B.; MALTESE, V. *Faceted Lightweight Ontologies*. Italy, 2009. Citado 2 vezes nas páginas 97 e 99.
- GIUNCHIGLIA, F.; DUTTA, B.; MALTESE, V. *From Knowledge Organization to Knowledge Representation*. Italy, 2013. Citado 2 vezes nas páginas 53 e 260.
- GIUNCHIGLIA, F.; ZAIHRAYEU, I. *Lightweight Ontologies*. Italy, 2007. Citado 2 vezes nas páginas 54 e 99.
- GLASS, R.; VESSEY, I. Contemporary application-domain taxonomies. *IEEE Software*, v. 12, n. 4, p. 63–76, July 1995. Citado na página 66.
- GLUSHKO, R. *The Discipline of Organizing*. Cambridge, MA: The MIT Press, 2013. 539 p. Citado 22 vezes nas páginas 24, 47, 48, 56, 58, 59, 62, 65, 67, 73, 74, 78, 82, 83, 89, 90, 101, 187, 259, 260, 265 e 267.
- GÖDERT, W.; HUBRICH, J.; NAGELSCHMIDT, M. *Semantic Knowledge Representation for Information Retrieval*. [S.l.]: De Gruyter Saur, 2014. 294 p. Citado 21 vezes nas páginas 49, 51, 52, 53, 56, 57, 60, 64, 83, 84, 85, 86, 87, 91, 100, 112, 261, 264, 265, 266 e 268.
- GODFREY, M.; TU, Q. Evolution in open source software: A case study. In: *Proceedings International Conference on Software Maintenance*. San Jose: [s.n.], 2000. Citado na página 162.
- GOH, Y. et al. Foundations of computational intelligence vol. 6: Data mining. In: _____. Springer, 2009. cap. From faceted classification to knowledge discovery of semi-structured text records, p. 151–170. Disponível em: <<https://dspace.lboro.ac.uk/2134/8009>>. Citado 11 vezes nas páginas 46, 49, 50, 51, 52, 65, 66, 67, 68, 90 e 264.
- GOMES, H. *Manual de Elaboração de Tesouros Monolíngues*. Brasília, 1990. Citado 4 vezes nas páginas 83, 92, 93 e 95.
- GÓMEZ-PÉREZ, A. Towards a framework to verify knowledge sharing technology. *Expert Systems with Applications*, v. 11, n. 4, p. 519–529, 1996. Citado 2 vezes nas páginas 152 e 183.
- GÓMEZ-PÉREZ, A. Evaluation of ontologies. *International Journal of Intelligent Systems*, v. 16, p. 391–409, 2001. Citado 2 vezes nas páginas 152 e 184.
- GÓMEZ-PÉREZ, A.; CORCHO, O.; FERNÁNDEZ-LÓPEZ, M. *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Secaucus, NJ, USA: Springer-Verlag, 2004. (Advanced Information and Knowledge Processing). Citado 4 vezes nas páginas 97, 99, 104 e 105.

GÓMEZ-PÉREZ, A.; FERNÁNDEZ, M.; VICENTE, A. Towards a method to conceptualize domain ontologies. In: *12th European Conference on Artificial Intelligence (ECAI'96)*. [S.l.: s.n.], 1996. Citado 2 vezes nas páginas 108 e 183.

GONZÁLEZ-PÉREZ, C.; HENDERSON-SELLERS, B. Ontologies for software engineering and software technology. In: _____. [S.l.]: Springer Berlin Heidelberg, 2006. cap. An Ontology for Software Development Methodologies and Endeavours, p. 123–151. Citado 2 vezes nas páginas 33 e 280.

GONZÁLEZ, R.; MEER, K. van der. Standard metadata applied to software retrieval. *JOURNAL OF INFORMATION SCIENCE*, 30, n. 4, p. 300–309, 2004. ISSN 0165-5515. Citado na página 170.

GOTTIPATI, S.; LO, D.; JIANG, J. Finding relevant answers in software forums. In: *Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering*. Washington, DC, USA: IEEE Computer Society, 2011. (ASE '11), p. 323–332. Citado na página 164.

GRAAF, K. Annotating software documentation in semantic wikis. In: *Proceedings of the Fourth Workshop on Exploiting Semantic Annotations in Information Retrieval*. New York, NY, USA: ACM, 2011. (ESAIR '11), p. 5–6. Citado na página 162.

GREEFHORST, D.; KONING, H.; VLIET, H. The many faces of architectural descriptions. *Information Systems Frontiers*, v. 8, p. 103–113, 2006. Citado na página 176.

GREENBERG, J. Metadata: A cataloger's primer. In: _____. [S.l.]: Routledge, 2005. cap. Understanding Metadata and Metadata Schemes, p. 17–36. Citado 3 vezes nas páginas 119, 123 e 124.

GRUBER, T. A translation approach to portable ontology specifications. *Knowledge Acquisition*, v. 5, n. 1, p. 199–220, 1993. Citado 4 vezes nas páginas 95, 132, 260 e 265.

GRUBER, T. Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.*, Academic Press, Inc., Duluth, MN, USA, v. 43, n. 5-6, p. 907–928, dez. 1995. ISSN 1071-5819. Citado na página 152.

GRZYWACZEWSKI, A.; IQBAL, R. Task-specific information retrieval systems for software engineers. *Journal of Computer and System Sciences*, v. 78, n. 4, p. 1204–1218, 2012. Citado 2 vezes nas páginas 32 e 155.

GUARINO, N. Formal ontology and information system. In: *Proceedings of FOIS'98*. Trento, Italy: IOS Press, 1998. Citado na página 97.

GUIZZARDI, G. *Ontological Foundations for Structural Conceptual Models*. Tese (Doutorado) — Telematica Instituut - Centre for Telematics and Information Technology, Enschede, The Netherlands, 2005. Citado 3 vezes nas páginas 132, 260 e 264.

GUMM, D. Distribution dimensions in software development projects: A taxonomy. *IEEE Software*, v. 23, n. 5, p. 45–51, 2006. Citado na página 273.

- GUTWIN, C.; PENNER, R.; SCHNEIDER, K. Group awareness in distributed software development. In: *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*. New York, NY, USA: ACM, 2004. (CSCW '04), p. 72–81. Citado na página 162.
- HAGEDORN, K. *The Information Architecture Glossary*. [S.l.], 2000. Acessado em 28 de janeiro de 2014. Disponível em: <http://argus-acia.com/white_papers/iaglossary.html>. Citado 4 vezes nas páginas 28, 50, 52 e 78.
- HALVORSON, K.; RACH, M. *Content Strategy for the Web*. 2nd. ed. Berkeley, CA: New Riders, 2012. 216 p. Citado na página 28.
- HAMDAN, K. et al. A software cost ontology system for assisting estimation of software project effort for use with case-based reasoning. In: *2006 Innovations in Information Technology*. [S.l.]: IEEE, 2006. Citado na página 278.
- HAPPEL, H.; SEEDORF, S. Applications of ontologies on software engineering. In: *Proceedings of the International Workshop on Semantic Web Enabled Software Engineering (SWESE'06)*. Athens, USA: [s.n.], 2006. Citado na página 168.
- HARPRING, P. *Introduction to Controlled Vocabularies: Terminology for Art, Architecture, and Other Cultural Works*. Los Angeles, CA, USA: Getty Publications, 2010. Acessado em 13 de junho de 2015. Disponível em: <http://www.getty.edu/research/publications/electronic_publications/intro_controlled_vocab/>. Citado 12 vezes nas páginas 73, 81, 82, 89, 91, 100, 101, 104, 117, 140, 187 e 269.
- HARTMANN, J. et al. *Methods for ontology evaluation*. [S.l.], 2005. Citado na página 141.
- HASSAN, A. The road ahead for mining software repositories. In: *Frontiers of Software Maintenance, 2008. FoSM 2008*. [S.l.]: IEEE, 2008. p. 48–57. Citado 2 vezes nas páginas 163 e 164.
- HAWKING, D. Modern information retrieval. In: _____. 2nd. ed. New York, NY, USA: Addison Wesley, 2011. cap. Enterprise Search, p. 645–687. Citado 2 vezes nas páginas 23 e 51.
- HAWKING, D.; ZOBEL, J. Does topic metadata help with web search? *Journal of the American Society for Information Science and Technology*, v. 58, n. 5, p. 613–628, 2007. Citado na página 121.
- HAYNES, D. *Metadata: For Information Management and Retrieval*. London, UK: Facet Publishing, 2004. 186 p. (Become an Expert). Citado na página 121.
- HEATH, T.; BIZER, C. *Linked Data: Evolving the Web into a Global Data Space*. [S.l.]: Morgan & Claypool Publishers, 2011. 129 p. Citado 2 vezes nas páginas 114 e 115.
- HEDDEN, H. *The Accidental Taxonomist*. Medford, N.J.: Information Today, Inc., 2010. 442 p. Citado 20 vezes nas páginas 73, 74, 82, 83, 85, 86, 87, 91, 92, 93, 94, 96, 103, 111, 140, 184, 185, 187, 266 e 269.
- HIEMSTRA, D. Information retrieval: Searching in the 21st century. In: _____. [S.l.]: Wiley, 2009. cap. Information Retrieval Models, p. 1–19. Citado 2 vezes nas páginas 50 e 51.

- HILERA, J.; FERNÁNDEZ-SANZ, L. Developing domain-ontologies to improve software engineering knowledge. In: *Proceedings - 5th International Conference on Software Engineering Advances, ICSEA 2010*. Nice: IEEE, 2010. p. 380–383. Citado 4 vezes nas páginas 34, 167, 279 e 280.
- HILL, J. A software safety risk taxonomy for use in retrospective safety cases. In: *31st Annual IEEE Software Engineering Workshop, SEW-31 2007*. Columbia, MD, United States: IEEE, 2007. p. 179–183. Citado na página 274.
- HILL, J.; VICTOR, D. The product engineering class in the software safety risk taxonomy for building safety-critical systems. In: *Proceedings of the Australian Software Engineering Conference, ASWEC*. Perth, WA, Australia: [s.n.], 2008. p. 617–626. Citado na página 274.
- HINDLE, A.; GERMAN, D.; HOLT, R. What do large commits tell us? a taxonomical study of large commits. In: *Proceedings of the 2008 International Working Conference on Mining Software Repositories*. New York, NY, USA: ACM, 2008. (MSR '08), p. 99–108. Citado na página 272.
- HITZLER, P. et al. *OWL 2 Web Ontology Language Primer (Second Edition)*. 2012. Acessado em 30 de abril de 2015. Disponível em: <<http://www.w3.org/TR/2012/REC-owl2-primer-20121211/>>. Citado na página 117.
- HJØRLAND, B. Domain analysis in information science: Eleven approaches – traditional as well as innovative. *Journal of Documentation*, v. 58, n. 4, p. 422 – 462, 2002. Citado 5 vezes nas páginas 101, 102, 104, 134 e 193.
- HJØRLAND, B. *Multidimensional classification (Multifaceted classification)*. 2006. Acessado em: 24 de novembro de 2016. Disponível em: <http://www.iva.dk/bh/lifeboat_ko/CONCEPTS/multidimensional_classification.htm>. Citado 2 vezes nas páginas 89 e 90.
- HJØRLAND, B. *Enumerative systems*. 2007. Acessado em 14 de novembro de 2016. Disponível em: <http://www.iva.dk/bh/lifeboat_ko/CONCEPTS/enumerative_systems.htm>. Citado na página 66.
- HJØRLAND, B. *Semantic relations (meaning relations)*. 2007. Acessado em: 29 de abril de 2016. Disponível em: <http://www.iva.dk/bh/lifeboat_ko/CONCEPTS/semantic_relations.htm>. Citado 2 vezes nas páginas 87 e 267.
- HJØRLAND, B. *Semantic tool*. 2007. Acessado em: 29 de abril de 2016. Disponível em: <http://www.iva.dk/bh/lifeboat_ko/CONCEPTS/semantic_tools.htm>. Citado 4 vezes nas páginas 23, 53, 261 e 262.
- HJØRLAND, B. *Semantics*. 2008. Acessado em: 29 de abril de 2016. Disponível em: <<http://www.iva.dk/bh/Core%20Concepts%20in%20LIS/articles%20a-z/semantics.htm>>. Citado 2 vezes nas páginas 52 e 267.
- HJØRLAND, B. What is knowledge organization (KO)? *Knowledge Organization*, v. 35, n. 2/3, p. 86–101, 2008. Citado 3 vezes nas páginas 58, 67 e 134.
- HJØRLAND, B. Concept theory. *Journal of the American Society for Information Science and Technology*, v. 60, n. 8, p. 1519–1536, 2009. Citado 3 vezes nas páginas 59, 60 e 268.

- HJØRLAND, B. Is classification necessary after google? *Journal of Documentation*, v. 68, n. 3, p. 299–317, 2012. Citado na página 66.
- HJØRLAND, B. Classical databases and knowledge organization: A case for boolean retrieval and human decision-making during searches. *Journal of the Association for Information Science and Technology*, v. 66, n. 8, p. 1559–1575, 2015. Citado na página 49.
- HJØRLAND, B.; ALBRECHTSEN, H. Toward a new horizon in information science: Domain analysis. *Journal of the American Society for Information Science*, v. 46, n. 6, p. 400 – 425, 1995. Citado na página 134.
- HJØRLAND, B.; PEDERSEN, K. A substantive theory of classification for information retrieval. *Journal of Documentation*, v. 61, n. 5, p. 582 – 597, 2005. Citado 4 vezes nas páginas 31, 60, 65 e 260.
- HLOMANI, H.; STACEY, D. Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey. *Semantic Web Journal*, p. 1–5, 2014. Citado 4 vezes nas páginas 141, 152, 183 e 186.
- HOBEL, H.; REVENKO, A. On the quality of annotations with controlled vocabularies. In: _____. *Collective Online Platforms for Financial and Environmental Awareness: First International Workshop on the Internet for Financial Collective Awareness and Intelligence, IFIN 2016 and First International Workshop on Internet and Social Media for Environmental Monitoring, ISEM 2016, Florence, Italy, September 12, 2016, Revised Selected Papers*. [S.l.]: Springer International Publishing, 2016. p. 98–114. Citado 3 vezes nas páginas 128, 129 e 186.
- HODGE, G. *Systems of Knowledge Organization for Digital Libraries: Beyond Traditional Authority Files*. Washington, DC, USA, 2000. Citado 5 vezes nas páginas 23, 59, 81, 268 e 269.
- HOFF, A.; PARTOVI, H.; THAI, T. *The Open Software Description Format (OSD)*. 1997. Acessado em: 4 de novembro de 2015. Disponível em: <<http://www.w3.org/TR/NOTE-OSD>>. Citado na página 170.
- HOLIBAUGH, R. *Joint Integrated Avionics Working Group (JIAWG) Object-Oriented Domain Analysis Method (JODA)*. [S.l.], 1993. Citado 3 vezes nas páginas 134, 136 e 259.
- HOOLAND, S.; VERBORG, R. *Linked Data for Libraries, Archives and Museums: How to Clean, Link and Publish your Metadata*. London, UK: Facet Publishing, 2014. 254 p. Citado na página 114.
- HOUAISS, A.; VILLAR, M.; FRANCO, F. *Dicionário Houaiss da língua portuguesa*. Primeira edição. Rio de Janeiro: Editora Objetiva, 2001. Citado na página 76.
- HOYER, R.; HOYER, B. What is quality? *Quality Progress*, p. 52–62, 2001. Citado 2 vezes nas páginas 137 e 266.
- IBGE. *Pesquisa de inovação tecnológica 2008*. Rio de Janeiro, RJ, 2010. Citado 2 vezes nas páginas 46 e 47.

- ICHII, M. et al. Software component recommendation using collaborative filtering. In: *Proceedings of the 2009 ICSE Workshop on Search-Driven Development-Users, Infrastructure, Tools and Evaluation*. Washington, DC, USA: IEEE Computer Society, 2009. (SUITE '09), p. 17–20. ISBN 978-1-4244-3740-5. Citado na página 164.
- IEEE. *IEEE std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology*. New York, NY, 1990. Citado 8 vezes nas páginas 54, 63, 76, 155, 259, 261, 263 e 265.
- IEEE. *IEEE 830-1993 IEEE Recommended Practice for Software Requirements Specifications*. New York, NY, 1993. Citado na página 156.
- IEEE. *IEEE 730.1-1995 - IEEE Guide for Software Quality Assurance Planning*. New York, NY, USA, 1996. Citado 2 vezes nas páginas 38 e 264.
- IEEE. *IEEE 1484.12.1 IEEE Standard for Learning Object Metadata*. New York, NY, 2002. Citado 4 vezes nas páginas 120, 121, 130 e 263.
- IEEE. *IEEE 1074:2006 IEEE Standard for Developing a Software Project Lifecycle Process*. New York, NY, 2006. Citado 9 vezes nas páginas 158, 159, 162, 166, 178, 179, 187, 189 e 259.
- IEEE. *IEEE Std 1061TM-1998 (R2009) - IEEE Standard for a Software Quality Metrics Methodology*. 2009. Citado 4 vezes nas páginas 148, 149, 181 e 186.
- IEEE. *IEEE 828:2012 IEEE Standard for Configuration Management in Systems and Software Engineering*. New York, NY, 2012. Citado 3 vezes nas páginas 155, 163 e 269.
- IEEE. *ISO/IEC/IEEE 31320-2:2012(E) - Information technology – Modeling Languages – Part 2: Syntax and Semantics for IDEF1X97 (IDEFobject)*. 2012. Citado 2 vezes nas páginas 132 e 264.
- IEEE. *2014 IEEE Thesaurus Version 1.0*. 2014. Acessado em 30 de março de 2017. Citado 3 vezes nas páginas 93, 200 e 203.
- IEEE. *IEEE Std 730 - 2014 - IEEE Standard for Software Quality Assurance Processes*. New York, NY, USA, 2014. Citado 2 vezes nas páginas 33 e 266.
- IEEE. *SWEBOK: Guide to the Software Engineering Body of Knowledge*. 3.0. ed. IEEE Computer Society, 2014. Acessado em 17 de setembro de 2016. Disponível em: <<http://www.computer.org/portal/web/swebok>>. Citado 10 vezes nas páginas 133, 139, 154, 156, 157, 193, 195, 197, 201 e 275.
- IEEE. *Software and Systems Engineering Vocabulary*. 2015. Acessado em 29 de setembro de 2016. Disponível em: <http://pascal.computer.org/sev_display/index.action>. Citado na página 266.
- IEEE. *SE VOCAB - Software and Systems Engineering Vocabulary*. 2016. Acessado em 18 de setembro de 2016. Disponível em: <https://pascal.computer.org/sev_display/index.action>. Citado 10 vezes nas páginas 50, 54, 56, 141, 156, 166, 171, 200, 202 e 203.
- IEEE. *IEEE Advancing Technology for Humanity*. 2017. Acessado em 16 de março de 2017. Disponível em: <<https://www.ieee.org/>>. Citado na página 197.

- IHMC. *Cmap*. 2014. Acessado em 19 de setembro de 2015. Disponível em: <<http://cmap.ihmc.us/>>. Citado na página 41.
- ILK, N. et al. Semantic enrichment process: An approach to software component reuse in modernizing enterprise systems. *Information Systems Frontiers*, v. 13, n. 3, p. 359–370, 2010. ISSN 1572-9419. Citado na página 164.
- INPIROM, A.; PROMPOON, N. Diagram change types taxonomy based on analysis and design models in UML. In: *2013 4th IEEE International Conference on Software Engineering and Service Science, ICSESS 2013*. [S.l.]: IEEE, 2013. p. 283–287. Citado na página 273.
- IQBAL, R. et al. An analysis of ontology engineering methodologies: A literature review. *Research Journal of Applied Sciences, Engineering and Technology*, v. 6, n. 16, p. 2993 – 3000, 2013. Citado na página 106.
- IQ INTERNATIONAL. *IQ International Glossary*. 2017. Acessado em 19 de maio de 2017. Disponível em: <<http://www.iqint.org/knowledge/fundamentals/glossary/>>. Citado na página 183.
- ISAAC, A.; SUMMERS, E. *SKOS Simple Knowledge Organization System Primer*. 2009. Acessado em 4 de maio de 2015. Disponível em: <<http://www.w3.org/TR/skos-primer/>>. Citado 2 vezes nas páginas 60 e 116.
- ISACA. *COBIT 5: Process Reference Guide Exposure Draft*. 2011. Citado na página 265.
- ISO. *ISO 214:1976 Documentation - Abstracts for publications and documentation*. Geneva, Switzerland, 1976. Citado na página 118.
- ISO. *ISO 5963:1985 Documentation – Methods for examining documents, determining their subjects, and selecting indexing terms*. Switzerland, 1985. Citado na página 118.
- ISO. *ISO 1087-1:2000 Terminology work - Vocabulary - Part 1: Theory and application*. Switzerland, 2000. Citado 5 vezes nas páginas 60, 85, 129, 179 e 183.
- ISO. *ISO - 5127: Information and documentation - Vocabulary*. Geneva, Switzerland, 2001. Citado na página 63.
- ISO. *ISO/IEC 12207:2008: Systems and software engineering - Software life cycle processes*. Geneva, Switzerland, 2008. Citado 10 vezes nas páginas 32, 54, 155, 156, 157, 163, 171, 260, 264 e 266.
- ISO. *ISO/IEC 15288 IEEE Std 15288-2008 Systems and software engineering — System life cycle processes*. 2008. Citado 5 vezes nas páginas 260, 264, 266, 267 e 269.
- ISO. *ISO 23081-1:2009 Information and documentation - Records management Processes - Metadata for records - Part 1: Principles*; 2009. Citado 3 vezes nas páginas 121, 125 e 206.
- ISO. *ISO 23081-2:2009 Information and documentation - Managing metadata for records - Part 2: Conceptual and implementation issues*. 2009. Citado na página 206.
- ISO. *ISO 704:2009 Terminology work - Principles and methods*. Switzerland, 2009. Citado na página 129.

- ISO. *ISO/IEC/IEEE 24765 Systems and Software Engineering Vocabulary*. Switzerland, 2010. Citado 27 vezes nas páginas 33, 38, 54, 137, 142, 154, 155, 156, 158, 159, 166, 170, 171, 179, 181, 183, 185, 198, 202, 203, 260, 261, 262, 264, 265, 266 e 268.
- ISO. *ISO 10241-1:2011 Terminological entries in standards – Part 1: General requirements and examples of presentation*. Switzerland, 2011. Citado na página 129.
- ISO. *ISO 25964-1:2011 Thesauri and interoperability with other vocabularies – Part 1: Thesauri for information retrieval*. Switzerland, 2011. Citado 2 vezes nas páginas 118 e 140.
- ISO. *ISO 25964-2:2011 Thesauri and interoperability with other vocabularies – Part 2: Interoperability with other vocabularies*. Switzerland, 2011. Citado na página 118.
- ISO. *ISO/IEC/IEEE 29148 (E) Systems and software engineering — Life cycle processes — Requirements engineering*. 2011. Citado na página 267.
- ISO. *ISO/IEC/IEEE 42010 Systems and software engineering – Architecture description*. Switzerland, 2011. Citado 6 vezes nas páginas 76, 77, 132, 176, 259 e 269.
- ISO. *ISO 10241-2:2012 Terminological entries in standards – Part 2: Adoption of standardized terminological entries*. Switzerland, 2012. Citado na página 129.
- ISO. *ISO standards What's the bottom line?* 2012. Acessado em 5 de março de 2015. Disponível em: <http://www.iso.org/iso/bottom_line.pdf>. Citado na página 117.
- ISO. *Benefits of International Standards*. 2015. Acessado em 29 de março de 2016. Disponível em: <<http://www.iso.org/iso/home/standards/benefitsofstandards.htm>>. Citado na página 117.
- ISO. *ISO/IEC 2382:2015 Information technology - Vocabulary*. Switzerland, 2015. Citado na página 129.
- ISO. *ISO/IEC/IEEE 15289:2017(E) - Systems and software engineering - Content of life-cycle information products (documentation)*. Switzerland, 2017. Citado 3 vezes nas páginas 48, 162 e 163.
- ISTQB. *ISTQB Glossary*. 2017. Acessado em 19 de maio de 2017. Disponível em: <<http://glossary.istqb.org/>>. Citado na página 183.
- IVANOV, K. *Quality control of information*. Tese (Doutorado) — Royal Institute of Technology - University of Stockholm, 1972. Citado 2 vezes nas páginas 138 e 139.
- JABREF. *JabRef reference manager*. 2015. Acessado em 19 de setembro de 2015. Disponível em: <<http://jabref.sourceforge.net/>>. Citado na página 41.
- JACKA, J.; KELLER, P. *Business Process Mapping: Improving Customer Satisfaction*. 2nd. ed. Hoboken, N.J.: Wiley, 2009. 336 p. Citado na página 35.
- JACOB, E. Classification and categorization: A difference that makes a difference. *Library Trends*, v. 52, n. 3, p. 515–540, 2004. Citado 7 vezes nas páginas 50, 54, 58, 62, 65, 90 e 260.

- JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. *The Unified Software Development Process*. Reading, Mass.: Addison Wesley, 1999. 463 p. Citado 3 vezes nas páginas 157, 160 e 162.
- JAKUS, G. et al. *Concepts, Ontologies, and Knowledge Representation*. [S.l.]: Springer, 2013. (SpringerBriefs in Computer Science). Citado 4 vezes nas páginas 92, 96, 112 e 269.
- JANNUZZI, C. A.; MONTALLI, K. M. L. Informação tecnológica e para negócios no brasil: Introdução a uma discussão conceitual. *Ciência da Informação*, v. 28, n. 1, p. 28–36, 1999. Citado 2 vezes nas páginas 46 e 263.
- JATAIN, A. *Comparison of Domain Analysis Methods and their supporting Quality Attributes*. Dissertação (Mestrado) — Thapar University, 2009. Citado na página 135.
- JATAIN, A.; GOEL, S. Comparison of domain analysis methods in software reuse. *International Journal of Information Technology and Knowledge Management*, v. 2, n. 2, p. 347–352, 2009. Citado 2 vezes nas páginas 134 e 135.
- JIMENO-YEPES, A. et al. Reuse of terminological resources for efficient ontological engineering in life sciences. *BMC Bioinformatics*, v. 10, n. S-10, 2009. Citado 3 vezes nas páginas 34, 100 e 112.
- JOHNSTON, P. *Expressing Dublin Core metadata using the DC-Text format*. Acessado em 28 de abril de 2015., 2007. Disponível em: <<http://dublincore.org/documents/2007/12/03/dc-text/>>. Citado na página 130.
- JOHNSTON, P. *Expressing Dublin Core metadata using HTML/XHTML meta and link elements*. Acessado em 28 de abril de 2015., 2008. Disponível em: <<http://dublincore.org/documents/dc-html/>>. Citado na página 130.
- JOHNSTON, P.; POWELL, A. *Expressing Dublin Core Description Sets using XML (DC-DS-XML)*. Acessado em 28 de abril de 2015., 2008. Disponível em: <<http://dublincore.org/documents/dc-ds-xml/>>. Citado na página 130.
- JOHSON, J.; BLAIS, C. Ontology-based solutions for software reuse. In: *Proceedings of the 6th Annual Aquisition Research Symposium of the Naval Postgraduate School*. [S.l.: s.n.], 2009. Citado na página 33.
- JONES, D.; BENCH-CAPONAND, T.; VISSER, P. Methodologies for ontology development. In: *Proceedings of the IT and KNOWS Conference of the 15th FIP World Computer Congress*. [S.l.: s.n.], 1998. Citado na página 106.
- JONES, S. Some thoughts on classification for retrieval. *Journal of Documentation*, v. 61, n. 5, p. 571 – 581, 2005. Citado na página 65.
- KANG, D. et al. A complexity measure for ontology based on uml. In: *Proceedings of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems*. Washington, DC, USA: IEEE Computer Society, 2004. (FTDCS '04), p. 222–228. Citado 4 vezes nas páginas 142, 152, 183 e 186.
- KANG, K. et al. *Feature-Oriented Domain Analysis (FODA) Feasibility Study*. Pittsburgh, Pennsylvania, USA, 1990. Citado 6 vezes nas páginas 132, 134, 135, 197, 259 e 264.

- KANG, W.; LIANG, Y. A security ontology with mda for software development. In: *2013 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*. [S.l.]: IEEE, 2013. p. 67–74. Citado na página 276.
- KAYED, A. et al. Towards an ontology for software product quality attributes. In: *Proceedings of the 2009 4th International Conference on Internet and Web Applications and Services*. [S.l.: s.n.], 2009. p. 200–204. Citado na página 280.
- KHONDOKER, M.; MUELLER, P. Comparing ontology development tools based on an online survey. In: *Proceedings of the World Congress on Engineering WCE2010*. [S.l.: s.n.], 2010. v. 1. Citado na página 111.
- KIRYAKOV, A. et al. Semantic annotation, indexing, and retrieval. *Web Semantics: Science, Services and Agents on the World Wide Web*, v. 2, n. 1, p. 49 – 79, 2004. ISSN 1570-8268. Citado 2 vezes nas páginas 128 e 259.
- KISHORE, R.; ZHANG, H.; RAMESH, R. A helix-spindle model for ontological engineering. *Communications of the ACM*, ACM, New York, NY, USA, v. 47, n. 2, p. 69–75, 2004. ISSN 0001-0782. Citado na página 276.
- KLESS, D.; MILTON, S. Towards quality measures for evaluating thesauri. In: _____. *Metadata and Semantic Research: 4th International Conference, MTSR 2010, Alcalá de Henares, Spain, October 20-22, 2010. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 312–319. Citado 3 vezes nas páginas 146, 152 e 186.
- KLESS, D. et al. Thesaurus and ontology structure: Formal and pragmatic differences and similarities. *Journal of the Association for Information Science and Technology*, v. 66, n. 7, p. 1348–1366, 2015. Citado 4 vezes nas páginas 82, 88, 100 e 104.
- KOSKINEN, J.; SALMINEN, A.; PAAKKI, J. Hypertext support for the information needs of software maintainers. *J. Softw. Maint. Evol.: Res. Pract.*, v. 16, n. 3, p. 187–215, 2004. Citado na página 155.
- KOWALSKI, G.; MAYBURY, M. *Information Storage and Retrieval Systems*. 2nd. ed. Norwell, MA, USA: Kluwer Academic Publishers, 2002. 318 p. Citado 7 vezes nas páginas 48, 54, 55, 57, 74, 82 e 117.
- KRUCHTEN, P. *The Rational Unified Process: An Introduction*. 3rd. ed. Boston, MA, USA: Addison-Wesley Professional, 2003. 310 p. Citado na página 160.
- KUNDER, M. *The size of the World Wide Web (The Internet)*. 2017. Acessado em 1 de agosto de 2017. Disponível em: <<http://www.worldwidewebsite.com/>>. Citado na página 23.
- KWASNIK, B. The role of classification structures in reflecting and building theory. In: *Proceedings of the 3rd ASIS SIG/CR Classification Research Workshop*. [S.l.: s.n.], 1992. p. 63–82. Citado 3 vezes nas páginas 65, 89 e 90.
- KWASNIK, B. The role of classification in knowledge representation and discovery. *Library Trends*, v. 48, n. 1, p. 22–47, 1999. Citado 5 vezes nas páginas 66, 70, 71, 90 e 187.
- LACASTA, J. et al. An automatic method for reporting the quality of thesauri. *Data & Knowledge Engineering*, v. 104, p. 1–14, 2016. Citado na página 152.

- LACERDA, F.; LIMA-MARQUES, M. Reframing information architecture. In: _____. [S.l.]: Springer International Publishing, 2014. cap. Information Architecture as a Discipline - A Methodological Approach, p. 1–10. Citado na página 79.
- LANCASTER, F. *Vocabulary Control for Information Retrieval*. 2nd. ed. Arlington, Va.: Information Resource Press, 1986. 270 p. Citado 12 vezes nas páginas 63, 81, 82, 84, 87, 88, 93, 95, 101, 118, 267 e 269.
- LANCASTER, F. *Indexação e resumos: teoria e prática*. Segunda edição. [S.l.]: Briquet de Lemos, 2004. Citado 5 vezes nas páginas 74, 81, 84, 88 e 269.
- LANKHORST, M. *Enterprise Architecture at Work: Modeling, Communication and Analysis*. 3rd. ed. New York, NY, USA: Springer, 2012. 364 p. (The Enterprise Engineering Series). Citado na página 176.
- LANTOW, B. Ontometrics: Putting metrics into use for ontology evaluation. In: FRED, A. L. N. et al. (Ed.). *KEOD*. [S.l.]: SciTePress, 2016. p. 186–191. Citado 2 vezes nas páginas 147 e 152.
- LANTOW, B.; SANDKUHL, K. An analysis of applicability using quality metrics for ontologies on ontology design patterns. *Intelligent Systems in Accounting, Finance and Management*, v. 22, n. 1, p. 81–99, 2015. ISSN 1099-1174. Citado na página 152.
- LAPLANTE, P. *What every engineer should know about software engineering*. Boca Raton, FL, USA: CRC Press, 2007. Citado na página 161.
- LASSILA, O. *Introduction to RDF Metadata*. 1997. Acessado em 28 de abril de 2015. Disponível em: <<http://www.w3.org/TR/NOTE-rdf-simple-intro>>. Citado 2 vezes nas páginas 31 e 113.
- LAU, K.; RANA, T. A taxonomy of software composition mechanisms. In: *Proceedings - 36th EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA 2010*. Lille, France: [s.n.], 2010. p. 102–110. Citado na página 272.
- LEE, Y. et al. Aimq: A methodology for information quality assessment. *Inf. Manage.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 40, n. 2, p. 133–146, dez. 2002. ISSN 0378-7206. Citado na página 144.
- LEHNERT, S.; FAROOQ, Q.; RIEBISCH, M. A taxonomy of change types and its application in software evolution. In: *Engineering of Computer Based Systems (ECBS), 2012 IEEE 19th International Conference and Workshops on*. [S.l.]: IEE, 2012. p. 98 – 107. Citado na página 273.
- LEISE, F.; FAST, K.; STECKEL, M. *What Is A Controlled Vocabulary?* 2002. Boxesandarrows. Acessado em 10 de junho de 2015. Disponível em: <<http://boxesandarrows.com/what-is-a-controlled-vocabulary/>>. Citado na página 85.
- LEITE, A.; GIRARDI, R.; CAVALCANTE, U. An ontology for multi-agent domain and application engineering. In: *IEEE International Conference on Information Reuse and Integration, 2008. IRI 2008*. [S.l.: s.n.], 2008. p. 98–103. Citado na página 277.

- LETHBRIDGE, T.; SINGER, J.; FORWARD, A. How software engineers use documentation: The state of the practice. *IEEE Software*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 20, n. 6, p. 35–39, 2003. ISSN 0740-7459. Citado na página 156.
- LEWANDOWSKI, D. Web searching, search engines and information retrieval. *Information Services & Use*, v. 18, n. 3, 2005. Citado na página 55.
- LI, Y.; BELKIN, N. A faceted approach to conceptualizing tasks in information seeking. *Information Processing and Management*, v. 44, p. 1822–1837, 2008. Citado na página 71.
- LIAO, L.; LEUNG, H. Studies on the semantic web, volume 17 : Semantic web enabled software engineering. In: _____. [S.l.]: IOS Press, 2014. cap. A Software Process Ontology and its Application. Citado na página 279.
- LIBRARY OF CONGRESS. *Fascinating Facts*. 2017. Acessado em 1 de agosto de 2017. Disponível em: <<https://www.loc.gov/about/fascinating-facts/>>. Citado na página 23.
- LIMA-MARQUES, M. Outline of a theoretical framework of architecture of information: a school of brasilia proposal. In: BEZIAU J.-Y. & CONIGLIO, M. E. (Ed.). *Logic without Frontiers: Festschrift for Walter Carnielli on the occasion of his 60th Birthday*. London: College Publications, 2011, (Tribute Series, v. 17). Citado 3 vezes nas páginas 28, 79 e 80.
- LIMA-MARQUES, M. *Arquitetura da Informação: um novo olhar para a Informação organizacional*. 2014. Acessado em 23 de setembro de 2016. Disponível em: <<http://www.cbesp.com.br/images/material/8hm7mn70l4mz.pdf>>. Citado na página 80.
- LINDLAND, O.; SINDRE, G.; SØLVBERG, A. Understanding quality in conceptual modeling. *IEEE Software*, v. 11, p. 42–49, 1994. Citado 2 vezes nas páginas 139 e 150.
- LÓPEZ, C. et al. Bridging the gap between software architecture rationale formalisms and actual architecture documents: an ontology-driven approach. *Science of Computer Programming*, v. 77, n. 1, p. 66–80, 2012. Citado na página 277.
- LOSEE, R. Thesaurus structure, descriptive parameters, and scale. *Journal of the Association for Information Science and Technology*, 2015. Citado na página 83.
- LOUREIRO, M. Information organization and visualization in cyberspace: interdisciplinary study based in conceptual maps. In: *Interdisciplinarity and transdisciplinarity in the organization of scientific knowledge : Actas del VIII Congreso ISKO - España*. [S.l.]: Universidad de León, Servicio de Publicaciones, 2007. p. 215–224. Citado na página 75.
- LOZANO-TELLO, A.; GÓMEZ-PÉREZ, A. Ontometric: A method to choose the appropriate ontology. *Journal of Database Management*, v. 15, n. 2, p. 1–18, 2004. Citado 3 vezes nas páginas 112, 141 e 184.
- MA, L.; WANG, H.; LI, Y. A reference model of grouped-metadata object and a change model based on it applying for component-based software integration testing. In: *IEEE/ACS International Conference on Computer Systems and Applications, 2007*. [S.l.]: IEEE, 2007. p. 32–39. Citado na página 170.
- MAAREK, Y. Modern information retrieval. In: _____. 2nd. ed. New York, NY, USA: Addison Wesley, 2011. cap. Web Retrieval, p. 449–517. Citado 3 vezes nas páginas 52, 55 e 62.

- MADER, C.; HASLHOFER, B. Perception and relevance of quality issues in web vocabularies. In: *Proceedings of the 9th International Conference on Semantic Systems*. New York, NY, USA: ACM, 2013. (I-SEMANTICS '13), p. 9–16. ISBN 978-1-4503-1972-0. Citado na página 82.
- MAI, J.-E. Semiotics and indexing: An analysis of the subject indexing process. *Journal of Documentation*, v. 57, n. 5, p. 591–622, 2001. Citado na página 74.
- MAI, J.-E. Classification in context: Relativity, reality, and representation. *Knowledge Organization*, v. 31, n. 1, p. 39–48, 2004. Citado na página 90.
- MAI, J.-E. *Contextual Analysis for the Design of Controlled Vocabularies*. 2006. Bulletin of the American Society for Information Science and Technology. Acessado em 16 de maio de 2015. Disponível em: <<https://www.asis.org/Bulletin/Oct-06/mai.html>>. Citado 3 vezes nas páginas 101, 102 e 103.
- MAI, J.-E. Actors, domains, and constraints in the design and construction of controlled vocabularies. *Knowledge Organization*, v. 35, n. 1, p. 16–21, 2008. Citado 4 vezes nas páginas 81, 101, 102 e 269.
- MALA, M.; ÇIL, I. A taxonomy for measuring complexity in agent-based systems. In: *2011 IEEE 2nd International Conference on Software Engineering and Service Science, ICSESS 2011*. Beijing, China: IEEE, 2011. p. 851–854. Citado na página 275.
- MALLOY, M. et al. *An Information Architecture Framework for the USAF: Managing Information from an Enterprise Architecture*. [S.l.], 2010. Citado na página 176.
- MANGOLD, C. A survey and classification of semantic search approaches. *Int. J. Metadata, Semantics and Ontology*, v. 12, n. 1, 2007. Citado 2 vezes nas páginas 54 e 88.
- MANOJ, S. A comparative study of software quality models. *International Journal of Computer Science and Information Technologies*, v. 5, n. 4, p. 5634–5638, 2014. Citado na página 147.
- MARCHIONINI, G. Exploratory search: from finding to understanding. *Communications of the ACM*, v. 49, n. 4, p. 41–46, 2006. Citado na página 50.
- MARCONDES, C.; SAYÃO, L. Integração e interoperabilidade no acesso a recursos informacionais eletrônicos em c&t: a proposta da biblioteca digital brasileira. *Ciência*, v. 30, n. 3, p. 24–33, 2001. Citado 2 vezes nas páginas 31 e 121.
- MARCUS, A.; MENZIES, T. Software is data too. In: *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research*. New York, NY, USA: ACM, 2010. (FoSER '10), p. 229–232. ISBN 978-1-4503-0427-6. Citado na página 32.
- MARGOLIS, E.; LAURENCE, S. Concepts. In: ZALTA, E. N. (Ed.). *The Stanford Encyclopedia of Philosophy*. Spring 2014. Metaphysics Research Lab, Stanford University, 2014. Disponível em: <<https://plato.stanford.edu/archives/spr2014/entries/concepts/>>. Citado na página 60.
- MARQUES, A. et al. An ontology for task allocation to teams in distributed software development. In: *IEEE 8th International Conference on Global Software Engineering, ICGSE 2013*. Bari, Italy: IEEE, 2013. p. 21–30. Citado na página 278.

- MARTÍNEZ, A. et al. La estructura sistemática del tesauro: indicadores para evaluar su calidad. *Revista Española de Documentación Científica*, v. 34, n. 1, p. 29–43, 2011. Citado 4 vezes nas páginas 104, 141, 147 e 186.
- MARTINS, A. et al. Suggesting software components for reuse in search engines using discovered knowledge techniques. In: *Software Engineering and Advanced Applications, 2009. SEAA '09. 35th Euromicro Conference on*. [S.l.: s.n.], 2009. p. 412–419. ISSN 1089-6503. Citado na página 164.
- MATHES, A. *Folksonomies: Cooperative Classification and Communication Through Shared Metadata*. [S.l.], 2004. Acessado em: 20 de fevereiro de 2015. Disponível em: <<http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html>>. Citado na página 73.
- MATHEUS, R. Rafael capurro e a filosofia da informação: abordagens, conceitos e metodologias de pesquisa para a ciência da informação. *Perspectivas em Ciência da Informação*, v. 10, n. 2, p. 140–165, 2005. Citado na página 46.
- MCGEE, S.; GREER, D. A software requirements change source taxonomy. In: *4th International Conference on Software Engineering Advances, ICSEA 2009*. Porto, Portugal: [s.n.], 2009. p. 51–58. Citado na página 271.
- MCGEE, S.; GREER, D. Software requirements change taxonomy: Evaluation by case study. In: *2011 19th IEEE International Requirements Engineering Conference (RE)*. [S.l.]: IEEE, 2011. p. 25 – 34. Citado na página 271.
- MCMILLAN, C. et al. Exemplar: A source code search engine for finding highly relevant applications. *IEEE Transactions on Software Engineering*, v. 38, n. 5, p. 1069–1087, 2012. Citado na página 164.
- MEHTA, N.; MEDVIDOVIC, N.; PHADKE, S. Towards a taxonomy of software connectors. In: *Proceeding ICSE '00 Proceedings of the 22nd international conference on Software*. [S.l.: s.n.], 2000. p. 178–187. Citado na página 272.
- MEIER, R.; CAHILL, V. Taxonomy of distributed event-based programming systems. *Computer Journal*, v. 48, n. 5, p. 602–626, 2005. Citado na página 275.
- MENDES, O.; ABRAN, A. Software engineering ontology: A development methodology. *Metrics News*, v. 9, p. 68–76, 2004. Citado na página 168.
- MICHEL, D. Final report to the alcts/ccs subject analysis committee. In: _____. ALCTS, 1997. cap. Appendix B: Taxonomy of Subject Relationships, p. 18. Acessado em: 28 de abril de 2016. Disponível em: <<http://www.ala.org/alcts/sites/ala.org.alcts/files/content/mgrps/camms/cmtes/sac/inact/subjrel/msrscu2.pdf>>. Citado na página 88.
- MILES, A.; BECHHOFFER, S. *SKOS Simple Knowledge Organization System Reference*. 2009. Acessado em 4 de maio de 2015. Disponível em: <<http://www.w3.org/TR/skos-reference/>>. Citado 2 vezes nas páginas 99 e 116.
- MILI, H. et al. Another nail to the coffin of faceted controlled-vocabulary component classification and retrieval. *SIGSOFT Softw. Eng. Notes*, ACM, New York, NY, USA, v. 22, n. 3, p. 89–98, maio 1997. ISSN 0163-5948. Citado na página 167.

- MILLER, G. The characteristics of agile software processes. In: *Proceedings of the 39th Int'l Conf. and Exhibition on Technology of Object-Oriented Languages and Systems (TOOLS'01)*. [S.l.]: IEEE, 2001. Citado na página [161](#).
- MILLER, H. The multiple dimensions of information quality. *Information Systems Manquality*, v. 13, n. 2, p. 79–82, 1996. Citado 2 vezes nas páginas [138](#) e [144](#).
- MILLER, S. *Metadata for Digital Collections*. New York, NY, USA: Neal-Schuman Publishers, Inc., 2011. 230 p. Citado 20 vezes nas páginas [47](#), [48](#), [49](#), [81](#), [89](#), [91](#), [117](#), [119](#), [120](#), [121](#), [122](#), [123](#), [126](#), [127](#), [129](#), [163](#), [206](#), [263](#), [267](#) e [269](#).
- MILLER, U. Thesaurus construction: Problems and their roots. *Information Processing & Management*, v. 33, n. 4, p. 481–493, 1997. Citado na página [93](#).
- MIRBEL, I. Oflossc, an ontology for supporting open source development communities. In: *ICEIS (4)*. [S.l.: s.n.], 2009. p. 47–52. Citado na página [279](#).
- MOCKUS, A.; FIELDING, R.; HERBSLEB, J. Two case studies of open source software development: Apache and mozilla. *ACM Transactions on Software Engineering and Methodology*, v. 11, n. 3, p. 309–346, 2002. Citado na página [162](#).
- MÖLLER, K. Lifecycle models of data-centric systems and domains: The abstract data lifecycle model. *Semantic Web - Linked Data for science and education*, IOS Press, Amsterdam, The Netherlands, The Netherlands, v. 4, n. 1, p. 67–88, January 2013. ISSN 1570-0844. Citado na página [126](#).
- MONPERRUS, M. et al. What should developers be aware of? an empirical study on the directives of api documentation. *Empirical Software Engineering*, v. 17, n. 6, p. 703–737, 2012. Citado na página [272](#).
- MONTALLI, K. M. L.; CAMPELLO, B. Fontes de informação sobre companhias e produtos industriais: uma revisão de literatura. *Ciência da Informação*, v. 26, n. 3, p. 321–326, 1997. Citado na página [47](#).
- MOODY, D. Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data & Knowledge Engineering*, v. 55, p. 243–276, 2005. Citado 6 vezes nas páginas [40](#), [133](#), [139](#), [140](#), [150](#) e [170](#).
- MOODY, G. *Rebel code: Linux and the open source revolution*. New York: Allen Lane, 2001. Citado na página [162](#).
- MOOERS, C. Zatacoding applied to mechanical organization of knowledge. *American Documentation*, v. 2, p. 20–32, 1951. Citado 3 vezes nas páginas [23](#), [50](#) e [266](#).
- MORATO, J.; SANCHEZ-CUADRADO, S.; DIMOU, C. Evaluation of semantic retrieval systems on the semantic web. *Library Hi Tech*, v. 31, n. 4, p. 638–656, 2013. Citado na página [57](#).
- MORO, R.; FALBO, R. Uma ontologia para o domínio de qualidade de software com foco em produtos e processos de software. In: *Third Workshop on Ontologies and Metamodeling in Software and Data Engineering - WOMSDE 2008*. [S.l.: s.n.], 2008. p. 37–48. Citado na página [280](#).

- MORVILLE, P.; ROSENFELD, L. *Information Architecture for the World Wide Web: Designing Large-Scale Web Sites*. 3rd. ed. Sebastopol, CA: O'Reilly, 2006. 528 p. Citado 9 vezes nas páginas 28, 58, 59, 78, 88, 91, 93, 94 e 193.
- MUKHTAR, N. et al. Ontology for feature based selection of web development tools. In: *Eighth International Conference on Digital Information Management (ICDIM)*. [S.l.]: IEEE, 2013. p. 90–95. Citado na página 277.
- MUSTAFA, J.; KHAN, S.; LATIF, K. Ontology based semantic information retrieval. In: *2008 4th International IEEE Conference "Intelligent Systems"*. [S.l.: s.n.], 2008. p. 22–14 – 22–19. Citado 2 vezes nas páginas 52 e 53.
- MYLOPOULOS, J. Conceptual modeling, databases, and case : An integrated view of information systems development. In: _____. New York, NY, USA: [s.n.], 1992. cap. Conceptual modeling and Telos. Citado na página 133.
- NAKAKOJI, K. et al. Evolution patterns of open-source software systems and communities. In: *International Workshop on Principles of Software Evolution 2002*. Orlando, FL, USA: [s.n.], 2002. Citado na página 162.
- NARDI, J.; FALBO, R. Uma ontologia de requisitos de software. In: *9th Workshop Iberoamericano de Ingenieria de Requisitos y Ambientes de Software*. La Plata, Argentina: [s.n.], 2006. p. 111–124. Citado na página 276.
- NAUMANN, F.; ROLKER, C. Assessment methods for information quality criteria. In: *Proceedings of 5th International Conference on Information Quality*. [S.l.: s.n.], 2000. p. 148–162. Citado na página 145.
- NAVARRO, J.; GARCÍA-PEÑALVO, F.; THERÓN, R. A survey on ontology metrics. In: LYTRAS, M. D. et al. (Ed.). *WSKS (1)*. [S.l.]: Springer, 2010. (Communications in Computer and Information Science, v. 111), p. 22–27. Citado na página 186.
- NEUHAUS, F. et al. Towards ontology evaluation across the life cycle: The ontology summit 2013. *Applied Ontology*, IOS Press, Amsterdam, The Netherlands, The Netherlands, v. 8, n. 3, p. 179–194, 2013. ISSN 1570-5838. Citado 7 vezes nas páginas 104, 105, 106, 141, 142, 146 e 184.
- NIDHI, A.; JATAIN, A. Component retrieval techniques - a systematic review. *International Journal of Scientific & Engineering Research*, v. 5, n. 1, p. 1699–1706, 2014. Citado na página 165.
- NIE, L.; ZHONG, L. Component retrieval based on domain ontology and user interest. In: *E-Business and Information System Security, 2009. EBISS '09. International Conference on*. [S.l.: s.n.], 2009. p. 1–4. Citado na página 164.
- NILSSON, M. *Description Set Profiles: A constraint language for Dublin Core Application Profiles*. 2008. Acessado em 25 de abril de 2015. Disponível em: <<http://dublincore.org/documents/2008/03/31/dc-dsp/>>. Citado na página 125.
- NILSSON, M. *The Singapore Framework for Dublin Core Application Profiles*. 2008. Acessado em: 25 de abril de 2015. Disponível em: <<http://dublincore.org/documents/singapore-framework/>>. Citado 2 vezes nas páginas 125 e 130.

- NILSSON, M. et al. *Expressing Dublin Core metadata using the Resource Description Framework (RDF)*. Acessado em 28 de abril de 2015., 2008. Disponível em: <<http://dublincore.org/documents/dc-rdf/>>. Citado na página 130.
- NISO. *Understanding Metadata*. Bethesda, MD, USA, 2004. Citado 7 vezes nas páginas 120, 121, 122, 123, 124, 263 e 264.
- NISO. *ANSI/NISO Z39.19-2005 (R2010) Guidelines for the Construction, Format, and management of Monolingual Controlled Vocabularies*. Bethesda, MD, USA, 2005. Citado 31 vezes nas páginas 23, 47, 48, 54, 56, 63, 64, 67, 74, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 92, 101, 104, 118, 124, 140, 141, 184, 187, 262, 268 e 269.
- NISO. *A Framework of Guidance for Building Good Digital Collections*. 2007. Citado 6 vezes nas páginas 31, 48, 49, 120, 123 e 263.
- NOVAK, J.; KRAJNC, A.; ZONTAR, R. Taxonomy of static code analysis tools. In: *33rd International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2010*. Opatija, Croatia: [s.n.], 2010. p. 418–422. Citado na página 272.
- NOVELLINO, M. Instrumentos e metodologias de representação da informação. *Informação & Informação*, v. 1, n. 2, p. 37–45, 1996. Citado 2 vezes nas páginas 62 e 267.
- NOY, N.; MCGUINNESS, D. *Ontology Development 101: A Guide to Creating Your First Ontology*. [S.l.], 2001. Acessado em: 31 de agosto de 2016. Disponível em: <http://protege.stanford.edu/publications/ontology_development/ontology101.pdf>. Citado 3 vezes nas páginas 97, 107 e 187.
- NUNBERG, G. The future of the book. In: _____. Berkeley, CA, USA: University of California Press, 1996. cap. Farewell to the Information Age, p. 103–138. Citado na página 46.
- NUNES, B. *Integrando Gerência de Configuração de Software, Documentação e Gerência de Conhecimento em um Ambiente de Desenvolvimento de Software*. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, 2005. Citado na página 280.
- NURMULIANI, N.; ZOWGHI, D.; FOWELL, S. Analysis of requirements volatility during software development life cycle. In: *Proceedings - 2004 Australian Software Engineering Conference ASWEC 2004*. Melbourne, Vic., Australia: IEEE, 2004. v. 2004, p. 28–37. Citado na página 273.
- OBJECT MANAGEMENT GROUP. *Unified Modeling Language*. 2017. Acessado em 31 de agosto de 2017. Disponível em: <<http://www.uml.org/>>. Citado na página 40.
- OGC. *Managing Successful Projects with PRINCE2: 2009 Edition*. [S.l.]: Stationery Office Books, 2009. 342 p. Citado na página 265.
- OGDEN, C.; RICHARDS, I. *The Meaning of Meaning: A study of the influence of language upon thought and of the science of symbolism*. [S.l.]: Harcourt, Brace & company, 1923. Citado 4 vezes nas páginas 53, 63, 64 e 268.
- OH, S.; YEOM, H. A comprehensive framework for the evaluation of ontology modularization. *Expert Systems with Applications*, v. 39, n. 10, p. 8547–8556, 2012. Citado 2 vezes nas páginas 152 e 183.

- OMG. *Reusable Asset Specification OMG Available Specification Version 2.2*. 2005. Citado na página 170.
- ONTOLOGY ENGINEERING GROUP. *The NeOn Methodology*. 2015. Acessado em: 17 de abril de 2015. Disponível em: <<http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/methodologies/59-neon-methodology>>. Citado 2 vezes nas páginas 109 e 187.
- ONTOTEXT. *Semantic Annotation*. 2015. Acessado em 12 de março de 2016. Disponível em: <<http://ontotext.com/products/ontotext-semantic-platform/semantic-annotation/>>. Citado na página 128.
- OPEN HUB. *Open Hub*. 2017. Acessado em 23 de fevereiro de 2017. Disponível em: <<https://www.openhub.net/>>. Citado 2 vezes nas páginas 198 e 199.
- OPENSTACK. *Infrastructure User Manual*. 2015. Acessado em 8 de maio de 2015. Disponível em: <<http://docs.openstack.org/infra/manual/index.html>>. Citado na página 162.
- OREN, E. et al. *What are Semantic Annotations?* 2006. Acessado em 17 de setembro de 2016. Disponível em: <<http://www.siegfried-handschuh.net/pub/2006/whatissemannot2006.pdf>>. Citado 3 vezes nas páginas 117, 128 e 259.
- ORG. *Various Meanings of the term 'Ontology'*. 2007. Acessado em: 6 de março de 2015. Disponível em: <<http://org.buffalo.edu/OntologyDefs1.html>>. Citado 2 vezes nas páginas 95 e 265.
- ORME, A.; TAO, H.; ETZKORN, L. Coupling metrics for ontology-based system. *IEEE Software*, v. 23, n. 2, p. 102–108, 2006. Citado 2 vezes nas páginas 184 e 186.
- ORR, R. Measuring the goodness of library services: A general framework for considering quantitative measures. *Journal of Documentation*, v. 29, n. 3, p. 315–332, 1979. Citado na página 143.
- ORSO, A. et al. Using component metadata to regression test component-based software. *Software Testing Verification and Reliability*, v. 17, n. 2, p. 61–94, 2007. Citado na página 170.
- OSTERTAG, E. et al. Computing similarity in a reuse library system: an AI-based approach. *ACM Transactions on Software Engineering and Methodology*, v. 1, n. 2, p. 205–228, 1992. Citado na página 165.
- PEARCE-MOSES, R. *A glossary of archival and records terminology*. Chicago, IL: The Society of American Archivists, 2005. Citado 2 vezes nas páginas 67 e 260.
- PEPPER, M.; DRISCOLL, D. *Writing Definitions*. 2015. Acessado em 31 de maio de 2017. Disponível em: <<https://owl.english.purdue.edu/owl/resource/622/01/>>. Citado na página 40.
- PERRY, D.; STAUDENMEYER, N.; VOTTA, L. People, organizations, and process improvement. *IEEE Software*, v. 11, n. 4, p. 36–45, 1994. Citado na página 33.
- PETERSEN, K. et al. Systematic mapping studies in software engineering. In: *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*. Swinton, UK,: British Computer Society, 2008. (EASE'08), p. 68–77. Citado 3 vezes nas páginas 28, 40 e 201.

- PIATTINI, M. et al. Metrics for software conceptual models. In: _____. [S.l.]: Imperial College Press, 2005. cap. Towards a Framework for Conceptual Modelling Quality. Citado 4 vezes nas páginas 139, 140, 145 e 150.
- PIETO-DIAZ, R. Domain analysis: An introduction. *Software Engineering Notes*, v. 15, n. 2, p. 47–54, 1990. Citado 3 vezes nas páginas 32, 133 e 259.
- PINHEIRO, L.; LOUREIRO, J. Traçados e limites da ciência da informação. *Ciência da Informação*, v. 24, n. 1, 1995. Citado na página 46.
- PINTO, M. A user view of the factors affecting quality of thesauri in social science databases. *Library & Information Science Research*, v. 30, p. 216–221, 2008. Citado na página 146.
- PIZZOLETO, A.; OLIVEIRA, H. A systematic approach to evaluate enterprise ontologies using testing techniques of software usability. In: *Int'l Conf. Software Eng. Research and Practice (SERP'16)*. [S.l.: s.n.], 2016. Citado 2 vezes nas páginas 152 e 183.
- PMI. *A Guide to the Project Management Body of Knowledge (PMBOK GUIDE)*. 5th. ed. [S.l.]: Project Management Institute, 2013. 589 p. Citado 17 vezes nas páginas 23, 35, 36, 38, 39, 53, 137, 140, 143, 155, 176, 180, 194, 261, 264, 265 e 266.
- POMERANTZ, J. *Metadata*. Cambridge, MA, USA: The MIT Press, 2015. 256 p. Citado 8 vezes nas páginas 49, 113, 120, 121, 122, 123, 124 e 263.
- PONTES, F.; LIMA, G. A organização do conhecimento em ambientes digitais: aplicação da teoria da classificação facetada. *Perspectivas em Ciência da Informação*, v. 17, n. 4, p. 18–40, 2012. Citado na página 58.
- PORTER, M. Towards a dynamic theory of strategy. *Strategic Management Journal*, v. 12, p. 95–117, 1991. Citado na página 24.
- POVEDA-VILLALÓN, M.; GÓMEZ-PÉREZ, A.; SUÁREZ-FIGUEROA, M. Oops! (ontology pitfall scanner!): An on-line tool for ontology evaluation. *International Journal on Semantic Web & Information Systems*, IGI Global, Hershey, PA, USA, v. 10, n. 2, p. 7–34, 2014. Citado na página 142.
- POWELL, A. *Guidelines for implementing Dublin Core in XML*. Acessado em 28 de abril de 2015., 2003. Disponível em: <<http://dublincore.org/documents/dc-xml-guidelines/>>. Citado na página 130.
- POWELL, A. et al. *DCMI Abstract Model*. 2007. Acessado em 25 de abril de 2015. Disponível em: <<http://dublincore.org/documents/abstract-model/>>. Citado 2 vezes nas páginas 130 e 267.
- PRESSMAN, R. *Engenharia de Software: Uma Abordagem Profissional*. [S.l.]: McGraw-Hill, 2011. 776 p. Citado na página 33.
- PRIETO-DIAZ, R. Implementing faceted classification for software reuse. In: *Proceedings 12th International Conference on Software Engineering*. [S.l.: s.n.], 1990. Citado 3 vezes nas páginas 33, 82 e 167.
- PRIETO-DIAZ, R.; FREEMAN, P. Classifying software for reusability. *IEEE Software*, v. 4, n. 1, p. 6–16, Jan 1987. Citado 2 vezes nas páginas 71 e 200.

- PRINCETON UNIVERSITY. *WordNet*. 2010. Acessado em 26 de fevereiro de 2017. Disponível em: <<https://wordnet.princeton.edu/>>. Citado 6 vezes nas páginas 40, 179, 181, 183, 200 e 203.
- PUVIANI, M.; CABRI, G.; ZAMBONELLI, F. A taxonomy of architectural patterns for self-adaptive systems. In: *Proceedings of the International C* Conference on Computer Science and Software Engineering*. New York, NY, USA: ACM, 2013. (C3S2E '13), p. 77–85. Citado na página 272.
- QIN, J. Representation and organization of information in the web space: from marc to xml. *Informing Science*, v. 3, n. 2, p. 83–88, 2000. Citado na página 62.
- RAAD, J.; CRUZ, C. A survey on ontology evaluation methods. In: *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015) - Volume 2: KEOD*. [S.l.: s.n.], 2015. p. 179–186. Citado 3 vezes nas páginas 141, 152 e 183.
- RAHMAN, A.; SAHIBUDDIN, S.; IBRAHIM, S. A unified framework for software engineering process improvement - a taxonomy comparative analysis. In: *2011 5th Malaysian Conference in Software Engineering, MySEC 2011*. Johor Bahru, Malaysia: [s.n.], 2011. p. 153–158. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-84857336280&partnerID=40&md5=ecf5f650ec90c9c85055c1df456bf0ba>>. Citado na página 274.
- RAHMAN, A.; SAHIBUDDIN, S.; IBRAHIM, S. Using taxonomy comparative analysis for the unification of process improvement frameworks. *International Journal of Digital Content Technology and its Applications*, v. 6, n. 21, p. 34–42, 2012. Citado na página 274.
- RAJAPAKSE, D. Techniques for de-fragmenting mobile applications: A taxonomy. In: *20th International Conference on Software Engineering and Knowledge Engineering, SEKE 2008*. CA, USA: [s.n.], 2008. p. 923–928. Citado na página 272.
- RANGANATHAN, S. *Prolegomena to Library Classification*. [S.l.]: Asia Publishing House, 1967. Citado 2 vezes nas páginas 69 e 71.
- RAO, S.; MEDEIROS, H.; KAK, A. An incremental update framework for efficient retrieval from software libraries for bug localization. In: *Proc. WCRE 2013*. [S.l.: s.n.], 2013. p. 62–71. Citado na página 164.
- REDMOND-NEAL, A.; HLAVA, M. *ASIS&T Thesaurus of Information Science, Technology, and Librarianship*. 3rd. ed. [S.l.]: Information Today Inc., 2005. (Asist Monograph Series). Citado 3 vezes nas páginas 93, 94 e 104.
- REIMER, U. Learning a lightweight ontology for semantic retrieval in patient-centered information systems. *International Journal of Knowledge Management*, v. 7, n. 3, p. 11–26, 2011. Citado na página 53.
- REIMER, U. *Lightweight Ontologies (LWO) versus Full-Fledged Ontologies*. 2011. Acessado em 19 de junho de 2015. Disponível em: <<http://www.ai-one.com/2011/05/31/lightweight-ontologies-lwo-versus-full-fledged-ontologies/>>. Citado 4 vezes nas páginas 95, 99, 112 e 265.

- REIS, C. R.; FORTES, R. An overview of the software engineering process and tools in the mozilla project. In: *Proceedings of the Workshop on Open Source Software Development*. [S.l.: s.n.], 2002. p. 155–175. Citado na página 162.
- REITZ, J. *Online Dictionary for Library and Information Science*. [S.l.]: ABC-CLIO, 2012. Citado 5 vezes nas páginas 65, 89, 90, 260 e 268.
- REN, F.; BRACEWELL, D. Advanced information retrieval. *Electronic Notes in Theoretical Computer Science*, v. 225, p. 303–317, 2009. Citado na página 53.
- RESEARCH, S. C. for B. I. *Protégé*. 2015. Acessado em 2 de fevereiro de 2016. Disponível em: <<http://protege.stanford.edu/>>. Citado na página 111.
- RESMINI, A.; ROSATI, L. A brief history of information architecture. *Journal of Information Architecture*, v. 3, n. 2, 2012. Acessado em 17 de setembro de 2016. Disponível em: <<http://journalofia.org/volume3/issue2/03-resmini/jofia-0302-03-resmini.pdf>>. Citado 2 vezes nas páginas 76 e 78.
- RICHARDSON, I. et al. A process framework for global software engineering teams. *Information and Software Technology*, v. 54, p. 1175–1191, 2012. Citado na página 161.
- RIECHERT, T. et al. Towards semantic based requirements engineering. In: *Proc. of the 7th International Conference on Knowledge Management (IKNOW)*. [S.l.: s.n.], 2007. Citado na página 276.
- ROBILLARD, P. et al. Taxonomy for software teamwork measurement. *Journal of Software: Evolution and Process*, v. 26, n. 10, p. 910–922, 2014. ISSN 2047-7481. Citado na página 274.
- ROBLES, K. et al. Towards an ontology-based retrieval of UML class diagrams. *Information and Software Technology*, v. 54, p. 72–86, 2012. Citado 2 vezes nas páginas 32 e 164.
- ROCHA, R. et al. An ontology-based system to support distributed software development. In: *ICSEA 2013 : The Eighth International Conference on Software Engineering Advances*. [S.l.: s.n.], 2013. Citado na página 279.
- RODRIGUEZ, A. et al. A proposal for a semantic intelligent document repository architecture. In: *Electronics, Robotics and Automotive Mechanics Conference, 2009. CERMA '09*. [S.l.: s.n.], 2009. p. 69–75. Citado na página 169.
- ROGGIO, R.; GORDON, J.; COMER, J. Taxonomy of common software testing terminology: Framework for key software engineering testing concepts. In: *2013 Proceedings of the Conference for Information Systems Applied Research*. San Antonio, Texas, USA: [s.n.], 2013. Citado na página 274.
- ROONGKAEW, W.; PROMPOON, N. Software engineering tools classification based on swebok taxonomy and software profile. In: *2013 2nd International Conference on Informatics and Applications, ICIA 2013*. Lodz, Poland: IEEE, 2013. p. 122–128. Citado na página 275.
- ROWLEY, J. What is information ? *Information Services & Use*, v. 18, p. 243–254, 1998. Citado na página 46.

- ROWLEY, J.; HARTLEY, R. *Organizing Knowledge: An Introduction to Managing Access to Information*. 4th. ed. Burlington, VT: Ashgate Publishing Limited, 2008. Citado 18 vezes nas páginas 50, 51, 55, 56, 58, 64, 66, 72, 73, 74, 85, 92, 97, 100, 101, 105, 117 e 187.
- ROZANSKI, N.; WOODS, E. *Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives*. 2nd. ed. [S.l.]: Addison Wesley, 2011. Citado na página 182.
- RÜHLE, S.; BAKER, T.; JOHNSTON, P. *User Guide*. 2011. Acessado em 28 de abril de 2015. Disponível em: <http://wiki.dublincore.org/index.php/User_Guide>. Citado na página 130.
- RUIZ, F.; HILERA, J. Ontologies for software engineering and software technology. In: _____. [S.l.]: Springer, 2006. cap. Using Ontologies in Software Engineering and Technology, p. 49–102. Citado 3 vezes nas páginas 97, 166 e 168.
- RUNGRATRI, S.; USANAVASIN, S. Semantic based approach supporting cmmi gap analysis process. *Journal of Convergence Information Technology*, v. 7, n. 20, p. 127–137, 2012. Citado 2 vezes nas páginas 278 e 279.
- SAA. A glossary of archival and records terminology. Acessado em 28 de janeiro de 2014. 2014. Disponível em: <<http://www2.archivists.org/glossary>>. Citado 2 vezes nas páginas 50 e 266.
- SARACEVIC, T. Interdisciplinary nature of information science. *Ciência da Informação*, v. 24, n. 1, 1995. Citado 2 vezes nas páginas 23 e 25.
- SARMA, A. *A survey of collaborative tools in software development*. Irvine, 2005. Citado na página 33.
- SCACCHI, W. et al. Understanding free/open source software development processes. *Software Process Improvement and Practice*, v. 11, p. 95–105, 2006. Citado na página 162.
- SCHACH, S. *Engenharia de Software: Os Paradigmas Clássico e Orientado a Objetos*. Sétima. [S.l.]: McGraw Hill, 2008. 618 p. Citado na página 33.
- SCHREIBER, G.; RAIMOND, Y. *RDF 1.1 Primer*. 2014. Acessado em 28 de abril de 2015. Disponível em: <<http://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/>>. Citado na página 114.
- SCHULZ, S. et al. *Guideline on Developing Good Ontologies in the Biomedical Domain with Description Logics*. 2012. Citado na página 141.
- SCOTTO, M.; SILLITTI, A.; SUCCI, G. An empirical analysis of the open source development process based on mining of source code repositories. *International Journal of Software Engineering*, v. 17, n. 2, p. 231–247, 2007. Citado na página 162.
- SEAMAN, B. The information gathering strategies of software maintainers. In: *Software Maintenance, 2002. Proceedings. International Conference on*. [S.l.: s.n.], 2002. p. 141–149. ISSN 1063-6773. Citado na página 155.
- SHAN, T.; HUA, W. Taxonomy of java web application frameworks. In: *IEEE International Conference on e-Business Engineering, ICEBE 2006*. Shanghai, China: IEEE, 2006. p. 377–385. Citado na página 272.

- SHANKS, G.; DARKE, P. Quality in conceptual modelling: Linking theory and practice. In: *Proceedings of the Asia-Pacific Conference on Information Systems (PACIS)*. [S.l.: s.n.], 1997. p. 805–814. Citado na página 140.
- SHAW, M. Software engineering education: a roadmap. In: *Proceedings of the 22nd International Conference on Software Engineering: the future of software engineering*. New York, NY: ACM Press, 2000. p. 371–380. Citado na página 32.
- SHERMAN, S. *A Process-oriented Ontology for Representing Software Engineering Project Knowledge*. Tese (Doutorado) — Nova Southeastern University, 2009. Citado na página 278.
- SHIRI, A. *Powering Search: The Role of Thesauri in New Information Environments*. [S.l.]: Information Today, Inc., 2012. 318 p. (ASIST Monograph Series). Citado 4 vezes nas páginas 31, 73, 93 e 94.
- SHUJA, A.; KREBS, J. *IBM Rational Unified Process Reference and Certification Guide: Solution Designer (RUP)*. [S.l.]: IBM Press, 2008. Citado na página 162.
- SICILIA, M. Metadata, semantics and ontologies: providing meaning to information resources. *Int. J. Metadata, Semantics and Ontologies*, v. 1, n. 1, p. 83–86, 2006. Citado 2 vezes nas páginas 64 e 97.
- SICILIA, M.; CUADRADO, J.; RODRIGUEZ, D. Ontologies of software artifacts and activities: Resource annotation and application to learning technologies. In: *SEKE*. [S.l.: s.n.], 2005. p. 145–150. Citado na página 280.
- SICILIA, M. et al. Empirical findings on ontology metrics. *Expert Syst. Appl.*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 39, n. 8, p. 6706–6711, jun. 2012. ISSN 0957-4174. Citado na página 184.
- SILVESTRI, F. et al. Toward a search architecture for software components. *Concurrency Computat.: Pract. Exper.*, v. 18, p. 1317–1331, 2006. Citado na página 164.
- SIMMONS, G.; DILLON, T. Ifip international federation for information processing, open source systems. In: _____. Italy: Springer, 2006. v. 203, cap. Towards an ontology for open source software development, p. 65–75. Citado 4 vezes nas páginas 33, 97, 163 e 278.
- SIMPERL, E.; TEMPICH, C. Ontology engineering: A reality check. In: *Proceedings of the 5th International Conference on Ontologies, Databases, and Applications of Semantics ODBASE2006*. [S.l.: s.n.], 2006. Citado na página 105.
- SINGH, S. Article: An experiment in software component retrieval based on metadata and ontology repository. *International Journal of Computer Applications*, v. 61, n. 14, p. 33–40, January 2013. Citado na página 164.
- SIQUEIRA, A. *A Lógica e a Linguagem como fundamentos da Arquitetura da Informação*. Dissertação (Mestrado) — Universidade de Brasília, 2008. Disponível em: <<http://repositorio.unb.br/handle/10482/3185>>. Citado 2 vezes nas páginas 49 e 261.
- SIQUEIRA, A. *Arquitetura da Informação: Uma proposta para fundamentação e caracterização da disciplina científica*. Tese (Doutorado) — Universidade de Brasília, Faculdade de Ciência da Informação, Brasília, 2012. Disponível em: <<http://repositorio.unb.br/handle/10482/12157>>. Citado 2 vezes nas páginas 49 e 77.

- SMIRAGLIA, R. Cultural frames of knowledge. In: _____. [S.l.]: Ergon Verlag, 2012. cap. Epistemology of domain analysis, p. 111–124. Citado 2 vezes nas páginas 133 e 259.
- SMIRAGLIA, R. *Domain Analysis for Knowledge Organization: Tools for Ontology Extraction 1st Edition*. [S.l.]: Chandos Publishing, 2015. Citado na página 134.
- SOERGEL, D. Thesauri and ontologies in digital libraries: 2. design, evaluation, and development. In: *Proceedings of the 2Nd ACM/IEEE-CS Joint Conference on Digital Libraries*. New York, NY, USA: ACM, 2002. (JCDL '02), p. 419–419. ISBN 1-58113-513-0. Citado na página 141.
- SOERGEL, D. et al. Reengineering thesauri for new applications: the agrovoc example. *Journal of Digital Information*, v. 4, n. 4, 2004. Citado 4 vezes nas páginas 23, 53, 97 e 100.
- SOFTEX. *Guia de Implementação – Parte 1: Fundamentação para Implementação do Nível G do MR-MPS-SW:2016*. [S.l.], 2016. Acessado em 16 de setembro de 2016. Disponível em: <http://www.softex.br/wp-content/uploads/2016/04/MPS.BR_Guia_de_Implementacao_Parte_1_2016.pdf>. Citado na página 172.
- SOFTEX. *MPS.BR - Melhoria de Processo do Software Brasileiro - Guia Geral MPS de Software*. [S.l.], 2016. Acessado em 16 de setembro de 2016. Disponível em: <http://www.softex.br/wp-content/uploads/2016/04/MPS.BR_Guia_Geral_Software_2016-com-ISBN.pdf>. Citado na página 172.
- SOLINGEN, R.; BERGHOUT, E. *The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development*. [S.l.]: McGraw-Hill Inc., 1999. Citado na página 143.
- SOMMERVILLE, I. *Software Engineering*. 9th. ed. [S.l.]: Pearson, 2010. 792 p. Citado 3 vezes nas páginas 33, 154 e 261.
- SOUZA, E.; FALBO, R.; VIJAYKUMAR, N. Using ontology patterns for building a reference software testing ontology. In: *17th IEEE International Enterprise Distributed Object Computing Conference Workshops*. [S.l.: s.n.], 2013. p. 21–30. Citado na página 277.
- SOYDAN, G.; KOKAR, M. An owl ontology for representing the cmmi-sw model. In: *Proceedings 2nd Workshop on Semantic Web Enabled Software Engineering*. [S.l.: s.n.], 2006. Citado na página 279.
- SPENCER, D.; FEATHERSTONE, D. *A Practical Guide to Information Architecture*. Penarth, UK: Five Simple Steps, 2010. 310 p. Citado 2 vezes nas páginas 28 e 78.
- SPITERI, L. A simplified model for facet analysis. *Canadian Journal of Information and Library Science*, v. 23, p. 1–30, 1998. Citado 3 vezes nas páginas 69, 70 e 200.
- SRINIVAS, C.; RADHAKRISHNA, V.; RAO, C. Clustering and classification of software component for efficient component retrieval and building component reuse libraries. *Procedia Computer Science*, v. 31, p. 1044 – 1050, 2014. ISSN 1877-0509. Citado na página 164.
- STAAB, S. et al. Knowledge processes and ontologies. *IEEE Intelligent Systems*, v. 16, n. 1, p. 26–34, Jan 2001. ISSN 1541-1672. Citado 2 vezes nas páginas 108 e 187.

- STELLATO, A. et al. Collaborative development of multilingual thesauri with vocbench (system description and demonstrator). In: _____. *The Semantic Web: ESWC 2015 Satellite Events: ESWC 2015 Satellite Events, Portorož, Slovenia, May 31 – June 4, 2015, Revised Selected Papers*. Cham: Springer International Publishing, 2015. p. 149–153. Citado 2 vezes nas páginas 104 e 207.
- STOCK, W. Concepts and semantic relations in information science. *Journal of the American Society for Information Science and Technology*, v. 61, n. 10, p. 1951–1969, 2010. Citado 8 vezes nas páginas 53, 59, 60, 87, 184, 261, 266 e 268.
- STOICA, E.; HEARST, M. Automating creation of hierarchical faceted metadata structures. In: *Procs. of the Human Language Technology Conference*. [S.l.: s.n.], 2007. Citado 3 vezes nas páginas 31, 123 e 128.
- STRASUNSKAS, D.; TOMASSEN, S. Empirical insights on a value of ontology quality in ontology-driven web search. In: *Proceedings of the OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008. Part II on On the Move to Meaningful Internet Systems*. Berlin, Heidelberg: Springer-Verlag, 2008. (OTM '08), p. 1319–1337. ISBN 978-3-540-88872-7. Citado 2 vezes nas páginas 141 e 184.
- STRASUNSKAS, D.; TOMASSEN, S. On variety of semantic search systems and their evaluation methods. In: *Proceedings of the International Conference on Information Management and Evaluation*. [S.l.]: Academic Conferences International Limited, 2010. Citado 3 vezes nas páginas 54, 57 e 269.
- STRODE, D.; HUFF, S. A taxonomy of dependencies in agile software development. In: *23rd Australasian Conference on Information Systems, ACIS 2012*. Geelong, VIC, Australia: [s.n.], 2012. Citado na página 274.
- STVILIA, B. A model for ontology quality evaluation. *First Monday*, v. 12, n. 2, 2007. Acessado em 7 de maio de 2017. Disponível em: <<http://firstmonday.org/article/view/2043/1905>>. Citado 8 vezes nas páginas 141, 142, 146, 151, 152, 182, 183 e 184.
- STVILIA, B. et al. A framework for information quality assessment. *Journal of the American Society for Information Science and Technology*, v. 58, n. 12, p. 1720–1733, 2007. Citado 2 vezes nas páginas 145 e 181.
- SULISTIO, A.; YEO, C.; BUYYA, R. A taxonomy of computer-based simulations and its mapping to parallel and distributed systems simulation tools. *Software - Practice and Experience*, v. 34, n. 7, p. 653–673, 2004. Citado na página 275.
- SUN, J.; MIAO, H.; CAO, X. A domain formal ontology and the application in service component retrieval. In: *International Conference on Software Engineering Advances*. Tahiti: IEEE, 2006. Citado 2 vezes nas páginas 164 e 280.
- SUN, X. et al. Msr4sm: Using topic models to effectively mining software repositories for software maintenance tasks. *Information and Software Technology*, v. 66, p. 1–12, 2015. Citado na página 164.
- SUOMINEN, O. et al. *Publishing SKOS vocabularies with Skosmos*. 2015. Acessado em 21 de setembro de 2015. Disponível em: <<http://skosmos.org/publishing-skos-vocabularies-with-skosmos.pdf>>. Citado na página 111.

SURE, Y.; STAAB, S.; STUDER, R. On-to-knowledge methodology (otkm). In: STAAB, S.; STUDER, R. (Ed.). *Handbook on Ontologies*. [S.l.]: Springer Berlin Heidelberg, 2004, (International Handbooks on Information Systems). p. 117–132. ISBN 978-3-662-11957-0. Citado 2 vezes nas páginas 108 e 187.

SVAHNBERG, M.; GURP, J. V.; BOSCH, J. A taxonomy of variability realization techniques. *Software - Practice and Experience*, v. 35, n. 8, p. 705–754, 2005. Citado na página 272.

SVENONIUS, E. *The Intellectual Foundation of Information Organization*. 1st. ed. Cambridge, MA, USA: The MIT Press, 2000. 264 p. Citado 15 vezes nas páginas 58, 59, 62, 63, 67, 71, 72, 82, 84, 85, 86, 87, 101, 187 e 262.

SVENONIUS, E. Encyclopedia of library and information science. In: _____. New York, NY, USA: [s.n.], 2003. cap. Design of Controlled Vocabularies, p. 822 – 838. Citado 6 vezes nas páginas 82, 84, 87, 101, 187 e 267.

SWANSON, E. The dimensions of maintenance. In: *ICSE '76 Proceedings of the 2nd international conference on Software engineering*. CA, USA: IEEE Computer Society Press, 1976. p. 492–497. Citado na página 273.

SYAL, P. *An Introduction to Linguistic: Language, grammar and semantics*. 2nd. ed. [S.l.]: PHI Learning Private Limited, 2010. 188 p. Citado 7 vezes nas páginas 52, 53, 63, 88, 262, 267 e 268.

TABLESGENERATOR.COM. *LaTeX Table Generator*. 2017. Acessado em 30 de agosto de 2017. Disponível em: <<http://www.tablesgenerator.com/>>. Citado na página 41.

TAIVALSAARI, A.; MIKKONEN, T. Objects in the cloud may be closer than they appear: Towards a taxonomy of web-based software. In: *Proceedings - 13th IEEE International Symposium on Web Systems Evolution, WSE 2011*. Williamsburg, VA, USA: IEEE, 2011. p. 59–64. Citado na página 275.

TALEVSKI, A.; WONGTHONGTHAM, P.; KOMCHALIOW, S. Towards a software component ontology. In: *iiWAS '08 Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services*. New York, NY, USA: ACM, 2008. p. 503–507. Citado na página 277.

TANG, A.; LIANG, P.; VLIET, H. Software architecture documentation: The road ahead. In: *Proceedings of the 9th Working IEEE/IFIP Conference on Software Architecture (WICSA)*. Boulder, CO: [s.n.], 2011. p. 252 – 255. Citado na página 164.

TAYLOR, A.; JOUDREY, D. *The Organization of Information*. 3rd. ed. West Port, Conn.: Libraries Unlimited, 2009. 512 p. Citado 16 vezes nas páginas 28, 47, 58, 62, 64, 72, 73, 74, 82, 119, 120, 123, 124, 125, 126 e 206.

TENNIS, J. Two axes of domains for domain analysis. *Knowledge Organi*, v. 30, n. 3/4, p. 191–195, 2003. Citado na página 195.

TEXWORKS. *TeXworks - Lowering the entry barrier to the TeX world*. 2015. Acessado em 17 de setembro de 2016. Disponível em: <<https://www.tug.org/texworks/>>. Citado na página 41.

- The Information Architecture Institute. *What is Information Architecture ?* 2013. Acessado em 28 de janeiro de 2014. Disponível em: <http://iainstitute.org/en/learn/resources/what_is_ia.php>. Citado na página 28.
- The Open Group. *TOGAF Version 9.1*. [S.l.]: Van Haren Publishing, 2011. 654 p. Citado 3 vezes nas páginas 36, 176 e 259.
- The Open Group. *ArchiMate 2.0 Translation Glossary: English – Brazilian Portuguese*. 2013. Citado 2 vezes nas páginas 38 e 264.
- THE APACHE SOFTWARE FOUNDATION. *Apache OpenOffice O Pacote de Produtividade Livre e Aberto*. 2017. Acessado em 30 de agosto de 2017. Disponível em: <<https://www.openoffice.org/pt-br/>>. Citado na página 41.
- THE GETTY RESEARCH INSTITUTE. *Art & Architecture Thesaurus Online - About the AAT*. 2015. Acessado em 24 de agosto de 2016. Disponível em: <<http://www.getty.edu/research/tools/vocabularies/aat/about.html>>. Citado 4 vezes nas páginas 71, 104, 200 e 203.
- THE STANDISH GROUP. *The CHAOS Manifesto: Think big, act small*. [S.l.], 2013. Citado na página 33.
- TIBALDO, M. et al. Redefinition and statistical analysis of measures for evaluating the quality of ontologies. In: *I Simposio Argentino de Ontologías y sus Aplicaciones (SAOA)*. Rosario: [s.n.], 2015. p. 51–60. Citado 3 vezes nas páginas 152, 183 e 184.
- TOMS, E. Information interaction: Providing a framework for information architecture. *Journal of the American Society for Information Science and Technology*, v. 53, n. 10, p. 855–862, 2002. Citado na página 28.
- TUDHOPE, D.; ALANI, H.; JONES, C. Augmenting thesaurus relationships: Possibilities for retrieval. *Journal of Digital Information*, v. 1, n. 8, 2006. ISSN 1368-7506. Citado na página 87.
- TZITZIKAS, Y. Evolution of faceted taxonomies and ctca expressions. *Knowledge and Information Systems*, Springer-Verlag, v. 13, n. 3, p. 337–365, 2007. ISSN 0219-1377. Citado na página 92.
- URDICIAIN, B. Evaluación semántica y estructural de tesauros. *Revista General de Información y Documentación*, v. 8, n. 2, p. 193–199, 1998. Citado na página 186.
- USCHOLD, M. Building ontologies: Towards an unified methodology. In: *Proceedings of the 16th Annual Conference of the British Computer Society Specialist Group on Expert Systems*. Cambridge, UK: [s.n.], 1996. Acessado em: 20 de fevereiro de 2015. Disponível em: <<http://www.aiai.ed.ac.uk/project/pub/documents/1996/96-es96-unified-method.pdf>>. Citado 3 vezes nas páginas 96, 97 e 265.
- USCHOLD, M. Knowledge level modeling: concepts and terminology. *The Knowledge Engineering Review*, v. 13, n. 1, p. 5–29, 1998. Citado na página 96.
- USCHOLD, M.; GRUNINGER, M. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, v. 11, p. 93–136, 1996. Citado 4 vezes nas páginas 82, 106, 152 e 183.

USCHOLD, M.; JASPER, R. A framework for understanding and classifying ontology applications. In: *Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods*. Stockholm, Sweden: [s.n.], 1999. Citado na página 97.

USING, S.; AHMAD, R.; TAIB, S. Ontology of programming resources for semantic searching of programming related materials on the web. In: *2010 International Symposium on Information Technology, ITSIM'10*. [S.l.: s.n.], 2010. v. 2, p. 698–703. Citado na página 277.

UTTING, M.; PRETSCHNER, A.; LEGEARD, B. A taxonomy of model-based testing approaches. *Software testing, verification & reliability*, v. 22, n. 5, p. 297–312, 2012. Citado na página 274.

VALASKI, J. et al. Ontology to classify learning material in software engineering knowledge domain. In: *4th Seminar on Ontology Research in Brazil, ONTOBRAS 2011 - 6th International Workshop on Metamodels, Ontologies and Semantic Technologies, MOST 2011*. Gramado, Brazil: [s.n.], 2011. v. 776, p. 37–48. Citado na página 280.

VALERIO, A.; SUCCI, G.; FENAROLI, M. Domain analysis and framework-based software development. *ACM SIGAPP Applied Computing Review - Special issue on frameworks and patterns in software reuse*, ACM, New York, NY, USA, v. 5, n. 2, p. 4–15, 1997. ISSN 1559-6915. Disponível em: <<http://doi.acm.org/10.1145/297075.297081>>. Citado na página 134.

VANDERLEI, T. et al. A cooperative classification mechanism for search and retrieval software components. In: *Proceedings of the 2007 ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2007. (SAC '07), p. 866–871. ISBN 1-59593-480-4. Citado na página 164.

VATICAN LIBRARY. *History*. 2017. Acessado em 1 de agosto de 2015. Disponível em: <<https://www.vatlib.it/home.php?pag=storia>>. Citado na página 23.

VEGAS, S.; BASILI, V. A characterisation schema for software testing techniques. *Empirical Software Engineering*, v. 10, n. 4, p. 437–466, October 2005. Citado na página 274.

VEGI, L.; LISBOA-FILHO, J.; CROMPVOETS, J. A machine-processable dublin core application profile to analysis patterns to provide linked data. In: *International Conference on Dublin Core and Metadata Applications*. [S.l.: s.n.], 2012. p. 23–32. Citado na página 169.

VICKERY, B. *Faceted classification: a guide to construction and use of special schemes*. London: Aslib, 1960. 70 p. Citado na página 70.

VICKERY, B. *Faceted classification schemes*. New Brunswick, N.J.: Graduate School of Library Service at Rutgers University, 1966. v. 5. 108 p. (Rutgers series on systems for the Intellectual Organization of Information, v. 5). Citado 2 vezes nas páginas 70 e 71.

VISION, C. *Astah*. 2016. Acessado em 21 de dezembro de 2016. Disponível em: <<http://astah.net/>>. Citado na página 41.

VIZCAINO, A. et al. Towards an ontology for global software development. *IET Software*, v. 6, n. 3, p. 214–225, 2012. Citado na página 279.

- VRANDECIC, D. *Ontology evaluation*. Tese (Doutorado) — Karlsruhe Institute of Technology, 2010. Citado na página 151.
- WAGNER, S. *Software Product Quality Control*. [S.l.]: Springer, 2013. Citado 2 vezes nas páginas 147 e 180.
- WAL, T. V. *Folksonomy*. 2007. Acessado em: 20 de fevereiro de 2015. Disponível em: <<http://vanderwal.net/folksonomy.html>>. Citado na página 73.
- WALIA, G.; CARVER, J. A systematic literature review to identify and classify software requirement errors. *Information and Software Technology*, v. 51, n. 7, p. 1087 – 1109, 2009. ISSN 0950-5849. Citado na página 271.
- WALIA, G.; CARVER, J. Using error abstraction and classification to improve requirement quality: Conclusions from a family of four empirical studies. *Empirical Software Engineering*, v. 18, n. 4, p. 625–658, 2013. Citado na página 271.
- WAND, Y.; WANG, R. Anchoring data quality dimensions in ontological foundations. *Commun. ACM*, ACM, New York, NY, USA, v. 39, n. 11, p. 86–95, nov. 1996. ISSN 0001-0782. Citado 2 vezes nas páginas 144 e 145.
- WAND, Y.; WEBER, R. Research commentary: Information systems and conceptual modeling—a research agenda. *Information Systems Research*, v. 13, n. 4, p. 363–376, 2002. Citado na página 133.
- WANG, R.; PIERCE, E.; MADNICK, S. *Information Quality*. [S.l.]: Routledge, 2014. Citado 3 vezes nas páginas 138, 144 e 145.
- WANG, R.; STRONG, D. Beyond accuracy: What data quality means to data consumer. *Journal of Management Information Systems*, v. 12, n. 4, p. 5–34, 1996. Citado na página 145.
- WARNER, A. *Taxonomy Primer*. 2002. Acessado em: 28 de janeiro de 2014. Disponível em: <<https://www.ischool.utexas.edu/~i385e/readings/Warner-aTaxonomyPrimer.html>>. Citado 4 vezes nas páginas 81, 82, 91 e 269.
- WEBSTER, K. B.; OLIVEIRA, K. D.; ANQUETIL, N. A risk taxonomy proposal for software maintenance. In: *21st IEEE International Conference on Software Maintenance, ICSM 2005*. Budapest, Hungary: [s.n.], 2005. p. 453–464. Citado na página 273.
- WEI, B. et al. A survey of faceted search. *Journal of Web Engineering*, Rinton Press, Incorporated, Paramus, NJ, v. 12, n. 1-2, p. 41–64, 2013. ISSN 1540-9589. Citado na página 51.
- WEI, W.; BARNAGHI, P.; BARGIELA, A. Search with meanings: An overview of semantic search systems. *International Journal of Communications of SIWN*, v. 3, p. 76–82, 2008. Citado na página 54.
- WEIBEL, S. Metadata: The foundations of resource description. *D-Lib Magazine*, July 1995. Acessado em 27 de abril de 2015. Disponível em: <<http://www.dlib.org/dlib/July95/07weibel.html>>. Citado na página 130.
- WELLS, D. *Extreme Programming: A gentle introduction*. 2013. Acessado em 8 de maio de 2015. Disponível em: <<http://www.extremeprogramming.org/>>. Citado na página 162.

- WILD, P.; GIESS, M.; MCMAHON, C. Describing engineering documents with faceted approaches observations and reflections. *Journal of Documentation*, v. 65, n. 3, p. 420–445, 2009. Citado na página 68.
- WILL, L. *Software for building and editing thesauri*. 2013. Acessado em 5 de fevereiro de 2016. Disponível em: <<http://www.taxobank.org/content/thesauri-and-vocabulary-control-thesaurus-software>>. Citado na página 111.
- WILLE, C. et al. E-learning infrastructure for software engineering education: Steps in ontology modeling for swebok. In: *IASTED International Conference on Software Engineering*. [S.l.: s.n.], 2004. Citado na página 280.
- WONGTHONGTHAM, P. et al. Software engineering ontologies and their implementation. In: *IASTED International Conference on Software Engineering*. Innsbruck, Austria: [s.n.], 2005. p. 208–213. Citado na página 168.
- WONGTHONGTHAM, P. et al. A software engineering ontology as software engineering knowledge representation. In: *3rd International Conference on Convergence and Hybrid Information Technology, ICCIT 2008*. Busan, South Korea: [s.n.], 2008. v. 2, p. 668–675. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-57849167063&partnerID=40&md5=b2d98fbc6c154ea115360fa736f7af9e>>. Citado na página 280.
- WORKING GROUP ON GUIDELINES FOR MULTILINGUAL THESAURI. *Guidelines for Multilingual Thesauri*. [S.l.], 2009. Citado na página 140.
- WRIGHT, A. *Cataloging the World: Paul Otlet and the Birth of the Information Age*. 1st edition. ed. [S.l.]: Oxford University Press, 2014. Citado na página 23.
- WU, J. A novel software engineering knowledge representation method for multi-site software development. In: *2012 IEEE 3rd International Conference on Software Engineering and Service Science (ICSESS)*. Beijing: IEEE, 2012. p. 523–526. Citado na página 280.
- YAO, H.; ETZKORN, L.; VIRANI, S. Automated classification and retrieval of reusable software components. *Journal of the American Society for Information Science and*, v. 59, n. 4, p. 613–627, 2008. Citado na página 164.
- YAO, H.; ORME, A.; ETZKORN, L. Cohesion metrics for ontology design and application. *Journal of Computer Science*, v. 1, n. 1, p. 107–113, 2005. Citado 2 vezes nas páginas 184 e 186.
- YEE, K. et al. Faceted metadata for image search and browsing. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2003. (CHI '03), p. 401–408. ISBN 1-58113-630-7. Citado na página 123.
- YU, J.; THOM, J.; TAM, A. Ontology evaluation: using wikipedia categories for browsing, in. In: *Proceedings of the 16th Conference on Information and Knowledge Management, ACM*. [S.l.]: Press, 2007. p. 223–232. Citado na página 152.
- YU, L. *A Developer's Guide to the Semantic Web*. New York: Springer, 2011. 608 p. Citado 3 vezes nas páginas 113, 114 e 117.

ZHANG, H.; LI, Y.; TAN, H. Measuring design complexity of semantic web ontologies. *Journal of Systems and Software*, Elsevier Science Inc., New York, NY, USA, v. 83, n. 5, p. 803–814, maio 2010. Citado 3 vezes nas páginas [152](#), [183](#) e [184](#).

ZHANG, L. et al. A novel approach of components retrieval in large-scale component repositories. In: *Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on*. [S.l.: s.n.], 2012. p. 220–223. Citado na página [164](#).

ZHAO, Y.; DONG, J.; PENG, T. Ontology classification for semantic-web-based software engineering. *IEEE Transactions on Services Computing*, v. 2, n. 4, p. 303–317, 2009. ISSN 1939-1374. Citado 3 vezes nas páginas [33](#), [168](#) e [169](#).

Glossário

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#)

A

Anotação (1) processo de atribuição de informação a recurso de informação, ou resultado desse processo ([KIRYAKOV et al., 2004](#); [OREN et al., 2006](#)).

Análise de domínio (1) método para entender necessidades de informação e práticas em domínios e comunidades ([ABBAS, 2010](#)) **(2)** definição da base de conhecimento de uma comunidade ([SMIRAGLIA, 2012](#)) **(3)** processo por meio do qual a informação usada no desenvolvimento de software é identificada, capturada e organizada com o propósito de torná-la reusável na criação de novos sistemas ([PIETO-DIAZ, 1990](#)) **(4)** processo de identificação, coleta, organização e representação da informação relevante em um domínio, baseado no estudo de sistemas existentes e de suas histórias de desenvolvimento, no conhecimento capturado de especialistas no domínio, em teorias inerentes e tecnologias emergentes no domínio ([KANG et al., 1990](#)) **(5)** processo que objetiva definir estrutura de domínio e requisitos, e capturá-los em modelo de domínio ([HOLIBAUGH, 1993](#)) **(6)** processo em que domínio é analisado para construir modelo de domínio ([ALBRECHTSEN, 1992](#)).

Arcabouço (1) estrutura para entendimento de um domínio, estabelece conceitos e vocabulário para um conjunto de teorias explicativas; provê uma estrutura para entendimento de algo sendo apresentado ou explorado ([ABBAS, 2010](#); [GLUSHKO, 2013](#)).

Arcabouço de arquitetura (1) estrutura conceitual, ou estruturas conceituais, para desenvolvimento, implementação e manutenção de arquiteturas ([ISO, 2011e](#); [The Open Group, 2011](#)).

Arquitetura de sistema (1) estrutura organizacional de um sistema ([IEEE, 1990](#); [ISO, 2011e](#)).

Artefato (1) recurso de informação criado por uma ou por mais pessoas ([GLUSHKO, 2013](#)).

Atividade (1) corpo de trabalho a ser realizado, consiste de descrição de transformação de informação de entrada em informação de saída ([IEEE, 2006](#)).

Ativo (1) recurso com valor reconhecido sob controle de indivíduo ou organização ([DAMA, 2010](#)).

B

Busca semântica (1) busca onde termos são expandidos e ambiguidades removidas por meio de sinônimos e de termos mais específicos relacionados em vocabulário controlado (GIUNCHIGLIA; DUTTA; MALTESE, 2013).

C

Ciclo de vida (1) evolução, da concepção à retirada, de sistema, produto, serviço, projeto ou entidade feita pelo ser humano (ISO, 2008a; ISO, 2008b).

Ciclo de vida de projeto de software (1) porção de ciclo de vida de software aplicável a projeto específico de software (ISO, 2010).

Classificação (1) sistema de classes ordenado de acordo com conjunto de princípios para organizar entidades; processo pelo qual entidades são atribuídas a classes por meio da análise de suas características; conjunto de princípios por meio dos quais recursos são organizados de acordo com os seus assuntos (FEATHER; STURGES, 2003; JACOB, 2004) **(2)** processo de triagem (*sorting*) de objetos baseada em algum critério selecionado entre as propriedades dos objetos (HJØRLAND; PEDERSEN, 2005) **(3)** processo de atribuição sistemática de recursos a categorias intencionais (GLUSHKO, 2013) **(4)** processo de dividir objetos ou conceitos em hierarquias lógicas de classes, com base em características que as classes tem em comum e em características que as distinguem (REITZ, 2012).

Classificação facetada (1) sistema para organizar materiais em categorias baseado em combinação sistemática de características mutuamente exclusivas e coletivamente exaustivas dos materiais e apresentar características em um modo que mostre as suas relações (PEARCE-MOSES, 2005).

Computador (1) unidade funcional que pode realizar computações substanciais, inclusive numerosas operações aritméticas e lógicas sem a intervenção humana (ISO, 2010).

Conceituação (1) visão simplificada e abstrata do mundo que se deseja representar para determinado propósito (GRUBER, 1993) **(2)** interpretação estruturada de parte do mundo, usada por pessoas para pensar e comunicar acerca do mundo (BORST, 1997) **(3)** conjunto de conceitos usado para articular abstrações de estado de coisas em um dado domínio (GUIZZARDI, 2005) **(4)** modelo abstrato que possibilita descrever algo relevante do mundo (CORCHO; POVEDA-VILLALÓN; GÓMEZ-PÉREZ, 2015).

Controle de qualidade (1) conjunto de atividades desenhadas (*designed*) com o intuito de avaliar qualidade de produtos desenvolvidos ou fabricados (ISO, 2010).

D

Domínio (1) área de conhecimento ou atividade caracterizada por conjunto de conceitos e terminologia entendida por aqueles que atuam na área (ISO, 2010).

Domínio de conhecimento (1) área temática que pode ser delimitada (STOCK, 2010).

E

Engenharia de Software (1) estudo e aplicação de abordagem sistemática, disciplinada e quantificável ao desenvolvimento, à operação e à manutenção de software; a aplicação de engenharia ao software (ISO, 2010) **(2)** disciplina de engenharia relacionada a todos os aspectos de produção de software (SOMMERVILLE, 2010).

Engenheiro de *software* (1) pessoa que aplica princípios de Engenharia de *Software* ao *design*, desenvolvimento, manutenção, teste e avaliação de *software* (IEEE, 1990; PMI, 2013).

Escala de medição (1) conjunto ordenado de valores, contínuo ou discreto, ou um conjunto de categorias para o qual o atributo é mapeado (ISO, 2010).

Espaço de informação (1) coleção delimitada de recursos de informação (GÖDERT; HUBRICH; NAGELSCHMIDT, 2014) **(2)** delimitação de uma coleção de registros (SIQUEIRA, 2008).

F

Faceta (1) categoria genérica independente de domínio, que pode ser definida a priori; categoria específica dependente de domínio, primariamente definida a posteriori considerando-se conhecimento e investigação sobre categorias em domínio; categoria específica, articulada por usuários ao expressarem necessidades de informação (ALBRECHTSEN, 1992).

Falha (1) término da capacidade de um produto de executar uma função requerida ou a sua incapacidade de executá-la dentro de limites previamente especificados (ABNT, 2003).

Ferramenta (1) algo tangível usado na realização de atividade, voltado a produzir produto ou resultado (PMI, 2013).

Ferramenta semântica (1) ferramenta que provê informação semântica (HJØRLAND, 2007c).

Folksonomia (1) conjunto de etiquetas resultantes de processo de etiquetagem individual ou colaborativo (ABBAS, 2010).

G

Garantia de qualidade (1) padrão planejado e sistemático de todas ações necessárias para prover confiança adequada que um item ou um produto atende a requisitos técnicos estabelecidos (ISO, 2010) (2) conjunto de atividades definidas para avaliar processo pelo qual produtos são desenvolvidos e são fabricados (ISO, 2010).

Garantia de usuário (1) critério que justifica a escolha de termo pelo uso frequente do mesmo em solicitações de informação acerca do conceito ou em consultas de usuários (NISO, 2005; SVENONIUS, 2000).

Garantia estrutural (1) critério que justifica a escolha do termo por ter função estrutural, melhorar precisão, melhorar revocação e/ou facilitar navegação (SVENONIUS, 2000).

Garantia literária (1) critério que justifica a escolha de termo pelo uso frequente do mesmo na literatura (NISO, 2005; SVENONIUS, 2000).

Garantia organizacional (1) critério que justifica a escolha de termo pelas características e contexto da organização (NISO, 2005).

Guia de melhores práticas (1) fonte de informação que pode conter instruções, exemplos ou conselhos com o intuito de evitar erros comuns (ABBAS, 2010).

H

Hardware de computador (1) equipamento físico usado para processar, armazenar ou transmitir programas de computador ou dados (ISO, 2010).

Hiponímia (1) relacionamento entre item léxico mais genérico e um item léxico mais específico (SYAL, 2010).

Homofonia (1) fenômeno onde as palavras têm a mesma pronúncia, mas diferentes significados ou ortografias (SYAL, 2010).

Homografia (1) fenômeno onde duas ou mais palavras tem a mesma ortografia, mas diferente pronúncia ou significado (SYAL, 2010).

Homonímia (1) relação entre palavras com mesma forma fonética, mas com diferentes significados (SYAL, 2010).

I

Informação semântica (1) informação acerca dos significados de símbolos e de relações entre conceitos (HJØRLAND, 2007c).

Informação tecnológica (1) informação relacionada com o modo de fazer um produto ou prestar um serviço para colocá-lo no mercado (AGUIAR, 1991) (2) informação que trata da informação necessária, utilizada e gerada em procedimentos de aquisição, inovação e transferência de tecnologia; em procedimentos de metrologia, certificação da qualidade e normalização; em processos de produção (JANNUZZI; MONTALI, 1999).

Internet (1) rede global de computadores que utiliza a família de protocolos TCP/IP (ALBUQUERQUE, 2001).

L

Linguagem (1) meio sistemático para se comunicar ideias por sinais, sons, gestos ou marcas convencionadas, e regras para formação de expressões aceitáveis; um meio de comunicação, com sintaxe e semântica, consistindo de um conjunto de representações, convenções e regras associadas usadas para transmitir informação (IEEE, 1990).

M

Medida (1) valor numérico calculado a partir de dados de entrada (ARCHER; STINSON, 1995).

Medição (1) processo de comparação quantitativa entre resultados e requisitos (ASQ, 2017).

Meronímia (1) relacionamento de inclusão que reflete o relacionamento entre todo e parte (CRUSE, 2011).

Metadado (1) informação criada ou capturada sobre recurso de informação, por meio da qual é possível realizar certas funções em relação ao recurso de informação acerca do qual é o metadado (MILLER, 2011); descrição estruturada dos atributos essenciais de objeto de informação (GILL, 2008) (2) informação estruturada ou descritiva sobre recurso, é informação coletada e registrada sobre recurso (DLESE, 2005); informação estruturada sobre recurso de informação de qualquer tipo de mídia ou formato (CAPLAN, 2003) (3) informação estruturada que descreve, explica, localiza e facilita a recuperação, uso ou gestão de recurso de informação (NISO, 2004) (4) informação estruturada associada a objeto com propósito de descoberta, descrição, uso, gestão e preservação (NISO, 2007) (5) informação sobre objeto digital ou físico (IEEE, 2002) (6) enunciado sobre recurso potencialmente informativo (POMERANTZ, 2015) (7) descrição de informação em rede e de recursos digitais segundo padrão ou arcabouço criado para esse propósito, dados de catalogação e indexação criados para qualquer tipo de documento por meio de métodos tradicionais de descrição e organização

da informação (CHU, 2010) (8) qualquer enunciado sobre recurso de informação (GARSHOL, 2004) (9) esquema para descrever recurso de informação (NISO, 2004).

Metodologia (1) sistema de métodos a ser usado em classe de trabalho intelectualmente intensivo (IEEE, 1996; PMI, 2013).

Mineração de texto (1) processo de analisar texto para extrair informação com propósito particular (GOH et al., 2009).

Modelo (1) representação de algo, que suprime certos aspectos do que é modelado (IEEE, 2012b) (2) abstração de dada porção da realidade, articulada de acordo com conceituação de um domínio (GUIZZARDI, 2005).

Modelo conceitual (1) representação composta por conceitos relevantes a um empreendimento, que suprime certos aspectos do que é modelado (IEEE, 2012b).

Modelo de ciclo de vida (1) arcabouço de processos e atividades relacionadas a ciclo de vida, pode ser organizado em fases e servir de referência para comunicação e entendimento (ISO, 2008a; ISO, 2008b).

Modelo de domínio (1) modelo conceitual que engloba definições de funções, objetos e relações em domínio (KANG et al., 1990).

Modelo de maturidade de capacidade (1) modelo composto por elementos considerados essenciais para que processos sejam considerados efetivos, e que descreve caminho para melhoria de processos em uma ou mais disciplinas (ISO, 2010).

Mono-hierarquia (1) estrutura hierárquica onde cada termo tem só um termo imediatamente acima na hierarquia (GÖDERT; HUBRICH; NAGELSCHMIDT, 2014).

Monoreferencialidade (1) princípio, segundo o qual, um termo só pode representar um conceito (DODEBEI, 2002).

Método (1) abordagem definida e reproduzível para tratar certo tipo de problema (The Open Group, 2013).

Métrica (1) medida quantitativa do grau com o qual um sistema, componente, ou processo possui um dado atributo (ISO, 2010) (2) padrão para medição (ASQ, 2017).

Métrica de qualidade (1) descrição de atributo de projeto ou de produto e como o medir (PMI, 2013).

Norma (1) documento estabelecido por consenso e aprovado por um organismo reconhecido, que fornece regras, diretrizes ou características mínimas para atividades ou para seus resultados, visando à obtenção de um grau ótimo de ordenação em um dado contexto (ABNT, 2014).

O

Ontologia (1) teoria acerca da natureza da existência, dos tipos de coisas que existem, ou disciplina que estuda essas teorias (BERNERS-LEE; HENDLER; LASSILA, 2001) (2) teoria do que existe, estudo dos tipos de entidades na realidade e das relações entre essas entidades (ARP; SMITH; SPEAR, 2015) (3) ramo da Filosofia que visa prover classificação de entidades e relações em todas as esferas da realidade, tanto material quanto abstrata (DAVIES, 2010; ORG, 2007) (4) especificação explícita de uma conceituação, onde conceituação é visão simplificada e abstrata do mundo que se deseja representar para determinado propósito (GRUBER, 1993) (5) representação formal de conceitos em um domínio de discurso e de relações entre esses conceitos (REIMER, 2011b) (6) modelo de domínio de interesse descrito em uma linguagem lógica (DAVIES, 2010) (7) definição formal de conceituação compartilhada, onde conceituação consiste de interpretação estruturada de parte do mundo, usada por pessoas para pensar e comunicar acerca do mundo (BORST, 1997) (8) descrição ou representação explícita de parte de uma conceituação, onde conceituação é visão do mundo, forma de pensar sobre um domínio, tipicamente expressa por conjunto de conceitos, suas definições e suas inter-relações (USCHOLD, 1996).

P

Poli-hierarquia (1) estrutura hierárquica onde há termo com mais de um termo imediatamente acima na hierarquia (GÖDERT; HUBRICH; NAGELSCHMIDT, 2014).

Princípio de organização (1) diretiva para desenho ou arranjo de coleções de recursos (GLUSHKO, 2013).

Processo (1) série de atividades direcionada a causar um resultado final de modo que uma ou mais entradas sejam atuadas com o intuito de criar uma ou mais saídas (IEEE, 1990; PMI, 2013) (2) coleção de atividades que recebe uma ou mais entradas, e cria saídas de valor para a organização (ISACA, 2011) (3) conjunto estruturado de atividades projetado para atingir um objetivo específico (OGC, 2009).

Processo de desenvolvimento de ontologia (1) processo por meio do qual as necessidades de usuários são traduzidas em uma ontologia (BAONZA, 2010).

Processo de desenvolvimento de software (1) tradução de necessidades de usuários ou clientes em produtos de software (ISO, 2010).

Processo de software (1) conjunto coerente de políticas, estruturas organizacionais, tecnologias, procedimentos e artefatos necessários à concepção, desenvolvimento, implantação e manutenção de produto de *software* (FUGGETTA, 2000).

Produtividade (1) relação entre produto do trabalho e esforço do trabalho (ISO, 2010).

Produto (1) resultado de um processo (ISO, 2008b).

Programa de computador (1) combinação de instruções de computador e definições de dados que capacitam o hardware de um computador a realizar funções computacionais ou de controle (ISO, 2010).

Projeto (1) empreendimento com início e fim definidos, realizado para criar produto ou serviço de acordo com recursos e requisitos especificados, empreendimento temporário realizado para criar produto, serviço ou resultado único (ISO, 2008a; PMI, 2013).

Pré-coordenação (1) processo onde termos são combinados na indexação (AITCHISON; BAWDEN; GILCHRIST, 2000; HEDDEN, 2010).

Pós-coordenação (1) processo onde termos são combinados quando de consultas (AITCHISON; BAWDEN; GILCHRIST, 2000; HEDDEN, 2010).

Q

Qualidade (1) grau com o qual um produto ou processo atende a requisitos estabelecidos (IEEE, 2014b) **(2)** adequação ao uso (*fitness for use*) (HOYER; HOYER, 2001) **(3)** habilidade de produto, serviço, sistema, componente ou processo atender necessidades, expectativas ou requisitos de cliente ou usuário (ISO, 2010) **(4)** grau com o qual um conjunto de características inerentes satisfazem requisitos (PMI, 2013) **(5)** produto ou serviço livre de deficiências (ASQ, 2017).

R

Recuperação conceitual (1) processo onde a recuperação da informação considera conhecimento sobre conceitos (GÖDERT; HUBRICH; NAGELSCHMIDT, 2014).

Recuperação da informação (1) processo por meio do qual é recuperada informação que melhor corresponde à solicitação do usuário (IEEE, 2015; PMI, 2013; SAA, 2014) **(2)** processo ou método pelo qual potencial usuário de informação é capaz de converter a sua necessidade de informação em lista de citações para documentos armazenados contendo informação de utilidade para o mesmo (MOOERS, 1951).

Recuperação da informação embasada em conceitos (1) recuperação da informação no nível de conceitos, que vai além do nível de palavras (STOCK, 2010).

Recurso (1) qualquer coisa que possa ser identificada (POWELL et al., 2007) (2) qualquer coisa de valor que suporte atividade orientada a meta (GLUSHKO, 2013) (3) termo genérico que designa objeto ou coleção de objetos de interesse e que pode ser descrito de algum modo (MILLER, 2011) (4) ativo utilizado ou consumido durante a execução de um processo (ISO, 2008b).

Recurso de informação (1) unidade descritível de informação registrada (GLUSHKO, 2013).

Rede de computadores (1) sistema composto por meios físicos e lógicos por meio dos quais é possível trocar dados e compartilhar recursos entre computadores (ALBUQUERQUE, 2001).

Relação léxica (1) relação entre palavras (HJØRLAND, 2007b).

Relação paradigmática (1) relação permanente, verdadeira a priori ou por definição (LANCASTER, 1986; SVENONIUS, 2003).

Relação semântica (1) em sentido estrito, designa relação entre conceitos (HJØRLAND, 2007b) (2) em sentido amplo, designa relação entre conceitos ou entre palavras (HJØRLAND, 2007b).

Relação sintagmática (1) relação verdadeira só em determinados contextos (LANCASTER, 1986; SVENONIUS, 2003).

Repositório digital (1) local onde são depositados recursos de informação e/ou metadados sobre recursos de informação (MILLER, 2011).

Representação da informação (1) processo que envolve analisar assunto de recurso de informação e colocar o resultado dessa análise numa expressão linguística, ou resultado desse processo (NOVELLINO, 1996).

Requisito (1) enunciado que traduz ou expressa uma necessidade e suas restrições e condições associadas (ISO, 2011d).

Restrição (1) limitação externamente imposta a requisitos, desenho, ou implementação de sistema ou ao processo usado para desenvolver ou modificar um sistema (ISO, 2011d).

S

Semântica (1) estudo dos significados de palavras, expressões, sinais, sentenças e textos (HJØRLAND, 2008a) (2) nível de análise linguística que lida com o significado, nível de análise linguística onde significado é analisado (SYAL, 2010).

Significado (1) uma propriedade intrínseca (OGDEN; RICHARDS, 1923) (2) as outras palavras anexadas a uma palavra em um dicionário (OGDEN; RICHARDS, 1923) (3) a conotação de uma palavra (OGDEN; RICHARDS, 1923) (4) uma essência (OGDEN; RICHARDS, 1923) (5) aquilo ao qual o usuário de um símbolo acredita ele próprio estar referenciando (OGDEN; RICHARDS, 1923) (6) aquilo ao qual o interprete de um símbolo referencia (OGDEN; RICHARDS, 1923).

Significado conotativo (1) significado associado ao conceito, associações feitas ao conceito quando o mesmo é referenciado (SYAL, 2010).

Significado denotativo (1) qualidades essenciais de um conceito, as qualidades que distinguem o conceito de outros conceitos (SYAL, 2010).

Significado social (1) significado decorrente do uso em situações sociais e circunstâncias particulares (SYAL, 2010).

Sinonímia (1) relacionamento entre palavras onde similaridades semânticas entre palavras são mais salientes do que as diferenças (SYAL, 2010).

Sistema (1) coleção de componentes organizada para função ou conjunto de funções, combinação de componentes inter-relacionados organizados para atingir um ou mais propósitos definidos (ISO, 2010).

Sistema de armazenamento e recuperação de informação (1) operações, equipamentos, software e documentos associados, por meio dos quais recursos de informação são indexados e armazenados de modo que recursos de informação selecionados possam ser recuperados em resposta a solicitações realizadas por meio de comandos aceitos pelo sistema (NISO, 2005).

Sistema de classificação (1) lista de classes organizada de acordo com princípios estabelecidos, construída com o propósito de organizar recursos de informação em coleção, ou entradas em índice, bibliografia ou catálogo, agrupadas com base em similaridades e em diferenças, com o intuito de facilitar acesso e recuperação (REITZ, 2012).

Sistema de computador (1) sistema com um ou mais computadores e software associado (ISO, 2010).

Sistema de organização do conhecimento (1) esquema para organização da informação e promoção da gestão do conhecimento (HODGE, 2000) (2) sistema que organiza conceitos e as suas relações semânticas (HJØRLAND, 2009) (3) sistema de conceitos em um domínio (STOCK, 2010) (4) estrutura composta por conceitos relacionados construída para suporte à indexação e recuperação da informação (GÖDERT; HUBRICH; NAGELSCHMIDT, 2014).

Sistema de pesquisa semântica (1) sistema de recuperação da informação que usa tecnologias semânticas para melhorar a recuperação da informação (STRASUNSKAS; TOMASSEN, 2010).

Software (1) programas de computador, procedimentos, possivelmente documentação e dados associados, pertinentes à operação de um sistema de computador (IEEE, 2012a).

Stakeholder (1) indivíduo, equipe, organização ou classe com interesse em sistema (ISO, 2011e).

T

Taxonomia (1) meio de organizar conceitos (HEDDEN, 2010) (2) estrutura hierárquica onde relações são limitadas a relações *is a kind of* (JAKUS et al., 2013) (3) vocabulário controlado composto por termos preferidos conectados em hierarquia ou poli-hierarquia (NISO, 2005) (4) classificação baseada em assunto hierarquicamente organizada (GARSHOL, 2004).

U

Usuário (1) indivíduo ou grupo que se beneficia de um sistema durante a sua utilização (ISO, 2008b).

V

Vocabulário controlado (1) lista de termos autorizados (LANCASTER, 2004) (2) subconjunto controlado de uma linguagem natural, desenvolvido para um fim ou para uma comunidade, composto por uma lista de termos explicitamente enumerados (NISO, 2005) (3) lista de termos explicitamente enumerados com o intuito de organizar e representar informação para facilitar a recuperação da informação (MAI, 2008) (4) arranjo organizado de palavras e de frases usado para representar conceitos (HARPRING, 2010) (5) lista organizada de palavras e de frases, ou sistema de notação, para rotular conteúdo e encontrá-lo via navegação ou busca (WARNER, 2002) (6) lista padronizada de termos selecionados para uso consistente na descrição e indexação de recursos de informação (MILLER, 2011) (7) linguagem de indexação onde terminologia é controlada (LANCASTER, 1986) (8) conjunto de limitações a valores de elementos de metadados, ou conjunto pré-definido de valores de elementos de metadados (CAPLAN, 2003) (9) classe de sistema de organização de conhecimento ou classe de estrutura para organização de conhecimento (ABBAS, 2010; HODGE, 2000; NISO, 2005).

Apêndices

APÊNDICE A – Listas de vocabulários controlados

Este apêndice contém resultado de mapeamento sistemático de literatura realizado pelo autor desta tese com o objetivo de identificar vocabulários controlados no domínio da Engenharia de Software. Para facilitar o uso deste apêndice, cada vocabulário controlado está classificado considerando-se sua estrutura e a área de conhecimento principalmente abordada pelo vocabulário controlado.

A.1 Taxonomias em Engenharia de Software

Diversos vocabulários controlados classificados como taxonomias pelos seus autores foram identificados no processo de mapeamento sistemático de literatura realizado. As taxonomias identificadas abordam áreas de conhecimento da Engenharia de Software cujos nomes são listados a seguir.

- (a) Requisitos de software;
- (b) Desenho (*design*) de Software;
- (c) Construção de Software;
- (d) Manutenção de Software;
- (e) Gestão de Engenharia de Software;
- (f) Processo de Engenharia de Software;
- (g) Qualidade de Software.

A.1.1 Requisitos de Software

Foram identificadas as seguintes fontes de informação sobre taxonomias na área de conhecimento Requisitos de Software: (a) [McGee e Greer \(2009\)](#) e [McGee e Greer \(2011\)](#) contêm informação sobre taxonomia de fontes de mudanças de requisitos de software. No desenvolvimento dessa taxonomia foram realizados *workshops* com gerentes de projetos de parceiro na indústria, que consolidaram e classificaram fontes de mudanças identificadas na literatura por meio da técnica de desenho (*design*) *card sorting*; (b) [Walia e Carver \(2013\)](#) e [Walia e Carver \(2009\)](#) contêm informação sobre taxonomia de erros de requisitos

de software, no desenvolvimento dessa taxonomia cada classe de erro foi derivada de erros identificados em literatura sobre Engenharia de Software e Psicologia; (c) [Galster e Bucherer \(2008\)](#) contém informação sobre taxonomia de requisitos não funcionais no contexto de desenvolvimento orientado a serviço. O desenvolvimento dessa taxonomia englobou revisão de literatura.

A.1.2 Desenho (*design*) de Software

Foram identificadas as seguintes fontes de informação sobre taxonomias na área de conhecimento Desenho (*design*) de Software: (a) [Rajapakse \(2008\)](#) contém informação sobre taxonomia de técnicas para desfragmentar aplicações móveis (desfragmentar é fazer a aplicação se comportar do modo esperado em diversos contextos); (b) [Lau e Rana \(2010\)](#) contém informação sobre taxonomia de mecanismos de composição para desenvolvimento de software embasado em componentes (mecanismos de composição são mecanismos para compor unidades maiores a partir de unidades menores); (c) [Puviani, Cabri e Zambonelli \(2013\)](#) contém informação sobre taxonomia de padrões de arquitetura para sistemas auto-adaptativos, isto é, sistemas com capacidade de autonomamente adaptarem os seus comportamentos e as suas estruturas; (d) [Mehta, Medvidovic e Phadke \(2000\)](#) contém informação sobre taxonomia de conectores entre componentes de software. Essa taxonomia contém tipos de serviço no topo da hierarquia e tipos de conectores no nível hierárquico seguinte; (e) [Svahnberg, Gulp e Bosch \(2005\)](#) contém informação sobre taxonomia de técnicas para implementar variabilidade em software, isto é, capacidade de modificação ou configuração.

A.1.3 Construção de Software

Foram identificadas as seguintes fontes de informação sobre taxonomias na área de conhecimento Construção de Software: (a) [Damasevicius e Stuikys \(2008\)](#) contém informação sobre taxonomia de conceitos em metaprogramação. Nessa taxonomia, conceitos são classificados como conceitos estruturais ou conceitos de processos. A taxonomia foi construída a partir de literatura na área; (b) [Monperrus et al. \(2012\)](#) contém informação sobre taxonomia de tipos de diretivas em documentação de interface de programação de aplicação. No desenvolvimento dessa taxonomia foram estudadas documentações de interfaces de programação de aplicações e a cada diretiva foram associados nome, definição, discussão e exemplo; (c) [Novak, Krajnc e Zontar \(2010\)](#) contém informação sobre taxonomia de ferramentas de análise de código de software; (d) [Shan e Hua \(2006\)](#) contém informação sobre taxonomia de classes de arcabouços para desenvolvimento de aplicações *web*; (e) [Hindle, German e Holt \(2008\)](#) contém informação sobre taxonomia de operações *commit* (submissão de modificação de código a repositório). A estrutura dessa taxonomia adapta a

de taxonomia em [Swanson \(1976\)](#). Em seu desenvolvimento, foram analisados projetos de software aberto.

A.1.4 Manutenção de Software

Foram identificadas as seguintes fontes de informação sobre taxonomias na área de conhecimento Manutenção de Software: (a) [Swanson \(1976\)](#) contém informação sobre taxonomia de causas e escolhas em manutenção de software; (b) [Webster, Oliveira e Anquetil \(2005\)](#) contém informação sobre taxonomia de riscos em manutenção de software. Essa taxonomia apresenta três níveis hierárquicos, segue organização descrita em [Carr et al. \(1993\)](#) e foi construída a partir de informação resultante de revisão de literatura; (c) [Buckley et al. \(2005\)](#) contém informação sobre taxonomia de modificações em software, que aborda mecanismos de modificação e fatores que os influenciam. Nesse contexto, o termo “mecanismo de modificação” designa “ferramentas de software para evolução de software e algoritmos que embasam essas ferramentas”. Essa taxonomia é estruturada em temas e dimensões que caracterizam ou influenciam mecanismos de modificação; (d) [Inpirom e Prompoon \(2013\)](#) contém informação sobre taxonomia de modificações (modificações relacionadas em [Lehnert, Farooq e Riebisch \(2012\)](#)) em três tipos de diagramas na linguagem de modelagem UML (diagrama de caso de uso, diagrama de classe, diagrama de sequência) usados na representação de visões de análise e visões de desenho (*design*) de software; (e) [Lehnert, Farooq e Riebisch \(2012\)](#) contém informação sobre taxonomia de tipos de modificações, que aborda níveis de abstração, tipos de composição, tipos de operação e escopos de modificações. Essa taxonomia é embasada na distinção entre modificações atômicas e modificações compostas. Seu desenvolvimento englobou processo de revisão de literatura; (f) [Fluri e Gall \(2006\)](#) contém informação sobre taxonomia de modificações em código fonte de software, que enfoca códigos escritos em linguagens de programação orientadas a objetos, particularmente, códigos escritos em linguagem Java; (g) [Nurmuliani, Zowghi e Fowell \(2004\)](#) contém informação sobre taxonomia de tipos de mudanças de requisitos, desenvolvida para auxiliar no entendimento de problemas associados à volatilidade de requisitos. O processo de desenvolvimento dessa taxonomia englobou estudo de caso realizado para identificar e entender problemas e causas relacionadas a mudanças de requisitos em desenvolvimento de software.

A.1.5 Gestão de Engenharia de Software

Foram identificadas as seguintes fontes de informação sobre taxonomias na área de conhecimento Gestão de Engenharia de Software: (a) [Gumm \(2006\)](#) contém informação sobre taxonomia de gerência de engenharia de software, construída a partir de literatura, que relaciona dimensões (física, organizacional, temporal, distribuição entre grupos de *stakeholder*) em desenvolvimento distribuído de software; (b) [Gacek e Arief \(2004\)](#) con-

tém informação sobre taxonomia de usuários e de desenvolvedores de software aberto e taxonomia de características de projetos de software aberto. Essas taxonomias foram desenvolvidas por meio da investigação de projetos de software aberto, revisão de literatura, acesso a recursos de projetos de software aberto e entrevistas; (c) [Robillard et al. \(2014\)](#) contém informação sobre taxonomia de processos realizados por equipes (*team process*) em engenharia de software, que engloba fases e episódios em processos. Seu desenvolvimento da taxonomia englobou revisão de literatura.

A.1.6 Processo de Engenharia de Software

Foram identificadas as seguintes fontes de informação sobre taxonomias na área de conhecimento Processo de Engenharia de Software: (a) [Rahman, Sahibuddin e Ibrahim \(2011\)](#) e [Rahman, Sahibuddin e Ibrahim \(2012\)](#) contém informação sobre taxonomia de arcabouços de melhoria de processo de software, desenvolvida para entendimento e comparação de arcabouços. Seu desenvolvimento foi realizado a partir de literatura; (b) [Strode e Huff \(2012\)](#) contém informação sobre taxonomia de tipos de dependências em desenvolvimento ágil de software (*agile software development*). Seu desenvolvimento englobou revisão de literatura e estudo de projetos executados em diferentes classes de organizações.

A.1.7 Qualidade de Software

Foram identificadas as seguintes fontes de informação sobre taxonomias na área de conhecimento Qualidade de Software: (a) [Hill \(2007\)](#) e [Hill e Victor \(2008\)](#) contém informação sobre taxonomia de riscos de segurança de software, que relaciona aspectos relevantes ao desenvolvimento de software crítico em segurança. Os elementos e atributos de segurança são associados à taxonomia de riscos proposta em [Carr et al. \(1993\)](#); (b) [Bosu e Macdonell \(2013\)](#) contém informação sobre taxonomia de desafios de qualidade de dados em engenharia de software empírica (*empirical software engineering*), cujo desenvolvimento englobou revisão de literatura; (c) [Al-Shorafat \(2013\)](#) contém informação sobre taxonomia de atributos de qualidade de segurança (requisitos relacionados à segurança), que engloba requisitos puramente de segurança, requisitos de segurança significativos, requisitos de segurança de sistema e restrições de segurança; (d) [Vegas e Basili \(2005\)](#) contém informação sobre taxonomia (artefato denominado esquema de caracterização pelo seu autor) de técnicas de teste de software, cujo desenvolvimento englobou estudo de literatura, pesquisa com pessoas em diferentes papéis na área de teste e revisão por especialistas; (e) [Roggio, Gordon e Comer \(2013\)](#) contém informação sobre taxonomia de termos em teste de software (definições de termos sobre teste de software), desenvolvida por meio de revisão de literatura; (f) [Utting, Pretschner e Legeard \(2012\)](#) contém informação sobre taxonomia sobre dimensões de teste baseado em modelo (abordagem onde modelos são usados

na produção de casos de teste para o sistema em teste); (g) [Felderer e Schieferdecker \(2014\)](#) contém informação sobre taxonomia de teste baseado em risco (abordagem que usa avaliações de risco para guiar fases do processo de teste, melhorar esforços de teste e limitar riscos de sistemas de software). Essa taxonomia tem o objetivo de contribuir para o entendimento, categorização, avaliação e comparação de abordagens de teste. Foi desenvolvida a partir de publicações sobre teste baseado em risco; (h) [Delgado, Gates e Roach \(2004\)](#) contém informação sobre taxonomia de ferramentas para monitorar faltas em software (ferramentas para determinar se comportamentos são consistentes com especificações) em tempo de execução (*runtime*). Seu desenvolvimento englobou revisão de literatura sobre monitoração em tempo de execução; (i) [Mala e Çil \(2011\)](#) contém informação sobre taxonomia para medir complexidade em sistemas baseados em agentes de software, o desenvolvimento dessa taxonomia englobou consulta à literatura; (j) [Ali, Hosking e Grundy \(2013\)](#) contém informação sobre taxonomia de abordagens de crítica suportadas por computador, cujo desenvolvimento englobou revisão de literatura.

A.1.8 Diversas áreas de conhecimento em Engenharia de Software

Foram identificadas as seguintes fontes de informação sobre taxonomias que englobam conceitos em diversas áreas de conhecimento da disciplina Engenharia de Software: (a) [Meier e Cahill \(2005\)](#) contém informação sobre taxonomia de sistemas distribuídos de programação que usam modelo de comunicação baseado em eventos. Nessa taxonomia, são relacionadas propriedades funcionais e não funcionais desses sistemas; (b) [Roongkaew e Prompoon \(2013\)](#) contém informação sobre taxonomia de ferramentas em engenharia de software. Nessa taxonomia, classes foram definidas segundo áreas de conhecimento descritas em [IEEE \(2014c\)](#); (c) [Forward e Lethbridge \(2008\)](#) contém informação sobre taxonomia de classes de software, cujo desenvolvimento englobou seleção de classes a partir de taxonomias existentes e de outras fontes de informação, definição de critério de desenho (*design*) e processo de refino da taxonomia; (d) [Taivalsaari e Mikkonen \(2011\)](#) contém informação sobre taxonomia de software na *Web*; (e) [Carvallo et al. \(2004\)](#) contém informação sobre taxonomia de aplicações de negócio; (f) [Sulistio, Yeo e Buyya \(2004\)](#) contém informação sobre taxonomia para desenho (*design*) de simulações baseadas em computadores, composta por taxonomia de sistemas distribuídos e paralelos, por taxonomia de uso, por taxonomia de simulação e por taxonomia de desenho (*design*); (g) [Carr et al. \(1993\)](#) contém informação sobre taxonomia de desenvolvimento de software e taxonomia de riscos de desenvolvimento de software.

A.2 Ontologias em Engenharia de Software

Diversos vocabulários controlados classificados como ontologias pelos seus autores foram identificados no processo de mapeamento sistemático de literatura realizado. As

ontologias identificadas abordam áreas de conhecimento da Engenharia de Software cujos nomes são listados a seguir.

- (a) Requisitos de software;
- (b) Desenho (*design*) de Software;
- (c) Construção de Software;
- (d) Teste de Software;
- (e) Manutenção de Software;
- (f) Gestão de Engenharia de Software;
- (g) Processo de Engenharia de Software;
- (h) Modelos e Métodos de Engenharia de Software;
- (i) Qualidade de Software.

A.2.1 Requisitos de Software

Foram identificadas as seguintes fontes de informação sobre ontologias na área de conhecimento Requisitos de Software: (a) [Nardi e Falbo \(2006\)](#) contém informação sobre ontologia composta por conceitos e relações no domínio requisitos de software, desenvolvida com o objetivo de facilitar a comunicação entre pessoas, entre agentes de software e prover suporte ao desenvolvimento de ferramentas para apoio à Engenharia de Requisitos; (b) [Riechert et al. \(2007\)](#) contém informação sobre ontologia para engenharia de requisitos denominada *SoftWiki Ontology for Requirements Engineering* (SWORE), cujo objetivo é prover suporte à colaboração de interessados (*stakeholder*) em desenvolvimento de software, particularmente, em engenharia de requisitos.

A.2.2 Desenho (*design*) de Software

Foram identificadas as seguintes fontes de informação sobre ontologias na área de conhecimento Desenho (*design*) de Software: (a) [Garzas e Piattini \(2005\)](#) contém informação sobre ontologia de desenho (*design*) de microarquitetura de sistemas de software orientados a objetos, cujo objetivo é estruturar e unificar conhecimento sobre desenho (*design*) de microarquitetura de sistemas de software orientados a objetos; (b) [Garzas e Piattini \(2007\)](#) contém informação sobre ontologia de conceitos em desenho (*design*) orientado a objetos, cujo desenvolvimento seguiu processo descrito em [Kishore, Zhang e Ramesh \(2004\)](#); (c) [Kang e Liang \(2013\)](#) contém informação sobre ontologia de segurança, desenvolvida com o objetivo de promover suporte à identificação, desenho (*design*) e codificação de

aspectos de segurança em software; (d) [López et al. \(2012\)](#) contém informação sobre duas ontologias compostas por conceitos em arquitetura de software. A primeira ontologia tem o objetivo de representar arquitetura de software e a segunda ontologia tem o objetivo de descrever razões (*rationale*) de projeto de arquitetura de software; (e) [Babu et al. \(2007\)](#) contém informação sobre ontologia de arquitetura de software, cujo objetivo é auxiliar o entendimento de práticas em arquitetura de software; (f) [Talevski, Wongthongtham e Komchaliaw \(2008\)](#) contém informação sobre ontologia composta por conceitos, abstrações, relações e interações em engenharia de componentes de software; (g) [Akerman e Tyree \(2006\)](#) contém informação sobre ontologia de ativos de arquitetura, decisões de arquitetura, preocupações (*concern*) de interessados (*stakeholder*) e roteiros de arquitetura de software, cujo objetivo é prover suporte ao desenvolvimento de arquitetura de software.

A.2.3 Construção de Software

Foram identificadas as seguintes fontes de informação sobre ontologias na área de conhecimento Construção de Software: (a) [Leite, Girardi e Cavalcante \(2008\)](#) contém informação sobre ontologias composta por conceitos em Multi-agent Domain Engineering Methodology (MADEM) e Multi-agent Application Engineering Methodology (MAAEN). Essas ontologias têm o objetivo de prover suporte ao desenvolvimento de sistemas multi-agente; (b) [Using, Ahmad e Taib \(2010\)](#) contém informação sobre ontologia para representação de recursos de programação, cujo desenvolvimento englobou uso de questionário, realização de entrevistas e estudo de referências. Essa ontologia visa auxiliar estudantes e instrutores na recuperação da informação e materiais relacionados à programação; (c) [Mukhtar et al. \(2013\)](#) contém informação sobre ontologia desenvolvida para suporte a processo de seleção de ferramentas de desenvolvimento, a partir de opiniões de praticantes e pesquisadores em Engenharia de Software.

A.2.4 Teste de Software

Foram identificadas as seguintes fontes de informação sobre ontologias na área de conhecimento Teste de Software: (a) [Barbosa, Nakagawa e Maldonado \(2006\)](#) contém informação sobre ontologia de conceitos sobre teste de software, cujo desenvolvimento se baseou no padrão ISO/IEC 12207. Essa ontologia visa prover suporte à aquisição, reuso e compartilhamento de conhecimento sobre teste; (b) [Souza, Falbo e Vijaykumar \(2013\)](#) contém informação sobre ontologia composta por conceitos em teste de software. No desenvolvimento dessa ontologia foram usados padrões de ontologia na *Software Process Ontology Pattern Language* (SP-OPL). Essa ontologia foi desenvolvida com o objetivo de auxiliar processos de comunicação, integração, busca e representação de conhecimento sobre teste de software.

A.2.5 Manutenção de Software

Sobre a área de conhecimento Manutenção de Software, foi identificada a fonte de informação [Anquetil, Oliveira e Dias \(2006\)](#) sobre ontologia composta por conceitos relativos à prática de manutenção de software. Essa ontologia é dividida em cinco subontologias, Em seu desenvolvimento, definição de escopo e definição de conceitos foram realizadas a partir de literatura e de conhecimento dos autores da ontologia.

A.2.6 Gestão de Engenharia de Software

Foram identificadas as seguintes fontes de informação sobre ontologias na área de conhecimento Gestão de Engenharia de Software: (a) [Hamdan et al. \(2006\)](#) contém informação sobre ontologia composta por conceitos em projetos de software, desenvolvida para auxiliar na estimativa de custos e esforços em projetos; (b) [Marques et al. \(2013\)](#) contém informação sobre ontologia de domínio que engloba conceitos relacionados à alocação de tarefas em desenvolvimento distribuído de software. Essa ontologia foi desenvolvida a partir de mapeamento da literatura e de opiniões de especialistas, e tem o objetivo de promover o entendimento de conceitos relacionados à alocação de tarefas em equipes distribuídas e prover suporte a gerentes de projetos e a pesquisadores no desenvolvimento de propostas para alocação de tarefas. Em seu desenvolvimento foi usado o processo *Systematic Approach for Building Ontologies* (SABiO) proposto em [Falbo \(2004\)](#); (c) [Botzenhardt, Maedche e Wiesner \(2011\)](#) contém informação sobre ontologia composta por conceitos em gestão de produtos de software, desenvolvida para promover entendimento do domínio “gestão de produto de software”; (d) [Rungratri e Usanavasin \(2012\)](#) contém informação sobre ontologia composta por conceitos usados na descrição de ativos de projetos, e identificados a partir de conjunto de elementos em Dublin Core e propriedades em descrições de documentos; (e) [Sherman \(2009\)](#) contém informação sobre ontologia compostas por conceitos em projeto de software, cujo objetivo é promover a transferência de conhecimentos entre participantes de projetos de software.

A.2.7 Processo de Engenharia de Software

Foram identificadas as seguintes fontes de informação sobre ontologias na área de conhecimento Processo de Engenharia de Software: (a) [Bertoa, Vallecillo e García \(2006\)](#) e [García et al. \(2006\)](#) contém informação sobre ontologia composta por conceitos em medição de software, cujo objetivo é contribuir na harmonização de propostas e padrões em medição de software. Essa ontologia foi desenvolvida a partir de padrões internacionais e de resultados de pesquisas sobre medição de software; (b) [Simmons e Dillon \(2006\)](#) contém informação sobre ontologia composta por conceitos em processo de desenvolvimento de software aberto, cujo objetivo é promover a organização da informação em desenvolvimento de software aberto, particularmente, a organização da informação em repositórios; (c)

Barcellos e Falbo (2013) contém informação sobre ontologia de tarefa em processo de medição de software, cujo objetivo é auxiliar na solução de problemas decorrentes do uso conjunto de diferentes padrões relacionados a medições e prover suporte à integração semântica de aplicações de software que suportem medições. Essa ontologia é embasada na *Unified Foundational Ontology* (UFO); (d) Falbo e Bertollo (2005) contém informação sobre ontologia de processos de software, que estende a ontologia descrita em Falbo, Menezes e Rocha (1998) e tem o objetivo de facilitar entendimento e comunicação sobre processos de software em organizações; (e) Vizcaino et al. (2012) contém informação sobre ontologia de desenvolvimento global de software, essa ontologia tem o objetivo de auxiliar gerentes de projeto a melhor descreverem projetos globais de desenvolvimento de software; (f) Mirbel (2009) contém informação sobre ontologia desenvolvida para contribuir na coordenação e no compartilhamento de conhecimento em desenvolvimento de software aberto. Seu desenvolvimento adotou processo descendente e foi iniciado a partir de conceitos em *O'CoP*, ontologia composta por conceitos em comunidades de prática; (g) Liao e Leung (2014) contém informação sobre ontologia que tem o objetivo de definir a estrutura de modelos de processos de software; (h) Chaves et al. (2011) contém informação sobre ontologia composta por conceitos relacionados a ações, contextos de indivíduos, locais e ferramentas em desenvolvimento distribuído de software. Essa ontologia foi desenvolvida para minimizar ambiguidades e prover suporte à comunicação entre desenvolvedores geograficamente distribuídos; (i) Rocha et al. (2013) contém informação sobre ontologia desenvolvida para auxiliar projetos distribuídos de software por meio de um vocabulário comum para equipes distribuídas. Em seu desenvolvimento ontologia foi usada a linguagem OWL; (j) Soydan e Kokar (2006) contém informação sobre ontologia de conceitos no modelo CMMI-SW, codificada em OWL e desenvolvida via abordagem descendente; (k) Rungratri e Usanavasin (2012) contém informação sobre ontologia de conceitos no modelo CMMI v.1.3, essa ontologia foi desenvolvida para prover suporte à análise de diferenças entre situação da organização e modelo CMMI (*gap analysis*); (l) Hilera e Fernández-Sanz (2010) contém informação sobre ontologia composta por conceitos no modelo CMMI. No desenvolvimento dessa ontologia foram identificados conceitos no corpo de conhecimento de *CMMI Systems Engineering/Software Engineering*. Essa ontologia pode ser usada em processo de certificação CMMI.

A.2.8 Modelos e Métodos de Engenharia de Software

Sobre a área de conhecimento Modelos e Métodos de Engenharia de Software, Bagiampou e Kameas (2012) contém informação sobre ontologia de diagramas de casos de uso, desenvolvida com o objetivo de prover referência comum a desenvolvedores de material educacional e de material de treinamento para cursos em Engenharia de software. Essa ontologia pode prover suporte ao ensino de análise orientada a objetos ou servir como referência de termos em diagramas de casos de uso.

A.2.9 Qualidade de Software

Foram identificadas as seguintes fontes de informação sobre ontologias na área de conhecimento Qualidade de software: (a) [Kayed et al. \(2009\)](#) contém informação sobre ontologia de atributos de qualidade de produtos de software, cujo objetivo é promover entendimento sobre esses atributos. Em seu desenvolvimento, conceitos foram identificados por meio de revisão de literatura sobre medições, atributos e qualidade de software; (b) [Moro e Falbo \(2008\)](#) contém informação sobre ontologia composta por conceitos no domínio Qualidade de Software, enfoca aspectos comuns às perspectivas de produto e processo de software; (c) [Duarte e Falbo \(2000\)](#) contém informação sobre ontologia composta por conceitos no domínio Qualidade de Software, desenvolvida para promover compreensão nesse domínio.

A.2.10 Diversas áreas de conhecimento em Engenharia de Software

Foram identificadas as seguintes fontes de informação sobre ontologias que englobam conceitos em diversas áreas de conhecimento da disciplina Engenharia de Software: (a) [Sicilia, Cuadrado e Rodriguez \(2005\)](#) contém informação sobre ontologia de artefatos e de atividades identificadas em *SWEBOK*, cujo objetivo é promover representação de artefatos e atividades em Engenharia de Software; (b) [Sun, Miao e Cao \(2006\)](#) contém informação sobre ontologia composta por conceitos no domínio da Ciência da Computação, cujo objetivo é promover suporte à recuperação de componentes de software; (c) [Wongthongtham et al. \(2008\)](#) e [Wu \(2012\)](#) contêm informação sobre ontologia composta por conceitos em Engenharia de Software, desenvolvida a partir de livro texto em Engenharia de Software e do *SWEBOK*. Seu objetivo é facilitar a captura de conhecimento em Engenharia de Software, o compartilhamento desse conhecimento em projeto de desenvolvimento geograficamente distribuído de software e a troca de dados sobre projetos de software; (d) [Hilera e Fernández-Sanz \(2010\)](#) contém informação sobre ontologia de domínio desenvolvida a partir do *IEEE Glossary of Software Engineering*, com o objetivo de promover a aquisição de conhecimento em Engenharia de Software e facilitar a localização de recursos; (e) [Valaski et al. \(2011\)](#) contém informação sobre ontologia desenvolvida com o objetivo de facilitar a recuperação de material de aprendizado sobre Engenharia de Software. Essa ontologia é composta por conceitos identificados em *SWEBOK* e em *Rational Unified Process* (RUP) e apresenta estrutura hierárquica embasada nas áreas de conhecimento descritas no *SWEBOK*; (f) [González-Pérez e Henderson-Sellers \(2006\)](#) contém informação sobre ontologia composta por conceitos sobre empreendimentos de desenvolvimento de software; (g) [Nunes \(2005\)](#) contém informação sobre ontologia de artefatos de software, cujo objetivo é promover compartilhamento de artefatos, comunicação entre desenvolvedores e comunicação entre ferramentas em desenvolvimento de software; (h) [Wille et al. \(2004\)](#) contém informação sobre estrutura de ontologia embasada no guia *SWEBOK*.