

Fernando Antônio de Araújo Chacon de Albuquerque

**Modelo de *Framework* de Arquitetura da  
Informação Baseado em Roteiros**

Brasília

2014

Fernando Antônio de Araújo Chacon de Albuquerque

## **Modelo de *Framework* de Arquitetura da Informação Baseado em Roteiros**

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

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

Orientador: Prof. Dr. Mamede Lima-Marques

Brasília

2014

---

Fernando Antônio de Araújo Chacon de Albuquerque  
Modelo de *Framework* de Arquitetura da Informação Baseado em Roteiros /  
Fernando Antônio de Araújo Chacon de Albuquerque. – Brasília, 2014-  
145 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Mamede Lima-Marques

Dissertação (Mestrado) – Universidade de Brasília – UnB  
Faculdade de Ciência da Informação  
Programa de Pós-Graduação, 2014.

I. Framework de arquitetura da informação. II. Organização da informação.  
III. Processo de desenvolvimento de software. IV. Mamede Lima-Marques. V.  
Universidade de Brasília. VI. Faculdade de Ciência da Informação. VII. Modelo de  
Framework de Arquitetura da Informação Baseado em Roteiros.

CDU 02:141:005.7

---

## FOLHA DE APROVAÇÃO

**Título:** “Modelo de framework de Arquitetura da Informação baseado em roteiros”.

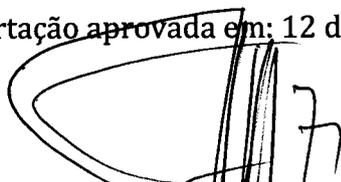
**Autor (a):** Fernando Antonio Araújo Chacon de Albuquerque

**Área de concentração:** Gestão da informação

**Linha de pesquisa:** Organização da Informação

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

Dissertação aprovada em: 12 de Fevereiro de 2014.



---

**Prof. Dr. Mamede Lima Marques**  
Presidente (UnB/PPGCINF)



---

**Prof. Dr. André Porto Ancona Lopez**  
Membro Interno (UnB/PPGCINF)

---

**Prof. Dr. Rildo Ribeiro dos Santos**  
Membro Externo (FACE)



---

**Prof. Dr. Romualdo Alves Pereira Júnior**  
Suplente - (AEB)

*Aos meus familiares.*

# Agradecimentos

Ao Prof. Dr. Mamede Lima-Marques pelo ensino, orientação e incentivo.

À Dra. Cláudia Augusto Dias pelo auxílio na revisão e sugestões apresentadas.

Aos professores da Faculdade de Ciência da Informação pelo ensino.

Aos membros da banca pelas sugestões apresentadas.

Às funcionárias Jucilene Gomes e Martha Araújo pelo apoio ao longo do curso.

Ao colega Paulo Argolo da Cruz Rios Filho pelo apoio ao longo do curso.

Aos meus familiares pela paciência.

# Resumo

A pesquisa relatada nesta dissertação investigou o desenvolvimento e uso de um modelo de *framework* de arquitetura da informação. O modelo de *framework* foi desenvolvido a partir de informações em The Open Group Architecture Framework (TOGAF), Project Management Body of Knowledge (PMBOK) Guide, Método de Arquitetura da Informação Aplicada (MAIA) e em outras fontes. Esta dissertação inclui resultados de pesquisa bibliográfica acerca de arquitetura da informação, *frameworks* e métodos de arquitetura da informação, sistemas para organização da informação, *frameworks* de gestão de projetos, processos de desenvolvimento de *software*; inclui proposta de modelo de *framework* de arquitetura da informação; inclui resultado do uso do modelo de *framework* proposto na descrição da arquitetura da informação em uma comunidade voltada ao desenvolvimento de *software* aberto.

**Palavras-chaves:** *Framework* de arquitetura da informação. Organização da informação. Processo de desenvolvimento de *software*.

# Abstract

The research reported in this dissertation has investigated the development and use of an information architecture framework model. The framework model was developed employing information from The Open Group Architecture Framework (TOGAF), Project Management Body of Knowledge (PMBOK) Guide, Método de Arquitetura da Informação Aplicada (MAIA) and other sources. This dissertation includes the results of a bibliographic research on information architecture, information architecture frameworks and methods, information organization systems, project management frameworks, software development processes; includes the proposal of a framework model for information architecture; includes the results of a case study in which the proposed framework model was employed to partially describe the information architecture in a community engaged in open source software development.

**Key-words:** Information architecture framework. Organization of information. Software development process.

# Lista de ilustrações

|  |     |
|--|-----|
| Figura 1 – Relacionamentos entre conceitos abordados na pesquisa bibliográfica. . . . .    | 27  |
| Figura 2 – Contexto de descrição de arquitetura segundo ISO (2011). . . . .                | 45  |
| Figura 3 – Modelo conceitual de descrição de arquitetura segundo ISO (2011). . . . .       | 46  |
| Figura 4 – Fase com iterações seriais e concorrentes. . . . .                              | 64  |
| Figura 5 – Fluxo de atividades em um processo. . . . .                                     | 64  |
| Figura 6 – Níveis de atividade de processos ao longo do ciclo de vida. . . . .             | 65  |
| Figura 7 – Contexto de projeto de descrição de arquitetura. . . . .                        | 82  |
| Figura 8 – Relacionamentos entre conceitos em projeto de descrição de arquitetura. . . . . | 83  |
| Figura 9 – Exemplo de roteiro desenvolvido. . . . .  | 90  |
| Figura 10 –Página inicial do portal do sítio Ohloh (OHLOH, 2013). . . . .                  | 96  |
| Figura 11 –Página inicial do portal do projeto OpenStack (OPENSTACK, 2013). . . . .        | 97  |
| Figura 12 –Exemplo de descrição de princípio. . . . .                                      | 101 |
| Figura 13 –Exemplo de descrição de papel. . . . .  | 102 |
| Figura 14 –Exemplo de descrição de serviço de negócio. . . . .                             | 102 |
| Figura 15 –Exemplo de descrição de função de negócio. . . . .                              | 103 |
| Figura 16 –Exemplo de descrição de localização. . . . .                                    | 103 |
| Figura 17 –Exemplo de diagrama do modelo de caso de uso. . . . .                           | 105 |
| Figura 18 –Exemplo de descrição de caso de uso. . . . .                                    | 105 |
| Figura 19 –Ferramenta para coleta de metadados. . . . .                                    | 106 |
| Figura 20 –Estrutura da planilha empregada na auditoria. . . . .                           | 107 |
| Figura 21 –Planilha com alguns poucos elementos da taxonomia. . . . .                      | 108 |
| Figura 22 –Roteiro Desenvolver termo de abertura do projeto. . . . .                       | 125 |
| Figura 23 –Roteiro Desenvolver registro das partes interessadas. . . . .                   | 126 |
| Figura 24 –Roteiro Desenvolver plano de gerenciamento do projeto. . . . .                  | 127 |
| Figura 25 –Roteiro Desenvolver documentação dos requisitos. . . . .                        | 128 |
| Figura 26 –Roteiro Desenvolver enunciado do escopo do projeto. . . . .                     | 128 |
| Figura 27 –Roteiro Desenvolver lista de marcos. . . . .                                    | 129 |
| Figura 28 –Roteiro Desenvolver lista de atividades. . . . .                                | 130 |
| Figura 29 –Roteiro Desenvolver diagrama de sequência de atividades. . . . .                | 130 |
| Figura 30 –Roteiro Desenvolver cronograma. . . . .   | 131 |
| Figura 31 –Roteiro Descrever princípios de arquitetura. . . . .                            | 132 |
| Figura 32 –Roteiro Descrever configuração do <i>framework</i> de arquitetura. . . . .      | 133 |

|   |     |
|---|-----|
| Figura 33 –Roteiro Descrever atores e papéis. . . . .                               | 133 |
| Figura 34 –Roteiro Descrever serviços e funções de negócio. . . . .                 | 134 |
| Figura 35 –Roteiro Descrever localizações. . . . .                                  | 134 |
| Figura 36 –Roteiro Desenvolver modelo de casos de uso de negócio. . . . .           | 135 |
| Figura 37 –Roteiro Descrever <i>business footprint</i> . . . . .                    | 136 |
| Figura 38 –Roteiro Descrever ciclo de vida do produto. . . . .                      | 136 |
| Figura 39 –Roteiro Descrever composição da organização. . . . .                     | 137 |
| Figura 40 –Roteiro Descrever processos do negócio. . . . .                          | 138 |
| Figura 41 –Roteiro Desenvolver taxonomia de termos e conceitos. . . . .             | 139 |
| Figura 42 –Roteiro Desenvolver mapa de conceitos. . . . .                           | 140 |
| Figura 43 –Roteiro Desenvolver ontologia de conceitos. . . . .                      | 141 |
| Figura 44 –Distribuição dos documentos referenciados por ano de publicação. . . . . | 143 |
| Figura 45 –Documentos por ano de publicação (Fonte: Web of Science). . . . .        | 144 |
| Figura 46 –Documentos por áreas de pesquisa (Fonte: Web of Science). . . . .        | 144 |
| Figura 47 –Documentos por ano de publicação (Fonte: Web of Science). . . . .        | 144 |
| Figura 48 –Documentos por áreas de pesquisa (Fonte: Web of Science). . . . .        | 145 |

# Lista de tabelas

|           |  |     |
|-----------|--|-----|
| Tabela 1  | – Fases e processos de gerenciamento do modelo de <i>framework</i> proposto. . . | 85  |
| Tabela 2  | – Fases e processos de arquitetura do modelo de <i>framework</i> proposto. . .   | 85  |
| Tabela 3  | – Fases e passos analisados em MAIA. . . . .                                     | 86  |
| Tabela 4  | – Fases e passos analisados em TOGAF. . . . .                                    | 86  |
| Tabela 5  | – Fases e artefatos de gerenciamento do modelo de <i>framework</i> proposto. .   | 87  |
| Tabela 6  | – Fases e artefatos de arquitetura do modelo de <i>framework</i> proposto. . .   | 87  |
| Tabela 7  | – Fases e artefatos analisados em MAIA. . . . .                                  | 88  |
| Tabela 8  | – Fases e artefatos analisados em TOGAF. . . . .                                 | 88  |
| Tabela 9  | – Fontes de componentes do modelo de <i>framework</i> proposto. . . . .          | 91  |
| Tabela 10 | – Fontes dos processos de gerenciamento do modelo de <i>framework</i> proposto.  | 91  |
| Tabela 11 | – Fontes dos processos de arquitetura do modelo de <i>framework</i> proposto.    | 91  |
| Tabela 12 | – Fontes dos artefatos de gerenciamento do modelo de <i>framework</i> proposto.  | 92  |
| Tabela 13 | – Fontes dos artefatos de arquitetura do modelo de <i>framework</i> proposto.    | 92  |
| Tabela 14 | – Processos e artefatos na fase Iniciação. . . . .                               | 97  |
| Tabela 15 | – Processos e artefatos na fase Organização e preparação. . . . .                | 98  |
| Tabela 16 | – Processos e artefatos na fase Execução do trabalho. . . . .                    | 98  |
| Tabela 17 | – Processos e artefatos na fase Encerramento. . . . .                            | 98  |
| Tabela 18 | – Desenvolvimento de <i>software</i> aberto segundo Lonchamp (2005). . . . .     | 100 |
| Tabela 19 | – Endereços de documentos acessados. . . . .                                     | 100 |
| Tabela 20 | – Nomes e descrições dos atores. . . . .   | 104 |
| Tabela 21 | – Relações entre atores e casos de uso. . . . .                                  | 104 |

# Lista de abreviaturas e siglas

|         |   |
|---------|---|
| ABNT    | Associação Brasileira de Normas Técnicas                      |
| ACF     | <i>Architecture Content Framework</i>                         |
| ACM     | <i>Association for Computing Machinery</i>                    |
| ADM     | <i>Architecture Development Method</i>                        |
| AF-EAF  | <i>Air Force Enterprise Architecture Framework</i>            |
| AIO     | Arquitetura da Informação Organizacional                      |
| BCE     | Biblioteca Central da Universidade de Brasília                |
| BDBComp | Biblioteca Digital Brasileira de Computação                   |
| BPEAM   | <i>Best-Practice Enterprise Architecture Method</i>           |
| BSI     | <i>British Standards Institution</i>                          |
| CAPES   | Coordenação de Aperfeiçoamento de Pessoal de Nível superior   |
| CDWA    | <i>Categories for Description of Works of Art</i>             |
| CIAF    | <i>Capgemini Integrated Architecture Framework</i>            |
| CMMI    | <i>Capability Maturity Model Integration</i>                  |
| CNPq    | Conselho Nacional de Desenvolvimento Científico e Tecnológico |
| CPAI    | Centro de Pesquisa em Arquitetura da Informação               |
| DARPA   | <i>Defense Advanced Research Projects Agency</i>              |
| DSDM    | <i>Dynamic Systems Development Method</i>                     |
| DTF     | <i>Department of Treasury and Finance</i>                     |
| E2AF    | <i>Extended Enterprise Architecture Framework</i>             |
| EAB     | <i>Enterprise Architecture Blueprinting</i>                   |
| EUA     | Estados Unidos da América                                     |
| EUP     | <i>Enterprise Unified Process</i>                             |

|       |  |
|-------|--|
| FEA   | <i>Federal Enterprise Architecture</i>                     |
| FEAF  | <i>Federal Enterprise Architecture Framework</i>           |
| GEAF  | <i>Gartner's Enterprise Architecture Framework</i>         |
| GPL   | <i>General Public License</i>                              |
| HEAF  | <i>Health Enterprise Architecture Framework</i>            |
| HTML  | <i>HyperText Markup Language</i>                           |
| IBGE  | Instituto Brasileiro de Geografia e Estatística            |
| IBICT | Instituto Brasileiro de Informação em Ciência e Tecnologia |
| IEC   | International Electrotechnical Commission                  |
| IEEE  | <i>Institute of Electrical and Electronics Engineers</i>   |
| IFLA  | <i>International Federation of Library Associations</i>    |
| IFW   | <i>IBM Information Framework</i>                           |
| INPI  | Instituto Nacional de Propriedade Industrial               |
| IRC   | <i>Internet Relay Chat</i>                                 |
| ISACA | <i>Information Systems Audit and Control Association</i>   |
| ISO   | <i>International Organization for Standardization</i>      |
| ITIL  | <i>Information Technology Infrastructure Library</i>       |
| JCR   | <i>Journal Citation Reports</i>                            |
| LOM   | <i>Learning Object Metadata</i>                            |
| MAIA  | Método de Arquitetura da Informação Aplicada               |
| MARC  | <i>Machine Readable Cataloging</i>                         |
| METS  | <i>Metadata Encoding and Transmission Standard</i>         |
| MODS  | <i>Metadata Object Description Schema</i>                  |
| MPEG  | <i>Moving Picture Expert Group</i>                         |
| NISO  | <i>National Information Standards Organization</i>         |
| OADP  | <i>Oracle Architecture Development Process</i>             |

|         |  |
|---------|--|
| OEAF    | <i>Oracle Enterprise Architecture Framework</i>                    |
| OGC     | <i>Office of Government Commerce</i>                               |
| OMG     | <i>Object Management Group</i>                                     |
| OMT     | <i>Object Modeling Technique</i>                                   |
| ONIX    | <i>Online Information Exchange</i>                                 |
| OOSE    | <i>Object Oriented Software Engineering</i>                        |
| OTAN    | Organização do Tratado do Atlântico Norte                          |
| PDF     | <i>Portable Document Format</i>                                    |
| PEAF    | <i>Pragmatic Enterprise Architecture Framework</i>                 |
| PMBOK   | <i>Project Management Body of Knowledge</i>                        |
| PMI     | <i>Project Management Institute</i>                                |
| PRINCE2 | <i>Projects in Controlled Environments, version 2</i>              |
| SAA     | <i>Society of American Archivists</i>                              |
| SBU     | Sistema de Bibliotecas da Unicamp                                  |
| SciELO  | <i>Scientific Electronic Library Online</i>                        |
| SEI     | <i>Software Engineering Institute</i>                              |
| SI      | Sistemas de Informação   |
| SOAP    | <i>Simple Object Access Protocol</i>                               |
| TAFIM   | <i>Technical Architecture Framework for Information Management</i> |
| TEAF    | <i>Treasury Enterprise Architecture Framework</i>                  |
| TEI     | <i>Text Encoding Initiative</i>                                    |
| TI      | Tecnologia da Informação   |
| TOGAF   | <i>The Open Group Architecture Framework</i>                       |
| PMBOK   | <i>Project Management Body of Knowledge</i>                        |
| RUP     | <i>Rational Unified Process</i>                                    |
| UDDI    | <i>Universal Description Discovery and Integration</i>             |

|       |  |
|-------|--|
| UML   | <i>Unified Modeling Language</i>                 |
| UnB   | Universidade de Brasília                         |
| URL   | <i>Uniform Resource Locator</i>                  |
| USDP  | <i>Unified Software Development Process</i>      |
| USPTO | <i>United States Patent and Trademark Office</i> |
| W3C   | <i>World Wide Web Consortium</i>                 |
| WSDL  | <i>Web Services Description Language</i>         |
| XP    | <i>Extreme Programming</i>                       |
| ZF    | <i>Zachman Framework</i>                         |

# Sumário

|   |           |
|---|-----------|
| <b>Introdução</b> . . . . .   | <b>19</b> |
| <b>I Preparação da pesquisa</b>   | <b>21</b> |
| <b>1 Preparação da pesquisa</b> . . . . .                               | <b>22</b> |
| 1.1 Problema de pesquisa . . . . .                                      | 22        |
| 1.2 Objetivos . . . . .   | 22        |
| 1.2.1 Objetivo geral . . . . .  | 22        |
| 1.2.2 Objetivos específicos . . . . .                                   | 22        |
| 1.3 Motivações e justificativas para a realização da pesquisa . . . . . | 23        |
| 1.4 Metodologia . . . . .   | 24        |
| 1.4.1 Classificação da pesquisa . . . . .                               | 24        |
| 1.4.2 Percurso metodológico . . . . .                                   | 24        |
| <b>II Referenciais teóricos</b>   | <b>26</b> |
| <b>Prólogo</b> . . . . .  | <b>27</b> |
| <b>2 Arquitetura da informação</b> . . . . .                            | <b>28</b> |
| 2.1 Conceitos acerca de arquitetura de sistema . . . . .                | 28        |
| 2.2 Conceitos acerca de arquitetura da informação . . . . .             | 28        |
| <b>3 Frameworks e métodos de arquitetura da informação</b> . . . . .    | <b>32</b> |
| 3.1 Conceitos acerca de <i>framework</i> de arquitetura . . . . .       | 32        |
| 3.2 <i>Frameworks</i> de arquitetura da informação . . . . .            | 32        |
| 3.2.1 <i>Framework for Enterprise Architecture</i> . . . . .            | 33        |
| 3.2.2 <i>The Open Group Architecture Framework</i> . . . . .            | 34        |
| 3.2.3 <i>EA3 Cube Framework</i> . . . . .                               | 34        |
| 3.3 Métodos de arquitetura da informação . . . . .                      | 35        |
| 3.3.1 Método descrito em Morville e Rosenfeld (2006) . . . . .          | 35        |
| 3.3.2 Método descrito em Spencer e Featherstone (2010) . . . . .        | 36        |
| 3.3.2.1 Inventário e auditoria de conteúdos . . . . .                   | 37        |
| 3.3.3 Método descrito em DTF (2010) . . . . .                           | 37        |
| 3.3.4 Método descrito em <i>Princeton University (2008)</i> . . . . .   | 39        |
| 3.3.5 Método de Arquitetura da Informação Aplicada . . . . .            | 39        |

|          |  |           |
|----------|--|-----------|
| 3.3.6    | Método de Arquitetura Ágil da Informação Organizacional . . . . .      | 41        |
| 3.3.7    | TOGAF Architecture Definition Method . . . . .                         | 41        |
| 3.3.8    | Método descrito em Bernard (2012) . . . . .                            | 42        |
| <b>4</b> | <b>Modelagem de arquitetura da informação . . . . .</b>                | <b>43</b> |
| 4.1      | Descrição de arquitetura de sistema . . . . .                          | 43        |
| 4.2      | Ponto de vista e visão de arquitetura de sistema . . . . .             | 45        |
| 4.3      | Modelo de arquitetura . . . . .  | 48        |
| 4.4      | Linguagens de modelagem . . . . .                                      | 48        |
| 4.4.1    | <i>Unified Modeling Language</i> . . . . .                             | 49        |
| 4.4.2    | Linguagem ArchiMate . . . . .  | 50        |
| <b>5</b> | <b>Sistemas para organização da informação . . . . .</b>               | <b>52</b> |
| 5.1      | Conceitos acerca de organização da informação . . . . .                | 52        |
| 5.1.1    | Indexação, catalogação e classificação . . . . .                       | 52        |
| 5.1.2    | Metadados . . . . .  | 53        |
| 5.2      | Sistemas para organização da informação . . . . .                      | 54        |
| 5.2.1    | Vocabulário controlado . . . . .                                       | 55        |
| 5.2.1.1  | Lista de termos . . . . .  | 56        |
| 5.2.1.2  | Anel de sinônimos . . . . .  | 57        |
| 5.2.1.3  | Taxonomia hierárquica . . . . .  | 57        |
| 5.2.1.4  | Tesauro . . . . .  | 58        |
| 5.2.2    | Ontologia . . . . .  | 58        |
| 5.2.2.1  | Desenvolvimento de ontologias . . . . .                                | 60        |
| 5.2.2.2  | Ontologias na engenharia de <i>software</i> . . . . .                  | 61        |
| 5.2.3    | Classificação social . . . . .   | 61        |
| <b>6</b> | <b><i>Frameworks</i> de gestão de projetos . . . . .</b>               | <b>63</b> |
| 6.1      | Projeto . . . . .  | 63        |
| 6.2      | Ciclo de vida de projeto . . . . .                                     | 63        |
| 6.3      | Processo . . . . .   | 64        |
| 6.3.1    | Processo de negócio . . . . .  | 65        |
| 6.4      | <i>Frameworks</i> e métodos de gestão de projetos . . . . .            | 66        |
| 6.4.1    | PMBOK Guide . . . . .  | 66        |
| 6.4.2    | PRINCE2 . . . . .  | 67        |
| <b>7</b> | <b>Processos de desenvolvimento de <i>software</i> . . . . .</b>       | <b>69</b> |
| 7.1      | Desenvolvimento de <i>software</i> . . . . .                           | 69        |
| 7.2      | Processos de desenvolvimento de <i>software</i> . . . . .              | 70        |
| 7.3      | Processos distribuídos de desenvolvimento de <i>software</i> . . . . . | 71        |
| 7.3.1    | Equipe virtual . . . . .   | 71        |

|                                       |   |            |
|---------------------------------------|---|------------|
| 7.3.2                                 | Comunidade virtual . . . . .  | 72         |
| 7.3.2.1                               | Comunidade virtual de profissionais . . . . .                           | 73         |
| 7.4                                   | Processo de desenvolvimento de <i>software</i> aberto . . . . .         | 74         |
| 7.4.1                                 | Participantes . . . . .   | 75         |
| 7.4.2                                 | Repositórios . . . . .  | 77         |
| <b>III Resultados</b>                 |   | <b>79</b>  |
| <b>8</b>                              | <b>Proposta de modelo de <i>framework</i></b> . . . . .                 | <b>80</b>  |
| 8.1                                   | Seleção de <i>frameworks</i> e métodos . . . . .                        | 80         |
| 8.2                                   | Definição de domínio e de pontos de vista de arquitetura . . . . .      | 81         |
| 8.3                                   | Projeto de descrição de arquitetura . . . . .                           | 82         |
| 8.4                                   | Modelo de ciclo de vida . . . . .                                       | 83         |
| 8.4.1                                 | Fases no ciclo de vida . . . . .  | 84         |
| 8.5                                   | Processos . . . . .   | 84         |
| 8.6                                   | Artefatos . . . . .   | 85         |
| 8.7                                   | Roteiros . . . . .  | 89         |
| 8.8                                   | Fontes de informação . . . . .  | 91         |
| <b>9</b>                              | <b>Exemplo de uso do modelo de <i>framework</i></b> . . . . .           | <b>94</b>  |
| 9.1                                   | Seleção do caso estudado . . . . .                                      | 94         |
| 9.2                                   | Fontes de informação . . . . .  | 96         |
| 9.3                                   | Visões de arquitetura adotadas . . . . .                                | 96         |
| 9.4                                   | Processos executados e artefatos construídos . . . . .                  | 97         |
| 9.5                                   | Descrições dos artefatos de gerenciamento construídos . . . . .         | 97         |
| 9.6                                   | Descrições dos artefatos de arquitetura construídos . . . . .           | 99         |
| 9.6.1                                 | Catálogo de princípios de arquitetura . . . . .                         | 101        |
| 9.6.2                                 | <i>Framework</i> de arquitetura configurado . . . . .                   | 101        |
| 9.6.3                                 | Catálogo de papéis . . . . .  | 101        |
| 9.6.4                                 | Catálogo de serviço/função de negócio . . . . .                         | 102        |
| 9.6.5                                 | Catálogo de localização . . . . .                                       | 103        |
| 9.6.6                                 | Modelo de casos de uso de negócio . . . . .                             | 103        |
| 9.6.7                                 | Taxonomia . . . . .   | 104        |
| 9.6.8                                 | Descrições de processos do negócio . . . . .                            | 107        |
| 9.7                                   | Considerações acerca dos artefatos de arquitetura construídos . . . . . | 108        |
| <b>Considerações finais</b> . . . . . |   | <b>109</b> |
| <b>Referências</b> . . . . .          |   | <b>112</b> |

|  |            |
|--|------------|
| <b>Apêndices</b>   | <b>124</b> |
| <b>APÊNDICE A Roteiros para processos de gerenciamento</b> . . . . . | <b>125</b> |
| <b>APÊNDICE B Roteiros para processos de arquitetura</b> . . . . .   | <b>132</b> |
| <b>APÊNDICE C Bibliometria</b> . . . . .                             | <b>142</b> |

# Introdução

A Ciência da Informação investiga propriedades e comportamentos das informações, forças que governam o fluxo e o uso das informações, e técnicas para o processamento das informações, visando aperfeiçoar o armazenamento, a recuperação e a disseminação das informações (BORKO, 1968). Interdisciplinaridade, conexão com a Tecnologia da Informação, participação na evolução da sociedade da informação, dimensão social e dimensão humana são características da Ciência da Informação (SARACEVIC, 1999). Anderson (2003) destaca que a Ciência da Informação lida com a descrição e a organização de artefatos pelos quais conhecimentos são representados e compartilhados, sendo a organização da informação uma contribuição desse campo para a sociedade. A Arquitetura da Informação é uma área da Ciência da Informação. Nesse contexto, *frameworks* para arquitetura da informação têm sido propostos. Alguns desses *frameworks* sugerem métodos para arquitetura da informação; mostram como componentes da arquitetura da informação se relacionam; recomendam produtos para documentação da arquitetura da informação; estabelecem vocabulários para descrição da arquitetura da informação (MALLOY et al., 2010).

Na Ciência da Informação, a informação é o fenômeno de interesse central (INGWERSEN, 1992). No desenvolvimento de *software*, a informação também assume papel importante. Produtos de software são desenvolvidos via processos intensos no uso de informações e colaborações entre os envolvidos (BANI-SALAMEH; JEFFERY; AL-GHARAIBEH, 2010; SARMA, 2005). Ao longo dos anos, a quantidade de produtos de *software* tem aumentado, assim como a complexidade dos mesmos. No passado recente, o desenvolvimento de *software* tem sido distribuído e muitos produtos de *software* passaram a ser desenvolvidos por comunidades virtuais. Nessas comunidades, na medida em que os seus membros interagem, informações se acumulam em repositórios, sendo importante a organização dessas informações.

Nesta dissertação é proposto um modelo de *framework* de arquitetura da informação baseado em roteiros, o qual é empregado na descrição parcial da arquitetura da informação em uma comunidade virtual voltada ao desenvolvimento de produto de *software* aberto. No contexto desta dissertação, um roteiro é uma sequência de passos para a execução de um processo. O modelo de *framework* proposto visa contribuir para a organização da informação, tendo sido construído a partir de informações em The Open Group Architecture Framework (TOGAF), Project Management Body of Knowledge (PMBOK) Guide, Método de Arquitetura da Informação Aplicada (MAIA) e outras fontes (referenciadas nesta dissertação).

---

Esta dissertação está estruturada em duas partes. Na primeira encontram-se capítulos onde são apresentados os resultados de pesquisa bibliográfica acerca dos seguintes temas: arquitetura da informação, *frameworks* e métodos para arquitetura da informação, modelagem da arquitetura da informação, linguagens para a construção de modelos de arquitetura, sistemas para a organização da informação, *frameworks* para a gestão de projetos, processos de desenvolvimento de *software*. A segunda parte é composta por dois capítulos e pelas considerações finais. Nesses capítulos, um modelo de *framework* de arquitetura da informação é proposto e são descritos resultados do uso desse modelo na descrição da arquitetura da informação em comunidade voltada ao desenvolvimento de *software* aberto.

# Parte I

## Preparação da pesquisa

# 1 Preparação da pesquisa

Neste capítulo, que tem o intuito de descrever atividades relacionadas à preparação da pesquisa, são descritos problema de pesquisa, objetivos de pesquisa, motivações para a realização da pesquisa, métodos de pesquisa empregados e percurso metodológico seguido.

## 1.1 Problema de pesquisa

No desenvolvimento de *software*, a informação apresenta papel importante, produtos de *software* são desenvolvidos via processos intensos no uso de informações e colaborações entre os envolvidos (BANI-SALAMEH; JEFFERY; AL-GHARAIBEH, 2010; SARMA, 2005). Diversos produtos de *software* são produtos de *software* aberto. Autores como Ankolekar, Herbsleb e Sycara (2003) destacam que um desafio no desenvolvimento de produtos de *software* aberto é a necessidade de se chegar com a informação certa à pessoa certa para a tarefa, e apresentá-la de forma compreensível e utilizável. Nesse contexto, esses autores destacam a importância da organização de informações originadas em diferentes fontes.

Apesar da importância da organização da informação no desenvolvimento de *software* e da importância da arquitetura da informação na organização da informação, consultas à fonte de informação Web of Science, realizadas em 20/11/2013, resultaram em poucas referências para documentos do tipo “*article*” ou “*proceedings paper*”, publicados entre 1994 e 2013, contendo, no tópico (título, *abstract*, palavras-chave), os termos “*information architecture*” e “*software engineering*”, ou “*information architecture*” e “*software development process*” (os resultados dessas consultas estão no apêndice C desta dissertação). Considerando o exposto, pesquisas que investiguem a arquitetura da informação em processos de desenvolvimento de *software* podem resultar em contribuições positivas para esses processos.

## 1.2 Objetivos

### 1.2.1 Objetivo geral

Propor um modelo de *framework* de arquitetura da informação baseado em roteiros.

### 1.2.2 Objetivos específicos

1. Identificar, na literatura, *frameworks* e métodos de arquitetura da informação.

2. Selecionar, e adaptar, *frameworks* e métodos de arquitetura da informação.
3. Para os processos integrantes do modelo de *framework* proposto, construir roteiros.
4. Exemplificar o uso do modelo de *framework* proposto na descrição parcial da arquitetura da informação em comunidade voltada ao desenvolvimento de *software* aberto.

### 1.3 Motivações e justificativas para a realização da pesquisa

1. Facilitar o acesso de desenvolvedores de *software* a informações necessárias à participação dos mesmos em comunidades voltadas ao desenvolvimento de *software* aberto, com o intuito de contribuir para a melhora da educação de desenvolvedores de *software*. Essa motivação se origina dos seguintes fatos: para serem produtivos, os desenvolvedores de *software* precisam aprender linguagens, ferramentas, jargões, processos e padrões, além de acessar múltiplas fontes de informação via variados canais de comunicação; na educação de desenvolvedores de *software*, é importante o estudo de produtos de *software* de elevada qualidade, desenvolvidos por outros, e ainda de seu reuso; é importante que desenvolvedores de *software* aprendam a aderir a padrões, valorizar a documentação, testar, considerar requisitos não funcionais e aplicar teorias à prática (FRASER et al., 2003; SHAW, 2000; RAGAN; FREZZA; CANNELL, 2009).
2. Considerando a importância da comunicação da informação tecnológica na indústria, no comércio, na inovação e no aperfeiçoamento de produtos e serviços, e a existência de barreiras a tal comunicação (FREIRE, 1991), a redução dessas barreiras é uma motivação para a realização desta pesquisa.
3. Considerando que processos de desenvolvimento de *software* são intensos no uso de informações e na colaboração entre os envolvidos (PRESSMAN, 2011; PERRY; STAUDENMEYER; VOTTA, 1994; SCHACH, 2008; SOMMERVILLE, 2010; BANISALAMEH; JEFFERY; AL-GHARAIBEH, 2010; SARMA, 2005), que a arquitetura da informação tem o potencial de facilitar o acesso às informações e o compartilhamento das mesmas, uma motivação para a realização desta pesquisa consiste em investigar a arquitetura da informação no contexto dos referidos processos de desenvolvimento.

## 1.4 Metodologia

### 1.4.1 Classificação da pesquisa

Esta é uma pesquisa em Ciência da Informação na área de conhecimento denominada Ciências Sociais Aplicadas pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Quanto à finalidade, esta pesquisa é aplicada pois gera conhecimentos para aplicação prática dirigidos à solução de problemas específicos, como descrito em Melo (2010) e Gil (2010).

Quanto à abordagem, esta pesquisa é exploratória pois visa possibilitar maior familiaridade com o problema com vistas a torná-lo mais explícito ou a construir hipóteses, como descrito em Melo (2010) e Gil (2010).

Quanto aos procedimentos técnicos, foram empregados pesquisa bibliográfica e estudo de caso, procedimentos descritos em Gil (2010) e Yin (1994).

### 1.4.2 Percurso metodológico

Para atingir os objetivos desta pesquisa, foram percorridas as seguintes fases: definição do tema de pesquisa; estudo preliminar; definição do problema de pesquisa; definição dos objetivos de pesquisa; definição dos métodos de pesquisa; realização de pesquisa bibliográfica; elaboração de proposta de modelo de *framework* de arquitetura da informação; emprego do modelo de *framework* proposto; elaboração e apresentação de dissertação.

A definição do tema de pesquisa decorreu da experiência profissional e de interesses do autor. O estudo preliminar envolveu o acesso a variadas fontes de informação com o intuito de estudar conceitos e teorias acerca do tema de pesquisa. A partir dos resultados desse estudo, foi identificado o problema de pesquisa e definidos os objetivos e os métodos de pesquisa.

A pesquisa bibliográfica envolveu o acesso a várias fontes de informação. Por exemplo: ABNT Coleção; ACM *Digital Library*; Base de dados ProQuest; Base de dados *Web of Science*; Biblioteca Central da Universidade de Brasília (BCE); Biblioteca do Instituto Brasileiro de Informação em Ciência e Tecnologia (IBICT); Biblioteca Digital Brasileira de Computação (BDBComp); Biblioteca digital EBRARY; IEEE Xplore *Digital Library*; Portal de periódicos da CAPES; Repositório Institucional da Universidade de Brasília; *Scientific Electronic Library Online* (SciELO); sítios das editoras Prentice Hall, Addison-Wesley e John Wiley and Sons; sítios das livrarias Amazon.com, Barnes & Noble, Livraria Cultura e Siciliano; sítios dos sistemas de busca Yahoo! e Google; *The Information Architecture Institute*.

A elaboração de proposta de modelo de *framework* de arquitetura da informação baseado em roteiros envolveu a execução das seguintes atividades: selecionar *frameworks* e métodos a serem configurados; definir domínio de arquitetura; definir pontos de vista de arquitetura; definir o modelo de ciclo de vida do projeto; definir as fases integrantes do modelo de ciclo de vida do projeto; definir processos de gerenciamento; definir processos de arquitetura; definir artefatos de gerenciamento; definir artefatos de arquitetura; descrever roteiros.

O emprego do modelo de *framework* proposto envolveu a execução das seguintes atividades: definir os critérios para a escolha do caso a ser estudado; selecionar o caso a ser estudado; selecionar as fontes de informação a serem acessadas; definir as visões de arquitetura adotadas; seguir roteiros do modelo de *framework* de arquitetura da informação proposto.

Com o intuito de selecionar a comunidade a ser estudada, foi consultado o sítio Ohloh (OHLOH, 2013). Esse sítio contém informações acerca de vários projetos de *software* aberto. Para selecionar a comunidade a ser estudada, foram definidos os seguintes critérios: projeto com mais de cem colaboradores no ano anterior às consultas; projeto identificado pelas etiquetas *cloudcomputing* ou *cloud\_computing*. A partir da análise de dados dos projetos que atenderam a esses requisitos, foi selecionada a comunidade do projeto com maior número de contribuições e colaboradores nos doze meses anteriores às datas das consultas.

A fase final da pesquisa foi dedicada à elaboração e à apresentação da dissertação. A dissertação foi estruturada em duas partes. Na primeira parte, foram apresentados os resultados da pesquisa bibliográfica realizada. Na segunda parte, foi proposto um modelo de *framework* de arquitetura da informação e foram apresentados resultados do uso desse modelo de *framework*.

## Parte II

### Referenciais teóricos

# Prólogo

Nesta etapa da dissertação são descritos resultados de pesquisa bibliográfica realizada com o intuito de possibilitar a identificação do estado teórico e prático acerca dos seguintes temas: arquitetura da informação, *frameworks* e métodos de arquitetura da informação, modelagem da arquitetura da informação, sistemas para organização da informação, *frameworks* para gestão de projetos e processos de desenvolvimento de *software*. Na figura 1 são estabelecidos relacionamentos entre alguns dos conceitos descritos nos próximos capítulos.

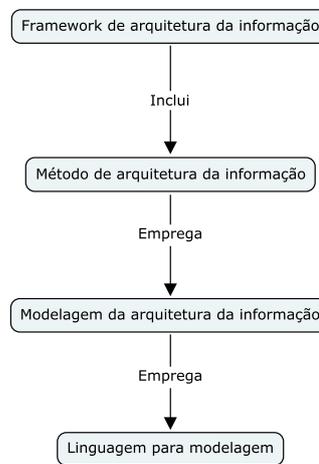


Figura 1 – Relacionamentos entre conceitos abordados na pesquisa bibliográfica.

No capítulo 2, dedicado à arquitetura da informação, são descritos conceitos acerca de arquitetura de sistemas e de arquitetura da informação. No capítulo 3, dedicado aos *frameworks* e métodos de arquitetura da informação, inicialmente são descritos conceitos acerca de tais *frameworks* e métodos, em seguida são descritos exemplos de tais *frameworks* e métodos. No capítulo 4, dedicado à modelagem da arquitetura da informação, inicialmente são apresentados conceitos acerca de descrição e modelagem de arquitetura, em seguida são descritos exemplos de linguagens de modelagem. No capítulo 5, dedicado a sistemas para organização da informação, inicialmente são apresentados conceitos acerca de organização da informação, em seguida são descritos exemplos de sistemas para organização da informação. No capítulo 6, dedicado a *frameworks* de gestão de projetos, inicialmente são descritos conceitos acerca de projeto, em seguida são descritos exemplos de *frameworks* de gestão de projetos. No capítulo 7, dedicado a processos de desenvolvimento de *software*, inicialmente são apresentados conceitos acerca de processo de desenvolvimento de *software*, em seguida são descritos exemplos de tais processos, com ênfase dada ao processo de desenvolvimento de *software* aberto.

## 2 Arquitetura da informação

Este capítulo tem o intuito de apresentar conceitos acerca de arquitetura de sistema, arquitetura da informação e arquitetura da informação organizacional.

### 2.1 Conceitos acerca de arquitetura de sistema

Em [The Open Group \(2011\)](#) e [The Open Group \(2013\)](#), o termo arquitetura tem as seguintes definições: (1) descrição formal de um sistema, ou plano detalhado de um sistema no nível de componentes, para guiar a sua implementação; (2) estrutura de componentes, inter-relações, princípios e diretrizes que governam o desenho (*design*) e a evolução ao longo do tempo. [Lankhorst \(2012\)](#) destaca que a arquitetura de um sistema provê uma visão integrada do sistema projetado ou estudado, e que uma arquitetura é uma estrutura associada a uma visão. Também destaca que, em vários domínios, a arquitetura de um sistema visa o entendimento e a definição de relacionamentos entre os usuários e o sistema desenhado.

Segundo a norma ISO/IEC/IEEE 42010 ([ISO, 2011](#)), a arquitetura de um sistema engloba conceitos fundamentais ou propriedades de um sistema no seu ambiente, incorporados nos seus elementos, relacionamentos, e nos princípios do seu desenho (*design*) e evolução, e expressa a essência, ou o fundamental, acerca de um sistema. A arquitetura de um sistema pode expressar: elementos do sistema; organizações e inter-relacionamentos dos elementos do sistema; princípios e padrões de organização ou desenho; princípios que governam a evolução do sistema.

Para [Greefhorst e Proper \(2011\)](#), existem distintas formas de arquitetura, que diferem em generalidade e escopo. Por exemplo: arquitetura de *software*, que define componentes, interações entre componentes, propriedades de componentes, relacionamentos entre componentes de *software* ([BACON et al., 2000](#); [BASS](#); [CLEMENTS](#); [KAZMAN, 2012](#); [ROZANSKI; WOODS, 2011](#)); arquitetura de referência, que é uma arquitetura genérica para sistemas com características similares; arquitetura de solução, que consiste na arquitetura de um sistema que oferece um conjunto coerente de funcionalidades ao seu ambiente ([GREEFHORST; PROPER, 2011](#)).

### 2.2 Conceitos acerca de arquitetura da informação

Alguns autores consideram que o termo arquitetura da informação foi originalmente proposto por Richard Saul Wurman, em 1976, para o título de uma conferência ([CAWKELL,](#)

2003). A definição de Wurman enfatiza a organização e apresentação das informações. Para Wurman, é possível estabelecer analogias entre problemas encontrados na coleta, organização e apresentação de informações, e problemas encontrados por arquitetos nos projetos de prédios para seus ocupantes. Para Jacob e Loehrlein (2009), a analogia entre arquitetura de prédios e da informação é uma metáfora, e permite conceituar o não físico em termos do físico.

The Information Architecture Institute (2013) e Morville e Rosenfeld (2006) apresentam as seguintes definições para o termo arquitetura da informação: desenho (*design*) estrutural de ambientes compartilhados de informação; combinação de organização, rotulação (*labeling*), sistemas de busca e navegação, em sítios; arte e ciência de dar forma a produtos de informação e a experiências, visando usabilidade e *findability*; disciplina emergente e comunidade de prática focada em trazer princípios de desenho (*design*) e arquitetura para o cenário digital. Dillon (2002) sugere que “arquitetura da informação é um termo usado para descrever o processo de desenhar (*design*), implementar e avaliar espaços de informação humanamente e socialmente aceitáveis a *stakeholders* visados”. Hagedorn (2000) define arquitetura da informação como arte e ciência de organizar informações, que visa auxiliar as pessoas a efetivamente preencherem as suas necessidades de informação. Ainda segundo Hagedorn (2000), a arquitetura da informação envolve investigação, análise, desenho (*design*) e implementação.

Ding e Lin (2009) sugerem que a arquitetura da informação visa organizar e simplificar as informações, e que, para atingir esses objetivos, projeta, integra, e agrega espaços de informação, criando modos das pessoas encontrarem, entenderem, trocarem e gerirem informações. Halvorson e Rach (2012) destacam que a arquitetura da informação envolve projetar sistemas de organização e navegação para auxiliar na localização e no gerenciamento de informações. Para Toms (2002), a arquitetura da informação inclui sistemas de classificação, identificação de conceitos dentro da estrutura de classificação, navegação, busca e acesso para determinado corpo de informação, e tem as seguintes origens: teorias da classificação e organização do conhecimento; categorização; pesquisas em desenho de menus; navegação hipertextual. Esse autor destaca que a interação das pessoas com ambientes digitais ricos em informação é influenciada pela arquitetura da informação nesses ambientes. Spencer e Featherstone (2010), por sua vez, sugerem que a arquitetura da informação envolve pessoas, conteúdos e contextos, e que visa organizar conteúdos, descrever conteúdos, prover meios para que as pessoas cheguem aos conteúdos.

Segundo Taylor e Joundrey (2009), embora existam várias definições para o termo arquitetura da informação, parece haver acordo quanto ao desejo de, a partir do entendimento das necessidades e comportamentos dos usuários, gerenciar documentos e facilitar o acesso às informações. Destacam que arquitetos da informação identificam usos que serão dados às informações, criam padrões para facilitar a localização das informações, e criam

interfaces atrativas para acesso às informações. Também ressaltam similaridades entre a arquitetura da informação e a organização da informação, tal como essa última é praticada em bibliotecas, arquivos e museus.

Siqueira (2012) destaca a necessidade de distinguir o termo arquitetura da informação com letras minúsculas, e o termo Arquitetura da Informação com letras maiúsculas. O primeiro designa a arquitetura da informação como produto, enquanto o segundo designa a disciplina. Observa que, tradicionalmente, a disciplina Arquitetura da Informação tem recebido uma abordagem tecnológica, voltada à organização de sítios na Internet ou à organização das informações para o seu consumo pelos usuários. A partir de revisão bibliográfica, esse autor apresenta as seguintes indicações: existem várias definições para o termo arquitetura da informação, que expressam diferentes conceitos; não há consenso acerca do corpo de conhecimento em Arquitetura da Informação; começam a surgir tentativas de formalização da Arquitetura da Informação e dos fundamentos teóricos que devem subsidiar a abordagem.

Segundo Albuquerque e Lima-Marques (2011), espaços de informação são compostos por conjuntos de informações, e o estado de um espaço de informação é uma configuração de informações em um intervalo de tempo. A arquitetura da informação pode então ser definida como a configuração dos estados dos elementos constituintes e suas propriedades, conceito esse aplicável a qualquer espaço de informação (LIMA-MARQUES, 2011). Para Lima-Marques (2011), considerando-se um espaço de informação em um domínio, obtida a configuração de informação desse espaço em um dado instante, é possível analisar a situação considerando-se paradigmas para a organização da informação no domínio e atingir um estado futuro desejado através de atos de transformação, que são eventos aplicados por um sujeito a um estado particular, visando mudanças em estados futuros.

Segundo Albuquerque e Lima-Marques (2011), o termo arquitetura da informação pode designar disciplina, produto da disciplina ou objeto de estudo da disciplina. Como disciplina, refere-se ao esforço sistemático de identificação de padrões e criação de metodologias para definir espaços de informação, assim como a criação de relacionamentos entre entidades linguísticas para definir esses espaços. Como produto, refere-se ao resultado do mencionado esforço. Como objeto de estudo, refere-se a um objeto, caracterizado como um espaço de conceitos inter-relacionados, que visa oferecer instrumentos para representação e manipulação da informação em determinados domínios. Albuquerque e Lima-Marques (2011) relacionam as seguintes categorias de definições para o termo arquitetura da informação: arquitetura da informação como desenho para ambientes específicos; arquitetura da informação como organização de espaços de informação de quaisquer tipos; arquitetura da informação como percepção da realidade. Propõem que a disciplina Arquitetura da Informação seja um programa de investigação epistemológica da realidade, e destacam que a Arquitetura da Informação pode adotar uma abordagem orientada a objeto, quando

visa analisar o fenômeno, ou orientada ao produto, quando visa criar um produto para atuar sobre o fenômeno.

No estudo da informação sobre as organizações, [Duarte \(2011\)](#) sugere o termo Arquitetura da Informação Organizacional (AIO), por considerar que o mesmo abrange qualquer tipo de organização e deixa clara a natureza da disciplina, além de um referencial teórico em que a Arquitetura da Informação Organizacional é composta por princípios, estruturas, métodos, padrões e instrumentos de tecnologia. Segundo [Duarte \(2011\)](#), a Arquitetura da Informação Organizacional dedica-se à análise, organização, desenho e comunicação de espaços de informação nas organizações. Para [Duarte \(2011\)](#), a Arquitetura da Informação Organizacional adota modelos, pode usar ontologias organizacionais, estuda a informação nos seus aspectos físicos, cognitivos e sociais, e abrange subespaços de informação conectados. A pesquisa descrita nesta dissertação se insere no contexto da AIO.

## 3 *Frameworks* e métodos de arquitetura da informação

Neste capítulo são apresentados conceitos sobre *frameworks* de arquitetura e *frameworks* de arquitetura da informação, são descritos exemplos de *frameworks*, são apresentados conceitos acerca de métodos de arquitetura e são descritos métodos de arquitetura da informação.

### 3.1 Conceitos acerca de *framework* de arquitetura

Um *framework* de arquitetura (*architectural framework*) é uma estrutura conceitual ou um conjunto de estruturas, para desenvolvimento, implementação e manutenção de arquiteturas (The Open Group, 2011). Esses *frameworks* facilitam o desenvolvimento incremental de arquiteturas e descrevem convenções e práticas de arquitetura em comunidades e domínios de aplicação (ISO, 2011; MALLOY et al., 2010). Esses *frameworks* podem incluir descrições de aspectos de interesse e escopo da arquitetura a ser desenvolvida; relacionamentos entre áreas de arquitetura; regras que expressam restrições a relacionamentos entre elementos na descrição da arquitetura; descrições de pontos de vista (*viewpoints*), convenções, princípios, práticas, técnicas, modelos e métodos; relação de resultados e descrições de como produzi-los; descrições de práticas para interpretação, análise e uso de descrições de arquiteturas; descrevem e relacionam vocabulários, modelos, processos, princípios, estratégias, ferramentas, arquiteturas de referência, catálogos de entregas e artefatos, roteiros de implementação, recomendações para governança, produtos, escopos de arquiteturas (GREEFHORST; KONING; VLIET, 2006; ISO, 2011; LANKHORST, 2012; The Open Group, 2011; BERNARD, 2012). Os *frameworks* de arquitetura podem ser usados em vários contextos, como em processos que facilitam a comunicação e interoperação entre projetos e/ou organizações; nas descrições de arquiteturas; no desenvolvimento de ferramentas e métodos para modelagem de arquiteturas ISO (2011). Existem vários *frameworks* de arquitetura, como os que constam da lista elaborada pelo grupo de trabalho ISO/IEC JTC1 SC7 WG42 (WG42, 2013).

### 3.2 *Frameworks* de arquitetura da informação

Por sua vez, os *frameworks* de arquitetura da informação (*information architecture frameworks*) proveem metodologias para a descrição de ativos de informação; mostram como os componentes da arquitetura se relacionam; recomendam produtos para documen-

tação de arquiteturas da informação; estabelecem vocabulários para descrição de produtos da arquitetura da informação (MALLOY et al., 2010). Entre as motivações para a existência de *frameworks* voltados à modelagem e à documentação de informações, tem-se a complexidade dos requisitos de dados, dos produtos de *software* e *hardware*; e dificuldades no compartilhamento de informações, que podem ilhar áreas de negócio e duplicar recursos (BERNARD, 2012). A seguir, serão descritos três *frameworks* de arquitetura da informação no contexto organizacional.

### 3.2.1 *Framework for Enterprise Architecture*

O *Framework for Enterprise Architecture* (ZACHMAN, 1987; SOWA; ZACHMAN, 1992) é uma classificação de representações para descrições de organizações (WG42, 2013). Moshiri e Hill (2011) e Sessions (2007) destacam que esse *framework* é um sistema para organizar artefatos de arquitetura e que o termo taxonomia melhor o define. Segundo Zachman (2008), esse *framework* engloba representações importantes à descrição de organizações, esse *framework* pode ser considerado uma ontologia, “uma teoria da existência de um conjunto estruturado de componentes essenciais de um objeto, para o qual expressões explícitas são necessárias, talvez mesmo mandatórias, para a criação, operação e modificação do objeto”, em que o termo objeto pode designar organização, departamento ou projeto. Para Zachman (2008), esse *framework* estabelece definições e não transformações. Não é metodologia e nem processo.

O *Framework for Enterprise Architecture* foi proposto após John Zachman analisar representações de arquitetura de produtos de engenharia e identificar representações que expressavam diferentes perspectivas. Ele constatou que as diferenças entre tipos de representação não eram apenas no detalhamento, os tipos de representação variavam nas suas essências, e representavam diferentes perspectivas. Cada tipo de representação de arquitetura apresentava características particulares.

O *Framework for Enterprise Architecture* é representado por uma matriz com seis colunas e seis linhas. Cada coluna representa uma abstração que corresponde a um esquema de classificação sugerido pelas seguintes questões: O QUE, COMO, ONDE, QUEM, QUANDO e POR QUE. Cada linha corresponde a uma perspectiva distinta e única de uma audiência. As perspectivas são as seguintes: Executiva (*Executive Perspective*), Gestão de Negócio (*Business Management Perspective*), Perspectiva de Arquiteto (*Architect Perspective*), Engenharia (*Engineer Perspective*), Técnica (*Technician Perspective*), Empresarial (*Enterprise Perspective*). Cada interseção entre coluna e linha contém uma célula com um modelo.

### 3.2.2 The Open Group Architecture Framework

O *The Open Group Architecture Framework* (TOGAF), *framework* mantido pelo *The Open Group*, engloba métodos e ferramentas para produção, aceitação, uso e manutenção de arquiteturas nos domínios de negócios, dados, aplicações e tecnologia. O TOGAF é um *framework* de arquitetura configurável que sugere a revisão dos componentes do *framework* para identificar sua aplicabilidade e necessidade ([The Open Group, 2011](#)). A configuração pode ocorrer, por exemplo, via integração do TOGAF a métodos de outros *frameworks*.

O TOGAF é composto de *TOGAF Architecture Development Method* (ADM), método para desenvolvimento de arquiteturas; *TOGAF Architecture Content Framework* (ACF), *framework* para definição, estruturação e apresentação de produtos resultantes das atividades no TOGAF ADM; e *Enterprise Continuum & Tools*, conjunto de métodos para classificação de arquiteturas, descrições de características para classificação e divisão de arquiteturas, recomendações de como empregar classificações de arquiteturas na estruturação de repositórios, recomendações para seleção de ferramentas para criação e gerenciamento de artefatos.

### 3.2.3 EA3 Cube Framework

No *EA3 Cube Framework*, *framework* descrito em [Bernard \(2012\)](#), as atividades são organizadas em visões com as perspectivas de estratégia, negócio e tecnologia. No sítio *The EA Pad*, é destacado que esse *framework* originalmente tinha intuito educacional, mas passou a ser empregado na academia, em programas de treinamento profissional e na prática. Em *The EA Pad* é destacado que esse *framework* influenciou a iniciativa *Common Approach to Federal EA*, descrita em [EOP \(2012\)](#).

O *EA3 Cube Framework* é organizado nos níveis hierárquicos “Objetivos e Iniciativas”, “Produtos e Serviços”, “Dados e Informações”, “Sistemas e Aplicações” e “Redes e Infraestrutura” ([BERNARD, 2012](#)). As atividades em “Objetivos e Iniciativas” identificam a direção estratégica, objetivos e iniciativas da empresa, a contribuição da TI, enquanto as atividades no nível “Produtos e Serviços” identificam produtos e serviços, a contribuição da tecnologia no suporte a esses produtos e serviços. As atividades em “Dados e Informações”, por sua vez, documentam como as informações são usadas, fluxos futuros de informação, desenhos de bases de dados, padrões, formatos de dados, dicionários de dados e repositórios, enquanto as atividades no nível “Sistemas e Aplicações” documentam sistemas de informação e aplicações na empresa. As atividades em “Redes e Infraestrutura” documentam visões atual e futura das redes de voz, dados e vídeo. No sentido vertical, cruzando os níveis hierárquicos descritos, encontram-se as linhas de negócio.

O *EA3 Cube Framework* relaciona artefatos que resultam da documentação de

componentes da arquitetura. São exemplos desses artefatos: plano estratégico, plano de negócios, plano de gestão do conhecimento, narrativas e diagramas de casos de uso, modelo lógico de dados, modelo físico de dados, dicionário de dados, modelo de processos de negócio, diagrama de fluxo de dados (BERNARD, 2012).

### 3.3 Métodos de arquitetura da informação

Um método pode ser definido como uma abordagem definida e reproduzível para tratar um tipo de problema (The Open Group, 2012; The Open Group, 2013). Segundo Lankhorst (2012), um método de arquitetura, por sua vez, é uma coleção estruturada de técnicas e processos para criação e manutenção de arquiteturas. Para Lankhorst (2012), os métodos de arquitetura especificam fases no ciclo de vida da arquitetura, quais resultados produzir nessas fases, como verificar e testar esses resultados. A seguir, são descritos exemplos de métodos de arquitetura da informação.

#### 3.3.1 Método descrito em Morville e Rosenfeld (2006)

Em Morville e Rosenfeld (2006), é descrito um método de arquitetura da informação de sítios composto pelas fases Pesquisa, Estratégia, Desenho, Implementação e Administração. A seguir são descritas as fases Pesquisa, Estratégia e Desenho, principais fases no desenho (*design*) de arquiteturas da informação.

Na fase Pesquisa, os autores sugerem as seguintes atividades: definir objetivos, arquitetura da informação existente, contexto, audiência, cronograma, orçamento, infraestrutura tecnológica, estrutura e cultura organizacional; pesquisar documentos; consultar técnicos, gestores de conteúdos, administradores de sistemas, desenvolvedores de *software*, gerentes e executivos para definir como os conteúdos serão criados e gerenciados; analisar conteúdos para identificar padrões, relacionamentos e metadados; estudar necessidades dos usuários; testar a usabilidade.

A fase Estratégia visa definir a estratégia de desenvolvimento e manutenção da arquitetura da informação. Nessa fase são identificadas tecnologias para desenvolvimento e gerência da arquitetura da informação e é escolhida uma estratégia para desenvolvimento. Por exemplo, ascendente ou descendente. De acordo com Brink, Gergle e Wood (2002), na estratégia ascendente, o desenvolvimento parte do entendimento do contexto, do conteúdo, das ferramentas e necessidades dos usuários. Os conteúdos são inicialmente coletados e organizados em categorias, e posteriormente agrupados em categorias de maior nível hierárquico. Na estratégia descendente, as categorias de maior nível hierárquico são inicialmente especificadas, e depois decompostas em categorias de menor nível hierárquico.

Na fase Desenho, a arquitetura da informação deve ser comunicada. Para isso, Morville e Rosenfeld (2006) recomendam a construção de *blueprints* em que grupos de

páginas, componentes de páginas, relacionamentos entre componentes e entre páginas sejam representados. Também sugerem a construção de *wireframes* que descrevam as páginas segundo a arquitetura da informação. Na fase Desenho, conteúdos em vários formatos devem ser organizados e mapeados para a arquitetura da informação. Para isso, [Morville e Rosenfeld \(2006\)](#) sugerem que os conteúdos sejam inventariados e que sejam construídos modelos em que conteúdos relacionados sejam agrupados e ligações entre grupos estabelecidas. Nessa fase, também sugerem o desenvolvimento de vocabulários controlados e a rotulação dos conteúdos. Uma fonte de rótulos podem ser taxonomias. Se não existirem taxonomias para o domínio, esses autores sugerem a criação de sistema de identificação via análise dos conteúdos, consultas a autores, especialistas no domínio e usuários.

### 3.3.2 Método descrito em [Spencer e Featherstone \(2010\)](#)

Em [Spencer e Featherstone \(2010\)](#) é descrito um método de arquitetura da informação de sítios composto pelas fases Abertura, Pesquisa, “Desenho e Teste” e Manutenção. A seguir são descritas as fases Abertura, Pesquisa, “Desenho e Teste”, principais fases no desenho (*design*) da arquitetura da informação. A fase Abertura visa identificar objetivos, requisitos, cultura organizacional, restrições ao projeto, perfis dos interessados, oportunidades e restrições tecnológicas. A fase Pesquisa engloba as seguintes atividades: estudar usuários para identificar necessidades de informação, o que é feito com as informações, onde as informações são usadas, quais informações são conhecidas, níveis de conhecimento das pessoas quanto a tecnologias, terminologias e conceitos importantes; compreender, detalhar e priorizar conteúdos; definir formatos e audiências dos conteúdos; planejar a disponibilização dos conteúdos. Nessa fase, sugerem o uso das seguintes técnicas: entrevista, grupo focal, observação, estudo de diários, *card sorting* e questionário. Também recomendam pesquisas às estatísticas de uso de sítios; aos registros de termos empregados; aos registros de mensagens em listas de correio eletrônico, fóruns, *call centers* e *help desks*. Para a identificação dos conteúdos existentes, sugerem inventários e auditorias, precedidos pela identificação das fontes de informação.

Na fase “Desenho e Teste”, [Spencer e Featherstone \(2010\)](#) sugerem o uso de soluções padronizadas para problemas recorrentes de desenho, a definição correta, consistente e clara dos rótulos que identificam conteúdos. Também sugerem um processo iterativo com as seguintes atividades: elaborar e criticar esboço de arquitetura considerando o público alvo e a distribuição dos conteúdos; revisar e detalhar esse esboço a cada iteração; após iterações suficientes, discutir o esboço com os interessados e realizar as modificações necessárias; realizar testes de usabilidade; documentar arquitetura, decisões de desenho e lições aprendidas.

### 3.3.2.1 Inventário e auditoria de conteúdos

Em diversos métodos de arquitetura da informação, como o descrito em [Spencer e Featherstone \(2010\)](#), são recomendados inventários e auditorias dos conteúdos existentes. Essas auditorias visam facilitar a definição do escopo de um projeto de conteúdos, a definição dos recursos necessários a um projeto de conteúdos, o entendimento e a localização dos conteúdos, o estabelecimento de referências aos conteúdos existentes quando do desenvolvimento de novos conteúdos ([HALVORSON; RACH, 2012](#)). Uma auditoria de conteúdos pode ser um inventário quantitativo ou uma auditoria qualitativa. Os inventários quantitativos identificam e localizam conteúdos, enquanto as auditorias qualitativas avaliam a qualidade e a eficácia dos conteúdos ([HALVORSON; RACH, 2012](#)). No inventário quantitativo de um sítio, para cada página, as seguintes informações podem ser coletadas: título, tópico ou área de conhecimento; descrição resumida; datas na qual a página foi criada e revisada; identificador de quem criou, aprovou e publicou os conteúdos; ligações para outras páginas; metadados, palavras-chave e etiquetas; estatísticas de uso; linguagens usadas ([HALVORSON; RACH, 2012](#)).

Em auditorias qualitativas de sítios, são realizadas atividades como comparar os conteúdos do sítio às melhores práticas no mercado, identificar conteúdos de baixa qualidade ou faltantes, avaliar se os conteúdos estão alinhados aos objetivos do negócio e dos usuários ([HALVORSON; RACH, 2012](#)). [Rockley e Cooper \(2012\)](#) destacam a necessidade de definição do escopo da auditoria e seleção de materiais representativos. Nas auditorias qualitativas, [Halvorson e Rach \(2012\)](#) sugerem que seja considerado o nível de conhecimento para entendimento do conteúdo, facilidade com a qual o conteúdo pode ser encontrado, audiências às quais se destina o conteúdo, precisão e atualidade do conteúdo. [Rockley e Cooper \(2012\)](#) destacam a importância dos conteúdos auxiliarem os seus consumidores a completarem tarefas, tomarem decisões e satisfazerem suas necessidades. Também sugerem que, na avaliação dos conteúdos, se atente para aspectos como propriedade dos conteúdos, terminologia, tom, nível de detalhamento e presença de lacunas.

### 3.3.3 Método descrito em [DTF \(2010\)](#)

Em [DTF \(2010\)](#) é sugerido um método de arquitetura da informação com as fases “Análise das necessidades do negócio”, “Pesquisa e análise da audiência”, “Desenho e documentação da arquitetura da informação”, “Desenho do sítio” e “Gestão operacional”. A fase “Análise das necessidades do negócio” visa identificar objetivos e escopo do sítio, identificar o contexto do negócio, minimizar a duplicação de informações e esforços. A fase “Pesquisa e análise da audiência” visa identificar usuários e suas características, descrever os usuários considerando seus requisitos de interação, definir modelos de interação, identificar necessidades de informação dos usuários, definir conteúdos a disponibilizar. Nessa fase são realizadas as seguintes atividades: analisar chamadas a *call centers*; entrevistar usuários;

analisar estatísticas de uso e de pesquisa ao sítio; analisar sítios similares; realizar grupos focais. Os resultados dessa fase podem incluir *storyboards*, *wireframes*, protótipos, diagramas da estrutura do sítio, de casos de uso, de interação, de sequência e de fluxo de dados.

Na fase “Desenho e documentação da arquitetura da informação”, são sugeridas as seguintes atividades: inventariar conteúdos; definir vocabulário controlado e metadados; definir estruturas de classificação e navegação; definir *layout* do sítio; documentar modelo de busca; avaliar se a arquitetura atende aos usuários. No desenho do sistema de navegação, podem ser realizadas as seguintes atividades: identificar as necessidades dos grupos de usuários; estabelecer balanceamento entre necessidades dos usuários e objetivos do negócio; prover múltiplos trajetos para os conteúdos visando atender diferentes grupos de usuários; analisar estatísticas de uso de sítios para identificar termos, páginas e caminhos de navegação populares; analisar sítios similares visando identificar os seus sistemas de classificação e navegação; identificar estilos de linguagem a empregar; definir rótulos de categorias e páginas; definir taxonomias para o sistema de navegação; definir categorias intuitivas aos usuários. Para documentar a arquitetura da informação, são sugeridas planilhas e mapas.

Na fase “Desenho e documentação da arquitetura da informação”, é recomendado definir os *lay-outs* das principais páginas via *wireframes*, que são diagramas com aspectos estruturais das páginas, mas sem detalhes visuais. Em relação ao modelo de busca, é sugerido que o mesmo inclua informações acerca de conteúdos frequentemente procurados; estilos de linguagem e termos usados nas buscas; abrangência da indexação e das buscas; modos como grupos de usuários interagirão com o mecanismo de busca; prioridades dos conteúdos nas pesquisas; como aplicar operadores nas buscas; quais recursos de busca devem ser providos; quantos e quais elementos devem ser apresentados nos resultados das buscas. Nessa fase também são sugeridos testes de usabilidade para avaliar se os usuários são capazes de localizar as informações e serviços necessários; reconhecer os elementos nas páginas; avaliar se a linguagem e os agrupamentos refletem o modo como os usuários pensam; se os conteúdos são claros e importantes. São aspectos avaliáveis nesses testes: legibilidade, *layout*, navegação, nomes de categorias, qualidade de conteúdos, modos de apresentação dos conteúdos e qualidade dos resultados das pesquisas.

Na fase “Desenho do sítio”, a arquitetura da informação é traduzida no desenho da interface com o usuário. Na fase “Gestão operacional”, a arquitetura da informação é monitorada, e modificada quando necessário. Para se avaliar a necessidade de modificações, o método em DTF (2010) sugere a monitoração das estatísticas de uso, termos usados em pesquisas, trajetos percorridos e o uso de conteúdos.

### 3.3.4 Método descrito em *Princeton University (2008)*

Em *Princeton University (2008)*, é sugerido método de arquitetura da informação com as seguintes fases: “Identificação dos objetivos das principais partes interessadas”, “Identificação dos objetivos e expectativas dos usuários”, “Definição das áreas de conteúdos do sítio”, “Organização das áreas de conteúdos”, “Construção do mapa do sítio”, “Esboço da estrutura de navegação”, “Rotulação das áreas de conteúdos”, “Construção de *wireframes*”.

Na fase “Identificação dos objetivos das principais partes interessadas”, procura-se determinar as audiências do sítio, as expectativas das partes interessadas em relação ao sítio, as finalidades e os requisitos funcionais do sítio, enquanto na fase “Identificação dos objetivos e expectativas dos usuários” procura-se identificar os objetivos e as expectativas dos usuários para possibilitar que as informações sejam organizadas e rotuladas de acordo com suas necessidades.

Na fase “Definição das áreas de conteúdos do sítio”, considerando-se a importância dos conteúdos para os usuários, os conteúdos são analisados para definir quais conteúdos acrescentar, atualizar ou descartar do sítio. Na fase “Organização das áreas de conteúdo”, grupos de conteúdos relacionados são criados, rotulados e empregados na construção do sistema de navegação, enquanto na fase “Criação do mapa do sítio”, é construído mapa que representa as áreas de conteúdo de modo hierárquico. Uma vez criado, esse mapa é avaliado por usuários. Na fase “Esboço da estrutura de navegação”, um esboço da estrutura de navegação é construído a partir do mapa do sítio desenvolvido. Os *links* podem ser posicionados em barras de navegação e devem conduzir a páginas de conteúdo.

Na fase “Rotulação das áreas de conteúdo”, são definidos termos que serão empregados para identificar grupos de conteúdos. Os nomes das categorias devem ser refinados em rótulos e os usuários devem testar a nomenclatura definida. Por fim, na fase “Construção de *wireframes*”, esboços que representam as organizações das páginas são construídos. Esses esboços são então usados no desenho gráfico das páginas e em outras atividades no processo de desenvolvimento do sítio.

### 3.3.5 Método de Arquitetura da Informação Aplicada

*Costa (2009)* sugere o Método de Arquitetura da Informação Aplicada (MAIA) para arquiteturas da informação aplicadas. Nesse método, as atividades são distribuídas em quatro momentos: Escutar, Pensar, Construir e Habitar. Os momentos ocorrem de modo sequencial e cíclico. Em cada momento, o sujeito executa ações sobre o espaço de informações e, sobre os resultados dessas ações, são aplicadas novas ações até se chegar à conclusão do ciclo. Os ciclos se sucedem e a evolução da arquitetura da informação ocorre em espiral. Os momentos são compostos por atos, não havendo uma delimitação nítida entre eles.

O momento Escutar é composto pelos atos ouvir e interpretar. O ouvir consiste no ato de captura das manifestações de informações e representa os procedimentos de conexão do arquiteto da informação com o espaço de informação a ser analisado. Ao revelar a sua estrutura, o espaço de informação passa a ser reconhecido como uma arquitetura da informação. Após a captura dos registros, ocorre o seu armazenamento. No passo seguinte é definida uma taxonomia para os registros armazenados que são então catalogados. [Costa \(2009\)](#) sugere a construção de mapas conceituais para gerar a representação. O momento Escutar resulta em ontologias que representam o espaço de informação inicial. No momento Escutar, o ato ouvir captura e armazena registros e o ato interpretar codifica ontologias a partir desses registros. O momento Escutar se completa quando for gerada uma representação esquemática do espaço de informação inicial, a partir da codificação das relações entre as informações.

O momento Pensar contém os atos interpretar e modelar. Nesse momento, o arquiteto da informação interpreta o mapa de conceitos que representa o espaço de informação inicial e constrói modelos que representam a arquitetura da informação. Esses modelos agregam características de função, forma e estética. O resultado dos atos de interpretar e modelar é uma representação ontológica da arquitetura da informação. O critério para conclusão da fase dedicada ao momento Pensar ocorre quando gerado artefato que represente forma, função e estética, e gerada representação ontológica da arquitetura da informação.

O momento Construir é composto por ações que migram o espaço de informação inicial para um novo estado, via atos de modelar e transformar. O ato modelar produz a representação dos registros do espaço de informação enquanto o ato transformar reconfigura o espaço de informação. Inicialmente o arquiteto da informação define os registros que serão criados, descartados ou alterados no espaço de informação com base nos modelos anteriormente criados. Em seguida, é produzido um roteiro de alterações. No ato transformar, registros de informação são manipulados de acordo com as ações no roteiro elaborado. A execução dessas ações resulta em um novo estado da arquitetura da informação que representa o desenho proposto no momento modelar. A fase dedicada ao momento Construir é concluída quando elaborada estratégia de manipulação que represente ações ordenadas de transformação do espaço de informação, e gerado um novo estado da arquitetura da informação.

No momento Habitar, o espaço de informação é incorporado pelo sujeito via os atos transformar e estar. Nesse momento, o arquiteto da informação, e outros interessados, usam, homologam e validam a arquitetura da informação, avaliando sua conformidade com critérios de desenho. O espaço de informação é usado de modo controlado e são registrados os resultados da avaliação. No final dos testes, é emitido parecer que informa se a arquitetura foi aceita. A fase dedicada ao momento Habitar é concluída após o novo

estado da arquitetura da informação ser incorporado à realidade e validado.

### 3.3.6 Método de Arquitetura Ágil da Informação Organizacional

Na Arquitetura Ágil da Informação Organizacional (2AIO), Duarte (2011) define método com as seguintes fases: Organização, Implantação, Planejamento e Gestão. Segundo Duarte (2011), a proposta 2AIO se baseia nos seguintes princípios: início simples, abordagem pragmática, evolução incremental, valorização das pessoas e trabalho colaborativo. Para Duarte (2011), são elementos da proposta 2AIO: ontologia organizacional; repositório com instâncias de elementos da ontologia e conteúdos; ambiente colaborativo de modelagem e consulta; estrutura de navegação na informação; estrutura de modelagem conceitual abordando negócio, aplicações e tecnologia; abordagem colaborativa de documentação e desenho.

Na proposta 2AIO, é sugerida a construção de uma ontologia da informação organizacional para identificar e definir os elementos da organização e suas relações. Para Duarte (2011), essa ontologia possibilita o controle do vocabulário e o compartilhamento de conhecimentos, e integra os conteúdos no repositório. Os conteúdos são relacionados às classes da ontologia via metadados. Duarte (2011) destaca que a 2AIO abrange subespaços (negócio, aplicações e tecnologia) com termos distintos, cujas ontologias identificam a forma de operar da organização, além de documentar e direcionar a engenharia dos elementos organizacionais. Para o autor, é possível considerar a existência dos seguintes mundos organizacionais interligados: negócios, processos, controles, dados, aplicações e tecnologia.

### 3.3.7 TOGAF Architecture Definition Method

O *TOGAF Architecture Definition Method* (TOGAF ADM) integra o *framework* TOGAF (The Open Group, 2011). O TOGAF ADM é organizado em ciclos de desenvolvimento, sendo cada ciclo composto por fases. Os nomes dessas fases são os seguintes:

- Fase Preliminar.
- Fase A: Visão de Arquitetura.
- Fase B: Arquitetura de Negócio.
- Fase C: Arquitetura de Sistemas de Informação – Arquitetura de Dados.
- Fase C: Arquitetura de Sistemas de Informação – Arquitetura de Aplicações.
- Fase D: Arquitetura de Tecnologia.
- Fase E: Oportunidades e Soluções.

- Fase F: Planejamento de Migração.
- Fase G: Governança de Implementação.
- Fase H: Gerenciamento de Mudança de Arquitetura.
- Gerenciamento de Requisitos de Arquitetura.

A fase Preliminar visa identificar princípios da organização, requisitos de arquitetura, *frameworks* e relacionamentos entre eles, e avaliar a maturidade da arquitetura existente. A fase “Visão de Arquitetura” visa definir o esforço de arquitetura e restrições a observar, garantir que existam definições para princípios, objetivos e motivações estratégicas para o negócio, garantir que existam definições para os princípios de arquitetura, definir os objetivos do trabalho de arquitetura, e obter aprovação da visão de arquitetura. As fases “Arquitetura do Negócio”, “Arquitetura de Sistemas de informação” e “Arquitetura de Tecnologia” visam desenvolver as arquiteturas alvo e identificar componentes de arquitetura a partir das diferenças entre as arquiteturas atuais e as arquiteturas alvo. A fase “Oportunidades e Soluções” visa definir trabalho para realizar a arquitetura alvo, identificar as arquiteturas de transição, iniciar a elaboração de planos de implementação e de migração para as arquiteturas alvo. A fase “Planejamento de Migração”, por sua vez, visa finalizar os planos de implementação e migrar para as arquiteturas alvo, enquanto a fase “Governança da Implementação” visa garantir a conformidade entre projetos na organização e as arquiteturas alvo. A fase “Gestão de Modificações da Arquitetura” visa garantir que as arquiteturas alcancem os objetivos e estabelecer procedimentos para modificações, e, por fim, a fase “Gestão de Requisitos” visa gerenciar requisitos.

### 3.3.8 Método descrito em [Bernard \(2012\)](#)

Em [Bernard \(2012\)](#) é descrito um método com as fases “Estabelecimento de Programa”, “Seleção de *Framework* e Ferramentas”, “Documentação” e “Uso e Manutenção”. A fase “Estabelecimento do Programa” visa estabelecer programa e metodologia de arquitetura, identificar arquiteto chefe, estabelecer governança da arquitetura, estabelecer ligações entre a arquitetura e outros processos, e planejar comunicação para adesão dos *stakeholders*. A fase “Seleção de *Framework* e Ferramentas” visa selecionar *framework*, métodos e ferramentas, identificar componentes a documentar, selecionar e estabelecer repositório *on-line*. A fase “Documentação da Arquitetura” visa documentar a arquitetura, desenvolver cenários futuros e usá-los na documentação da arquitetura, armazenar artefatos em repositório, e desenvolver plano para gerenciar mudanças na arquitetura. Por fim, a fase “Uso e Manutenção” visa usar a arquitetura no planejamento e na tomada de decisões, atualizar a arquitetura, manter o repositório, e atualizar o plano de gestão da arquitetura.

## 4 Modelagem de arquitetura da informação

Neste capítulo, inicialmente são apresentados conceitos acerca de descrição e modelagem de arquitetura, em seguida são apresentados conceitos acerca de linguagens para modelagem de arquitetura, importantes a esta pesquisa.

### 4.1 Descrição de arquitetura de sistema

A descrição da arquitetura de um sistema resulta de trabalho realizado para expressar a arquitetura do sistema, de atividades realizadas ao longo do ciclo de vida do sistema (ISO, 2011). Segundo Rozanski e Woods (2011), a descrição de uma arquitetura é composta por produtos que a documentam de tal forma que seus *stakeholders* possam entendê-la. A descrição deve demonstrar que a arquitetura atende requisitos, objetivos, intenções e aspirações dos *stakeholders*. De acordo com Lankhorst (2012), a descrição de uma arquitetura pode conter modelos e descrições textuais da arquitetura. Lankhorst (2012) destaca que as descrições de arquitetura são atualmente heterogêneas, isto é, cada domínio tem sua própria técnica de descrição, textual ou gráfica, informal ou com significado preciso.

Segundo a norma ISO/IEC/IEEE 42010 (ISO, 2011), *stakeholder* é indivíduo, equipe, organização ou classe com interesse no sistema. Em um projeto, Heldman (2011) define *stakeholder* como pessoa, ou organização, com interesse (têm algo a ganhar ou a perder) no projeto. Para o *Project Management Institute* (PMI, 2013), *stakeholder* é um indivíduo, grupo ou organização que pode afetar ou ser afetado, ou perceber-se afetado, por uma decisão, atividade ou resultado de um projeto. Em PMBOK (PMI, 2013) é destacado que *stakeholders* podem estar ativamente envolvidos no projeto ou ter interesses que podem ser afetados pelo desempenho ou conclusão do projeto. Os *stakeholders* de um projeto incluem membros da equipe do projeto e todas as entidades interessadas, internas ou externas à organização (PMI, 2013). No contexto de arquiteturas, *stakeholder* é pessoa, grupo, ou entidade, com interesses (*concerns*) relativos às arquiteturas (ROZANSKI; WOODS, 2011). Em TOGAF (The Open Group, 2011), o termo *concern* é definido como um interesse crucial para os *stakeholders*, um interesse que determina a aceitação do sistema.

Em PMBOK (PMI, 2013), no contexto de projetos, a gestão de *stakeholders* engloba identificação de *stakeholders*; planejamento da gestão de *stakeholders*; gerência e controle do engajamento de *stakeholders*. Existem várias técnicas para a identificação e análise de *stakeholders*. No TOGAF (The Open Group, 2011) é sugerido, como parte do *Architecture Definition Method* (ADM), que *stakeholders* sejam identificados, analisados e classificados.

O TOGAF destaca a importância de se entender os interesses dos *stakeholders* para a definição do que é importante na descrição da arquitetura. Em PMBOK (PMI, 2013) é destacado que a identificação dos *stakeholders*, o entendimento dos níveis de influência dos *stakeholders* sobre o projeto, e o balanceamento entre demandas, necessidades e expectativas dos *stakeholders*, podem ser considerados aspectos críticos para o sucesso de um projeto.

Para Rozanski e Woods (2011), uma boa descrição de arquitetura comunica efetiva e consistentemente os aspectos da arquitetura aos *stakeholders*. Greefhorst, Koning e Vliet (2006) destacam que cada *stakeholder* tipicamente se interessa por apenas parte da descrição da arquitetura. Rozanski e Woods (2011) consideram que uma descrição de arquitetura deve representar necessidades e interesses dos *stakeholders*; definir a arquitetura que atenda às necessidades dos *stakeholders*; conter detalhes suficientes para ser possível responder a questões importantes acerca da arquitetura; focar nos elementos importantes da arquitetura; ser compreensível aos *stakeholders*; e estar atualizada em relação à arquitetura que descreve.

A norma ISO/IEC/IEEE 42010 (ISO, 2011) destaca que uma arquitetura pode caracterizar vários sistemas e ser expressa por distintas descrições, que descrições de arquiteturas podem ser usadas para melhorar a comunicação e cooperação entre quem cria, utiliza e gerencia sistemas. Segundo a norma ISO/IEC/IEEE 42010 (ISO, 2011), a arquitetura de um sistema é uma concepção holística das principais propriedades do sistema. Ainda segundo essa norma, a arquitetura de um sistema pode ser mais bem entendida via múltiplas visões da arquitetura, compostas por modelos que a expressam sob pontos de vista distintos. A figura 2 apresenta o contexto de uma descrição de arquitetura segundo a norma ISO/IEC/IEEE 42010 (ISO, 2011).

A norma ISO/IEC/IEEE 42010 (ISO, 2011) relaciona os seguintes usos de descrições de arquiteturas de sistemas: análise e avaliação de alternativas no desenvolvimento de sistemas; documentação para desenvolvimento e manutenção de sistemas; documentação de aspectos essenciais aos sistemas; documentação de características comuns entre diferentes sistemas; comunicação entre os interessados no desenvolvimento de sistemas; revisão, análise e avaliação de sistemas ao longo de seus ciclos de vida; compartilhamento de conhecimentos por meio de padrões e estilos de arquitetura; suporte ao treinamento e à educação nas melhores práticas relacionadas à arquitetura e evolução de sistemas.

No desenvolvimento de *software*, Shaw (1995) destaca que a descrição da arquitetura de um *software* captura a organização do *software* e decisões dos desenvolvedores, facilitando a preservação dessas decisões na manutenção do *software*. A descrição da arquitetura de um *software* tem por finalidade demonstrar que a arquitetura atende requisitos, objetivos, intenções e aspirações dos *stakeholders*; comunicar, de modo efetivo e consistente, aspectos da arquitetura aos *stakeholders*; representar manifestações das primeiras decisões de projeto;

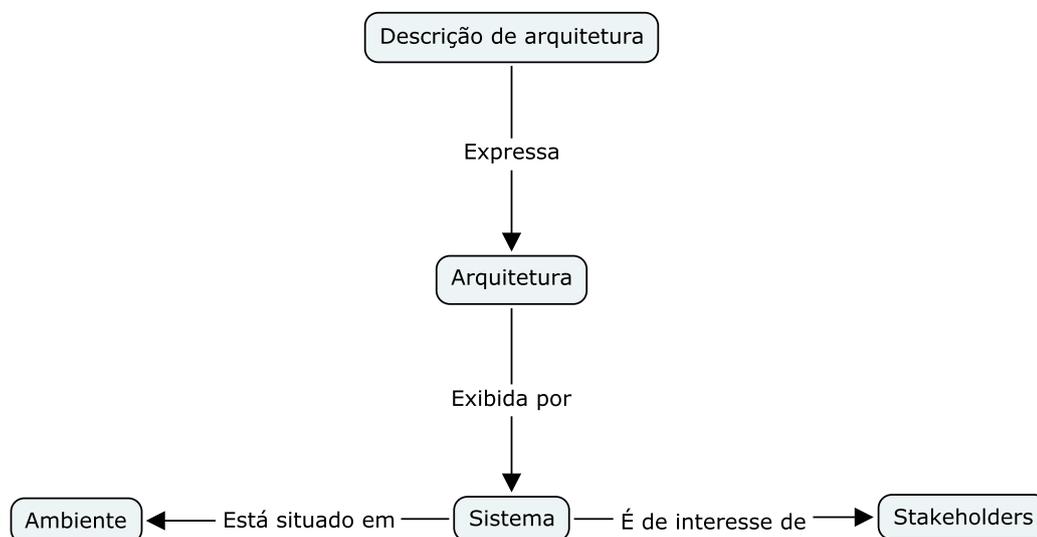


Figura 2 – Contexto de descrição de arquitetura segundo ISO (2011).

facilitar a comunicação entre desenvolvedores; facilitar o entendimento do sistema; expor restrições ao projeto; expor razões de determinadas escolhas; identificar interfaces entre componentes; facilitar o entendimento dos impactos de modificações; facilitar a estimativa dos custos de modificações; separar funcionalidades dos componentes dos mecanismos para conectá-los; prover oportunidades para a análise do sistema; e melhorar o entendimento dos requisitos, da estratégia de desenvolvimento e dos riscos (LUNG; JALNAPURKAR; EL-RAYESS, 1998; GARLAN, 2000; BASS; CLEMENTS; KAZMAN, 2012; ROZANSKI; WOODS, 2011).

## 4.2 Ponto de vista e visão de arquitetura de sistema

A descrição da arquitetura de um sistema pode incluir uma, ou mais, visões (*views*) da arquitetura (ISO, 2011). Cada visão pode ser composta por um ou mais modelos da arquitetura e deve englobar identificação e informações suplementares, de acordo com o que for especificado pela organização e/ou projeto; identificação do ponto de vista que governa a visão; modelos de arquitetura que endereçam os interesses associados ao ponto de vista que governa a visão. Rozanski e Woods (2011) destacam que não é possível capturar todas as funcionalidades e propriedades de sistemas complexos em um só modelo compreensível a todos os interessados. Para esses autores, um sistema complexo pode ser mais bem descrito por meio de visões, que representam um ou mais aspectos da arquitetura e ilustram como

a arquitetura aborda requisitos, objetivos, intenções ou aspirações dos interessados. Entre os benefícios da adoção de visões na descrição de arquiteturas, destacam-se: melhora dos processos de desenho (*design*), análise e comunicação, ao possibilitar que cada aspecto seja focado separadamente; melhora da comunicação com os *stakeholders*, por meio de linguagens e notações apropriadas ao seu público; redução da complexidade, ao tratar separadamente aspectos significativos do sistema; melhora do foco dos desenvolvedores, ao separar os aspectos de interesse dos desenvolvedores.

Para o TOGAF (The Open Group, 2011), uma visão (*view*) é algo visto de um ponto de vista, e pode ser representada por um modelo que demonstre aos *stakeholders* seus interesses na arquitetura. Segundo Lankhorst (2012), um ponto de vista (*viewpoint*) define como construir visões adequadas aos *stakeholders* a partir de conceitos, modelos, técnicas de análise e visualização usadas na construção de visões de uma descrição de arquitetura. O emprego de pontos de vista para expressar a arquitetura de um sistema é premissa da norma ISO/IEC/IEEE 42010 (ISO, 2011). Na figura 3, são apresentados, segundo a norma ISO/IEC/IEEE 42010 (ISO, 2011), relacionamentos entre os seguintes conceitos: descrição de arquitetura, interesse, *stakeholder*, ponto de vista, visão, tipo de modelo e modelo de arquitetura.

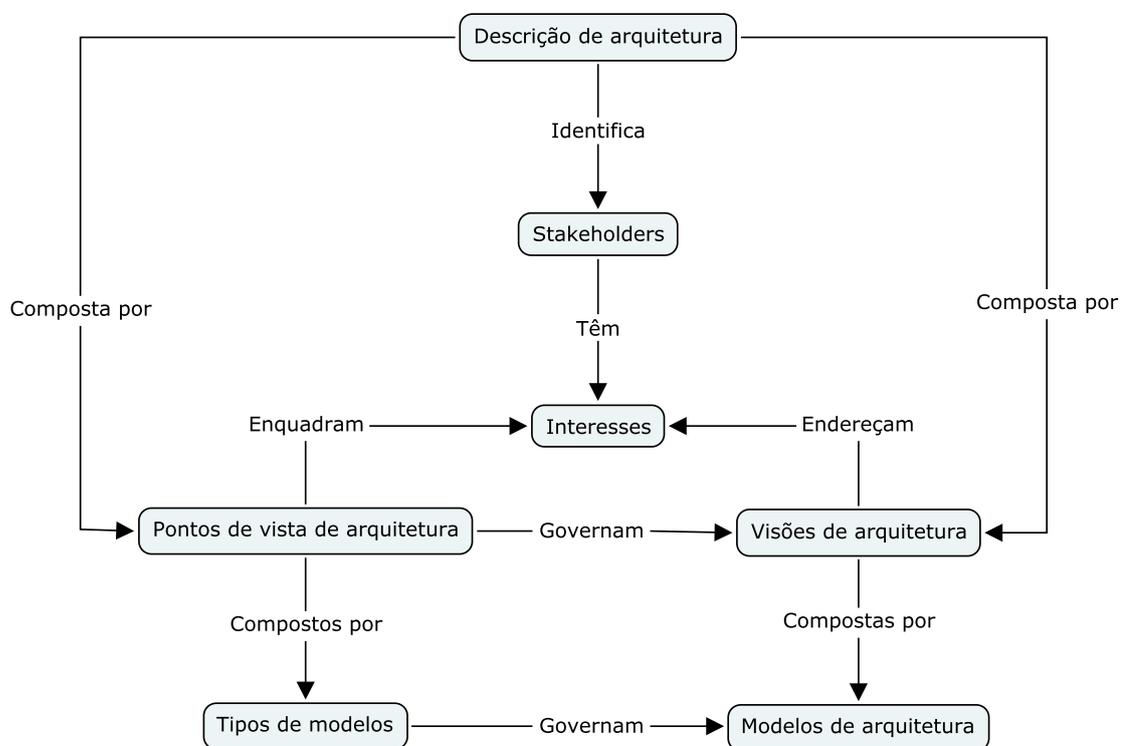


Figura 3 – Modelo conceitual de descrição de arquitetura segundo ISO (2011).

De acordo com a norma ISO/IEC/IEEE 42010 (ISO, 2011), um ponto de vista deve especificar interesses (*concerns*) abordados pelo ponto de vista; *stakeholders* com os interesses relacionados; tipos de modelos associados ao ponto de vista, linguagens, notações, convenções, técnicas de modelagem, métodos e outras operações usadas nos modelos associados ao ponto de vista; e as fontes referenciadas. Rozanski e Woods (2011) definem ponto de vista como coleção de padrões, *templates*, princípios e convenções, empregada na construção de um tipo de visão.

Lankhorst (2012) destaca que o arquiteto deve estar ciente dos interesses dos *stakeholders* quanto à arquitetura e ser capaz de explicar a arquitetura a *stakeholders* com *backgrounds* variados; que diferentes *stakeholders* podem requerer visões com focos específicos, que excluam certas informações. A descrição da arquitetura segundo visões com diferentes níveis de abstração facilita o entendimento da arquitetura e a comunicação entre *stakeholders*.

Greefhorst e Proper (2011) destacam que não existe um consenso acerca das visões que devem ser empregadas na descrição de uma arquitetura, e nem quais são os conteúdos de cada visão. Esses autores citam pesquisas que revelaram a existência de uma variedade de *frameworks* de arquitetura, e o fato de diferentes *frameworks* sugerirem diferentes visões. Por exemplo, o *framework* proposto por Zachman (1987) identifica 36 visões de arquitetura organizadas em seis níveis e seis aspectos. Para auxiliar na seleção de pontos de vista, The Open Group (2012) e Steen et al. (2004) sugerem um *framework* para definição e classificação de pontos de vista e visões baseado nas dimensões propósito e conteúdo.

Com o intuito de exemplificar os conceitos anteriormente descritos, a seguir, são apresentados exemplos no contexto de sistemas de *software*. A documentação da arquitetura de um *software* geralmente contempla diferentes visões, em que cada uma delas representa um aspecto da arquitetura e propriedades específicas do sistema (BUSCHMANN et al., 1996). Em sistemas de *software*, Kazman et al. (1994) sugerem que as visões contemplem funcionalidade, estrutura e alocação. A visão da funcionalidade descreve o que o *software* faz, a da estrutura revela como o *software* é construído de partes conectadas, e a da alocação identifica como as funcionalidades são realizadas na estrutura do *software*. Hofmeister, Nord e Soni (1999) sugerem outras visões: conceitual, módulo, execução e código. A visão conceitual descreve a arquitetura em termos de elementos no domínio para o qual o *software* se destina. Na visão de módulos, os elementos da visão conceitual são mapeados para módulos. A visão de execução descreve a arquitetura em termos de entidades em tempo de execução, enquanto a visão de código descreve como componentes da visão de módulos são mapeados para arquivos, relacionamentos entre arquivos e aspectos relacionados à instalação, gerência de configurações e testes. Kruchten (1995), por sua vez, sugere as visões lógica, de desenvolvimento, de processo e física. Na visão lógica, o *software* é decomposto em classes e objetos, enquanto na visão de desenvolvimento, o *software* é dividido em

unidades organizadas em camadas, subsistemas e módulos. Na visão de processo, as tarefas são atribuídas a entidades existentes em tempo de execução. Por fim, a visão física mapeia os elementos das outras visões para componentes físicos.

### 4.3 Modelo de arquitetura

Um modelo é uma concepção abstrata e não ambígua de algo no mundo real. Dependendo do seu objetivo, um modelo foca certos aspectos ou elementos específicos e ignora outros (LANKHORST, 2012). Em [The Open Group \(2011\)](#) e [The Open Group \(2013\)](#), por sua vez, modelo é definido como uma representação de um assunto de interesse, em escala menor, simplificada e/ou abstrata do assunto tratado, construído como um “meio para um fim”. A norma ISO/IEC/IEEE 42010 (ISO, 2011) destaca que diferentes comunidades empregam o termo modelo de formas distintas. No contexto dessa norma, “M é um modelo de S, se M pode ser usado para responder questões acerca de S”. Ainda segundo essa norma, cada modelo, que pode ser, por exemplo, um conceito (um modelo mental), ou um produto do trabalho. [Souza, Aken e Groesbeck \(2002\)](#) destacam que modelos são simplificações da realidade, que servem como filtros de informações irrelevantes e podem viabilizar o tratamento de assuntos complexos.

No contexto de arquiteturas, [Rozanski e Woods \(2011\)](#) consideram que um modelo é uma representação simplificada ou abstrata de um aspecto de uma arquitetura, e visa comunicar esse aspecto a *stakeholders*. Segundo [Lankhorst \(2012\)](#), modelos de arquitetura são criados para comunicar algo aos leitores do modelo, ou entre pessoas que participam da construção do modelo. Na construção de modelos, os participantes introduzem, concordam e se comprometem com representações do conhecimento. [Lankhorst \(2012\)](#) afirma que modelos não são os únicos produtos do processo de modelagem, e destaca as transformações nos conhecimentos, acordos e compromissos nas mentes das pessoas envolvidas nesse processo. Observa que os processos de modelagem são frequentemente interativos e iterativos, e incluem atividades como: estabelecimento de propósito, escopo e foco; seleção de pontos de vista; construção e estruturação de modelos; visualização de modelos; uso de modelos para comunicação com *stakeholders*; e manutenção de modelos.

### 4.4 Linguagens de modelagem

[Duarte \(2011\)](#) destaca que os modelos podem variar quanto aos objetivos, apresentações, conteúdos, detalhamento e linguagens. Quanto ao objetivo, é descritivo se descreve o que existe e é prescritivo se descreve estados desejados. Quanto à apresentação, pode ser textual, gráfico ou fórmula. Quanto ao conteúdo, pode ser estático ou dinâmico. Quanto ao nível de detalhe, pode ser conceitual ou técnico e quanto à linguagem, pode ser não formal, semiformal ou formal. Segundo [Souza, Aken e Groesbeck \(2002\)](#), existem variadas

alternativas para a construção de modelos, desde a rica e informal linguagem natural humana, até a formal e precisa matemática.

Lankhorst (2012) destaca que o ato de modelar inclui atividades que visam representar o modelo por meio de alguma linguagem e meio. Uma linguagem de modelagem normalmente inclui elementos de modelagem, uma notação visual para apresentação dos elementos e sugestões de como usar a linguagem. Não é necessário, entretanto, que todas as informações de um modelo sejam representadas em diagramas. Algumas dessas informações podem estar em textos, tabelas, gráficos etc. As sintaxe e semântica das linguagens de modelagem tipicamente possibilitam a construção de modelos de forma mais precisa do que uma linguagem natural. Para facilitar o entendimento dos modelos e a comunicação dos componentes do sistema, as notações geralmente são simples, com abstrações visuais.

O termo “linguagem para a descrição de arquitetura” (*architecture description language*) é empregado em comunidades variadas (ISO, 2011). Na arquitetura de *software*, as linguagens para descrição de arquiteturas suportam criação, refino e validação de arquiteturas; representação de estilos de arquitetura; diferentes visões de arquitetura; análise de arquiteturas; e comunicação da arquitetura entre *stakeholders* (BASS; CLEMENTS; KAZMAN, 2012; CLEMENTS, 1995; GARLAN; KOMPANEK, 2002). Rozanski e Woods (2011) destacam a existência de várias linguagens para descrição de arquiteturas, mas observam que a maioria encontra-se no campo das pesquisas.

Embora os arquitetos precisem expressar arquiteturas para o seu entendimento e para a comunicação com *stakeholders*, Lankhorst (2012) afirma que ainda não existe uma linguagem padrão para descrição precisa de arquiteturas em diferentes domínios. Os arquitetos em diferentes domínios tendem a empregar suas próprias técnicas e convenções na descrição de arquiteturas. Khoury (2007) destaca que, embora existam muitas linguagens para a modelagem de sistemas, poucas têm o poder semântico necessário ao uso em domínios diversos. Além disso, a complexidade de algumas dessas linguagens torna impraticável seu emprego em aplicações comerciais. Em decorrência disso, são frequentemente empregadas várias linguagens na descrição de uma arquitetura, inclusive linguagens informais de modelagem. Para Khoury (2007), na modelagem de empresas, o uso de várias linguagens aumenta a probabilidade dos modelos serem incompletos, inconsistentes e difíceis de entender. Nos próximos tópicos desta dissertação, serão apresentadas as linguagens *Unified Modeling Language* e *ArchiMate*, linguagens que podem ser empregadas na descrição de arquiteturas.

#### 4.4.1 *Unified Modeling Language*

Os potenciais benefícios de uma linguagem de modelagem padronizada motivaram o desenvolvimento da *Unified Modeling Language* (UML) a partir de 1994. Esse desenvolvimento foi iniciado por Grady Booch e Jim Rumbaugh visando unificar os métodos Booch

e Object Modeling Technique (OMT). Em 1995, Ivar Jacobson integrou-se ao grupo e o método Object-Oriented Software Engineering (OOSE) foi unificado. Em 1996, o trabalho, inicialmente desenvolvido na Rational Software Corporation, resultou nas versões 0.9 e 0.91 da UML. Em 1996, o Object Management Group (OMG) solicitou propostas para uma notação padrão para a modelagem orientada a objetos. A UML versão 1.0 foi proposta em 1997. Foi então criado um consórcio de interessados na UML, os trabalhos desse consórcio resultaram no documento UML version 1.1. Ainda em 1997, o OMG incluiu a UML entre as tecnologias suportadas e tornou-se responsável pela mesma (BOOCH; RUMBAUGH; JACOBSON, 2005).

A *Unified Modeling Language* (UML) é uma linguagem de modelagem para análise, desenho (*design*) e implementação de sistemas de *software*, bem como para modelagem de negócios e processos similares (OMG, 2011). Os modelos construídos com a UML usam classes, objetos, interfaces, componentes, pacotes, relacionamentos e estereótipos. As visões que descrevem um sistema podem ser documentadas com diagramas construídos com a UML.

Os diagramas construídos com a UML podem ser inter-relacionados e usados para se modelar diferentes aspectos de um sistema. Os diagramas podem prover diferentes perspectivas de um sistema e podem ser usados com outros documentos em um modelo. Por meio de diagramas podem ser representados aspectos estáticos e dinâmicos de um sistema. Por exemplo, diagramas de classe ou diagramas de objeto podem ser construídos com o intuito de representar aspectos estáticos de um sistema, enquanto diagramas de colaboração podem ser construídos com o intuito de representar aspectos dinâmicos desse mesmo sistema.

Para adequar a UML às necessidades de um domínio, podem se acrescentadas extensões, por meio da especificação de restrições a serem impostas a elementos do modelo. Rozanski e Woods (2011) destacam que adaptar e especializar linguagens de uso geral, como a UML, é uma alternativa ao uso de linguagens especificamente desenvolvidas para descrição de arquiteturas. Usar a linguagem UML sem modificá-la, aplicar restrições ao metamodelo da UML usando mecanismos para extensão da linguagem UML e expandir o metamodelo da UML para suportar os conceitos necessários na descrição de arquiteturas são estratégias para uso da UML na descrição de arquiteturas (MEDVIDOVIC et al., 2002).

#### 4.4.2 Linguagem ArchiMate

A linguagem ArchiMate complementa o *framework* TOGAF (The Open Group, 2012). ArchiMate é uma linguagem de modelagem visual que pode ser empregada na especificação de arquiteturas inter-relacionadas e na especificação de diferentes pontos de vista. O núcleo da linguagem encontra-se relacionado às seguintes fases em TOGAF

ADM: “Arquitetura de Negócio”, “Arquitetura de Sistemas de Informação” e “Arquitetura Tecnológica”.

Lankhorst (2012) destaca a importância do conceito serviço em ArchiMate. Um serviço é uma unidade de funcionalidade que um sistema expõe ao ambiente, ao mesmo tempo em que esconde operações internas (The Open Group, 2012). Para Lankhorst (2012), a orientação a serviço leva a uma organização dos modelos em camadas, onde serviços em uma camada podem ser prestados a outras camadas, e onde podem existir serviços internos às camadas. Segundo Lankhorst (2012), uma premissa da linguagem ArchiMate é que a estrutura genérica dos modelos em diferentes camadas é similar, são usados os mesmos conceitos e relacionamentos, o que facilita o alinhamento entre modelos em diferentes camadas.

Em The Open Group (2012) são definidas as seguintes camadas onde são organizadas as descrições elaboradas com a linguagem ArchiMate: Negócio, Aplicação e Tecnologia. A camada Negócio provê produtos e serviços a clientes externos, esses serviços são realizados por processos de negócio desempenhados por atores. A camada Aplicação provê, à camada Negócio, serviços realizados por aplicações de *software*. Por sua vez, a camada Tecnologia provê serviços de infraestrutura necessários às aplicações de *software* na camada Aplicação.

## 5 Sistemas para organização da informação

Neste capítulo, são apresentados conceitos acerca de organização da informação e são descritos sistemas para organização da informação. Sistemas para organização da informação podem ser componentes integrantes de *frameworks* de arquitetura da informação.

### 5.1 Conceitos acerca de organização da informação

Para uma informação atingir o seu propósito, precisa ser encontrada, apresentada a quem a necessita, no momento em que é necessária e em forma apropriada (MARTIN; DMITRIEV; AKEROYD, 2010). Para facilitar o compartilhamento e a recuperação das informações, é necessário organizá-las e armazená-las sistematicamente (CHOO, 2002). Organizar informações envolve indexá-las e classificá-las visando facilitar a recuperação das mesmas (DETLOR, 2010). Em alguns domínios, a organização da informação é denominada organização do conhecimento (DETLOR, 2010). Anderson (2003) considera que o termo “organização do conhecimento” é uma abreviação de “organização de recursos de conhecimento” ou “organização de representações do conhecimento”, e sinônimo de “organização da informação”. Segundo Anderson (2003), a organização do conhecimento é a descrição e organização, para recuperação, de mensagens representando conhecimentos, de textos onde conhecimentos são registrados e de documentos onde textos encontram-se contidos.

A organização da informação pode considerar aspectos semânticos (significados das informações). Quando a organização semântica não faz sentido, as informações podem ser organizadas em ordem alfabética, cronológica etc. Quando são empregados aspectos semânticos, a organização depende de como se deseja conceituar as categorias de informação. Os relacionamentos semânticos entre termos são definidos de acordo com os significados dos termos, e podem ser de equivalência, hierárquico ou associativo. É de equivalência se os termos têm o mesmo significado, hierárquico se o escopo de um termo encontra-se no de outro, e associativo se os termos estão associados de um modo que o relacionamento possa sugerir alternativas na indexação ou recuperação da informação (ROWLEY; HARTLEY, 2008).

#### 5.1.1 Indexação, catalogação e classificação

Na organização da informação, é importante a indexação, a catalogação e a classificação dos documentos. (ANDERSON, 2003). Anderson (2003) destaca que a indexação

é fundamental para a organização dos recursos de informação, e que a catalogação e a classificação dependem da indexação. A *National Information Standards Organization* (NISO, 2005) destaca que a indexação envolve a seleção de termos preferidos de um ou mais vocabulários controlados, ou de outras fontes, para descrever objetos de conteúdo. Nesse contexto, objeto de conteúdo é qualquer item a ser incluído em um sistema de recuperação de informação, sítio, ou outra fonte de informação (NISO, 2005). Para Fujita (2003), o desempenho da indexação reflete-se na recuperação da informação. Segundo a norma ISO 5963-1985 (ISO, 1985), a indexação envolve exame do documento e estabelecimento do assunto de seu conteúdo; identificação dos principais conceitos no assunto; e tradução dos conceitos em termos de uma linguagem de indexação. Essa norma recomenda que, na indexação, sejam escolhidos conceitos úteis e adequados aos usuários do sistema de informação.

Na indexação, os conceitos são analisados e representados por termos derivados da linguagem natural ou por símbolos de classificação. Uma vez identificados os conceitos, os já representados na linguagem de indexação são traduzidos em termos preferidos. Para os conceitos ainda não representados na linguagem de indexação, podem ser definidos e avaliados termos a partir de consultas a dicionários, enciclopédias, tesouros, esquemas de classificação e especialistas. Alternativamente, os termos usados pelo autor do documento podem ser usados na indexação. A norma ISO 5963-1985 (ISO, 1985) sugere que não seja previamente estabelecido limite para a quantidade de termos, a qual depende da quantidade de informações e das necessidades dos usuários do sistema de informação. Essa norma destaca ainda que a qualidade e a consistência da indexação dependem de quem realiza a indexação (capacidade, imparcialidade, conhecimento dos campos cobertos pelos documentos indexados, entendimento das regras e procedimentos da linguagem de indexação), da qualidade das ferramentas de indexação, da possibilidade de contato direto do indexador com os usuários, e da facilidade com que a linguagem de indexação possibilita modificações na terminologia.

### 5.1.2 Metadados

Na organização da informação, metadados podem ser empregados. Os metadados são informações estruturadas associadas a recursos de informação, que descrevem, explicam e localizam esses recursos, ou facilitam a recuperação, o uso ou a gestão desses recursos. (NISO, 2004). Os metadados são frequentemente definidos como dados acerca de dados, ou informações acerca de informações. São compostos por fatos e significados que descrevem, explicam e localizam recursos de informação. Em bibliotecas, metadado designa qualquer esquema formal de descrição de recurso, aplicável a qualquer tipo de objeto, digital ou não (NISO, 2004). Metadados podem descrever coleções ou componentes de coleções, assim como podem descrever recursos em diferentes níveis do modelo *Functional Requirements*

for *Bibliographic Records* proposto pela *International Federation of Library Associations* (IFLA).

Os metadados podem facilitar a descoberta, a recuperação, o uso, o gerenciamento, a identificação, a descoberta, a seleção, o acesso, o arquivamento, o entendimento, o compartilhamento, a preservação e a organização dos recursos de informação, além de promover a interoperabilidade entre sistemas (NISO, 2004; NISO, 2007; TAYLOR; JOUNDRY, 2009; FINKELSTEIN; AIKEN, 2000). Podem ser criados por profissionais especializados, autores de conteúdos ou usuários. Metadados podem ser descritivos, estruturais ou administrativos. Os descritivos facilitam a descoberta e a identificação dos recursos de informação. Os estruturais contêm informações sobre objetos compostos, como páginas em capítulos de livros. Os administrativos contêm informações para facilitar a gestão dos recursos de informação.

Existem diversos esquemas de metadados para descrever determinados tipos de recursos de informação necessários a certas comunidades, como por exemplo: *Dublin Core Metadata Element Set*, *Text Encoding Initiative* (TEI), *Metadata Encoding and Transmission Standard* (METS), *Metadata Object Description Schema* (MODS), *Learning Object Metadata* (LOM), *Online Information Exchange* (ONIX), *Categories for Description of Works of Art* (CDWA), *Machine Readable Cataloging* (MARC), *Moving Picture Expert Group* MPEG-7 e MPEG-21. Segundo a *National Information Standards Organization* NISO (2007), a escolha de qual esquema empregar deve levar em conta os objetivos da coleção digital; quais os usuários alvo; quais informações são necessárias aos usuários; qual o comportamento dos usuários na busca à informação; o nível no qual os materiais serão acessados (coleção, item ou ambos); necessidade de distinguir múltiplas versões ou manifestações dos objetos; a existência de metadados associados à coleção, ou aos seus objetos, antes da coleção ser construída; disciplina envolvida e padrões de metadados comumente usados na disciplina; padrões de metadados usados por organizações no domínio; padrões de metadados mais apropriados para a coleção em particular; quão rica precisa ser a descrição; a necessidade dos metadados informarem acerca de relacionamentos hierárquicos.

## 5.2 Sistemas para organização da informação

Sistemas para organização de informações geralmente agrupam informações em categorias, por meio de esquemas de classificação e definição de estruturas. Um esquema de classificação define características compartilhadas pelos conteúdos, enquanto uma estrutura define relacionamentos entre itens e grupos. Os esquemas para organizar informações podem ser exatos ou ambíguos. Um esquema exato divide as informações em grupos definidos e mutuamente exclusivos, tais como o esquema alfabético. Um esquema ambíguo, por sua

vez, classifica as informações em categorias sem definições exatas, como por exemplo, tipos de audiências às quais se destinam as informações (SPENCER; FEATHERSTONE, 2010; MORVILLE; ROSENFELD, 2006).

### 5.2.1 Vocabulário controlado

A eficácia da indexação na identificação e recuperação de informações depende de linguagens de indexação. As linguagens de indexação são compostas por vocabulários controlados e por regras para o uso dos mesmos. Nas linguagens de indexação, termos são representações de conceitos (NISO, 2005). As linguagens de indexação podem ser controladas ou naturais (ROWLEY; HARTLEY, 2008). Nas naturais, termos usados pelos autores dos documentos são empregados na indexação, quaisquer termos no recurso indexado podem ser empregados. As escolhas de quais termos fazem parte da linguagem de indexação, pode se basear, por exemplo, na análise das frequências dos termos (ROWLEY; HARTLEY, 2008).

Nas linguagens de indexação controladas, cada termo é designado como preferido (*preferred*) ou não preferido (*non-preferred*). Sendo um termo preferido, aquele termo que é usado consistentemente para representar um conceito na indexação. Um termo não preferido é aquele sinônimo, ou quase sinônimo, de um preferido. Os termos não preferidos não são atribuídos aos documentos, são usados como pontos de entrada em índices (ISO, 1985).

Os vocabulários controlados são listas fechadas de termos, em que cada termo é uma ou mais palavras usadas para representar um conceito (NISO, 2005). Um termo, por sua vez, é um rótulo para um conceito, que pode ser descrito por múltiplos termos (HEDDEN, 2010). O vocabulário é controlado, pois apenas os termos na lista podem ser usados para a área de assunto coberta (HEDDEN, 2010). Além disso, se o vocabulário for usado por mais de uma pessoa, é necessário controlar quem pode acrescentar termos, quando e como isso pode ser feito (HEDDEN, 2010). A necessidade de vocabulários controlados decorre de características das linguagens naturais (linguagens usadas por seres humanos para comunicação verbal), tais como existência de palavras ou termos que, apesar de distintos, representam um mesmo conceito; existência de palavras que, apesar de soletradas da mesma forma, representam conceitos distintos (palavras denominadas homógrafas) (NISO, 2005).

Nos sistemas de armazenamento e recuperação da informação, os vocabulários controlados podem, ou não, ser visíveis aos usuários. Os vocabulários controlados podem ser usados para organizar e indexar informações; a partir dos relacionamentos entre termos, sugerir termos de interesse nas consultas dos usuários, tornando-as mais genéricas ou específicas; prover suporte na busca e navegação; auxiliar no processo de navegação em sítios por meio de menus de navegação ou mapas de sítios; prover consistência na indexação

e rotulação; evitar ambiguidades em pesquisas; converter linguagens naturais em linguagens para indexação e recuperação; uniformizar termos; indicar relacionamentos semânticos entre termos; construir hierarquias para auxiliar usuários a localizarem conteúdos; auxiliar na pesquisa e localização de conteúdos; definir metadados; prover consistência na elaboração de textos em áreas técnicas (HEDDEN, 2010; NISO, 2005; ROWLEY; HARTLEY, 2008).

São princípios de desenvolvimento de vocabulários controlados: eliminação de ambiguidades; controle de sinônimos; estabelecimento, quando apropriado, de relacionamentos entre termos; teste e validação de termos (NISO, 2005). Nos vocabulários controlados, a definição dos termos deve ser não ambígua e não redundante. Se um termo identifica diferentes conceitos, exceto nos anéis de sinônimos, a ambiguidade é resolvida qualificando-se o termo. Se mais de um termo identifica um conceito, um termo deve ser identificado como preferido e os outros como sinônimos. Para desenvolver vocabulários controlados é preciso definir escopo e objetivos do vocabulário; definir quem pode acrescentar termos; definir quando e como os termos podem ser acrescentados; definir significados e escopos dos termos; distinguir entre termos homógrafos; selecionar padrões aos quais deve o vocabulário aderir; analisar vocabulários controlados existentes; identificar termos para conceitos; construir o vocabulário; estabelecer relacionamentos entre termos; estabelecer relacionamentos associativos entre sinônimos e quase sinônimos (termos não sinônimos, mas equivalentes no vocabulário controlado); revisar o vocabulário com a participação dos interessados; atualizar vocabulário; integrar o vocabulário ao sistema em desenvolvimento (HEDDEN, 2010; NISO, 2005).

Em função dos controles requeridos, os vocabulários controlados podem ser estruturados como lista de controle, anel de sinônimos, arquivo de autoridade, taxonomia e tesauro. Segundo a norma ANSI/NISO Z39.19-2005 (NISO, 2005), em ordem crescente de complexidade estrutural, se tem: lista, anel de sinônimos, taxonomia e tesauro. Ainda segundo essa norma, os seguintes controles são possíveis: listas controlam ambiguidades; anéis de sinônimos controlam sinônimos; taxonomias controlam ambiguidades, sinônimos e relacionamentos hierárquicos; e tesauros controlam ambiguidades, sinônimos, relacionamentos hierárquicos e associativos. Warner (2002) destaca os seguintes vocabulários em níveis crescentes de controle: sinônimos ou termos equivalentes; arranjo dos termos em hierarquias, dos genéricos para os específicos; determinação de outros relacionamentos de associação entre termos.

#### 5.2.1.1 Lista de termos

Na sua forma mais simples, um vocabulário controlado é uma lista restrita de termos, construída com um objetivo específico, por exemplo, indexação, rotulação ou categorização (HEDDEN, 2010). As listas de termos consistem em conjuntos de itens organizados segundo algum princípio lógico evidente, e são usadas para descrever conceitos

que apresentam possibilidades limitadas (NISO, 2005). Em uma lista de termos, os termos podem, ou não, seguir alguma ordem definida. São exemplos de listas de termos: glossários, dicionários e arquivos de autoridade. Os glossários são listas de termos, com definições empregadas em um campo ou em um trabalho específico, os dicionários são listas, em ordem alfabética, de termos com as suas definições, enquanto os arquivos de autoridade são vocabulários controlados empregados para controlar nomes de entidades ou valores de domínio para campos (GAIL, 2000). Arquivos de autoridade podem ser criados para relacionar nomes de pessoas, locais, países, produtos, organizações e trabalhos publicados (HEDDEN, 2010). Nos arquivos de autoridade, podem existir termos preferidos e variações dos mesmos.

#### 5.2.1.2 Anel de sinônimos

Um anel de sinônimos é um vocabulário controlado composto por termos equivalentes sob o ponto de vista da recuperação da informação (NISO, 2005). Nos anéis de sinônimos, não são designados termos preferidos e os sinônimos são organizados em um anel circular (HEDDEN, 2010). Podem ser estabelecidos relacionamentos entre palavras com um mesmo significado, entre palavras com significados semelhantes, entre acrônimos, entre palavras com variações no modo como são soletradas, entre termos científicos e termos populares. O controle de sinônimos tem o intuito de minimiza as variantes introduzidas pela linguagem natural.

Os anéis de sinônimo são empregados na recuperação da informação em interfaces de sistemas de informação e não na indexação (NISO, 2005). São definidos quais termos no conteúdo devem ser considerados equivalentes, e o controle é realizado por meio da interface de pesquisa. Um anel de sinônimos possibilita que um conceito descrito por termos, sinônimos sob o ponto de vista da recuperação da informação, seja recuperado por quaisquer desses termos. Anéis de sinônimos podem aumentar a relação entre a quantidade de documentos relevantes recuperados e o total de documentos relevantes (*recall ratio*), mas pode também reduzir a relação entre a quantidade de documentos relevantes recuperados e o total de documentos recuperados (*precision ratio*) (MORVILLE; ROSENFELD, 2006).

#### 5.2.1.3 Taxonomia hierárquica

Segundo o glossário da *Society of American Archivists* (SAA, 2014), taxonomia é uma estrutura para classificar materiais em uma hierarquia de categorias e subcategorias. Na norma ANSI/NISO Z39.19-2005 (NISO, 2005), taxonomia é definida como vocabulário controlado composto por termos preferidos, todos conectados em uma hierarquia ou poli-hierarquia. Hedden (2010) observa que o termo taxonomia pode designar sistemas de classificação hierárquicos ou, em sentido mais amplo, meios para organizar conhecimentos. Para esse autor, taxonomia designa meios para organizar conhecimentos, enquanto

taxonomia hierárquica designa vocabulário controlado com relacionamentos entre termos que identificam conceitos mais ou menos abrangentes, podendo ser representada por uma árvore.

#### 5.2.1.4 Tesouro

Um tesouro é um vocabulário controlado em que os termos podem apresentar relacionamentos hierárquicos, associativos ou de equivalência. Segundo a norma ANSI/NISO Z39.19-2005 (NISO, 2005), um tesouro é um vocabulário controlado disposto em uma ordem conhecida, e estruturado de modo que os relacionamentos entre os termos sejam claramente apresentados e identificados por indicadores padronizados de relacionamento, devendo os relacionamentos entre termos ser empregados de modo recíproco. Quando os vocabulários são extensos, sistemas de informações podem empregar tesouros para facilitar a recuperação de informações, pois o tesouro possibilita indexação mais precisa do que a taxonomia hierárquica. Os tesouros, além de facilitar a localização de informações, podem facilitar o entendimento das terminologias nas áreas de conhecimento para as quais foram desenvolvidos (HEDDEN, 2010). Nos tesouros, cada termo não precisa fazer parte de uma hierarquia, e podem existir várias pequenas hierarquias. Quando existem hierarquias significativas, a diferença entre tesouros e taxonomias hierárquicas torna-se menos clara (HEDDEN, 2010).

#### 5.2.2 Ontologia

O termo ontologia, cunhado em 1613 por Rudolf Göckel e Jacob Lorhard (SMITH, 2003), tem várias definições, algumas contraditórias (NOY; MCGUINNESS, 2001). Como ramo da filosofia, a ontologia procura prover uma classificação definitiva e exaustiva das entidades em todas as esferas do ser (SMITH, 2003). Corcho, Fernández-López e Gómez-Pérez (2006) destacam a necessidade de diferenciar uma ontologia, de Ontologia. Segundo esses autores, uma ontologia é uma classificação de categorias, enquanto Ontologia é um ramo da Filosofia. Guarino (1998) destaca que o termo ontologia pode ser usado com diferentes sentidos por diferentes comunidades. No sentido filosófico, é possível considerar que ontologia consiste em um sistema de categorias que representa certa visão do mundo (GUARINO, 1998). Para Uschold (1998), embora uma ontologia possa assumir várias formas, necessariamente inclui vocabulário de termos e especificações dos seus significados.

Segundo alguns autores, uma ontologia é um modo formal de descrever e representar conceitos e relacionamentos em uma área de conhecimento. Para Gruber (1993), uma ontologia é uma especificação explícita de uma conceituação, onde conceituação é uma visão simplificada e abstrata do mundo que se deseja representar para certo fim. Quanto ao termo conceituação, Borst (1997) o define como sendo uma interpretação estruturada de parte do mundo, usada para pensar e comunicar acerca do mundo, enquanto Uschold

(1998) define o termo conceituação como sendo uma visão do mundo, uma forma de pensar acerca de um domínio.

Para Borst (1997), uma ontologia pode ser definida como uma especificação formal de uma conceituação compartilhada. Borst (1997) também destaca que uma ontologia é uma especificação formal de uma parte de uma conceituação, pois define termos em um domínio e relacionamentos entre esses termos, mas não define os significados dos termos, sendo assumido que os significados são conhecidos pelo grupo para o qual a ontologia é desenvolvida. Segundo Fox e Gruninger (1997), uma ontologia consiste de uma descrição formal de entidades e de suas propriedades, forma uma terminologia compartilhada para objetos de interesse em um domínio, com definições para os significados de cada termo.

Corcho, Fernández-López e Gómez-Pérez (2006) destacam a definição de Studer, Benjamins e Fensel (1998), segundo a qual, uma ontologia é uma especificação formal e explícita de uma conceituação compartilhada. O termo formal refere-se ao fato de que a ontologia deve poder ser lida por máquina, o que exclui a linguagem natural. O termo compartilhado reflete a noção de que uma ontologia captura conhecimento consensual, aceito por um grupo.

Os componentes de uma ontologia são os seguintes: classes, também denominadas conceitos; propriedades das classes, que definem atributos das classes e relacionamentos entre elas; restrições a propriedades; e instâncias de classes (NOY; MCGUINNESS, 2001; ROWLEY; HARTLEY, 2008). Segundo Corcho, Fernández-López e Gómez-Pérez (2006), em uma ontologia, o conjunto mínimo de componentes é composto por classes, relacionamentos e instâncias. As classes representam conceitos, os relacionamentos representam associações entre conceitos, e as instâncias representam os elementos ou os indivíduos na ontologia.

Guarino (1998) classifica as ontologias, quanto ao nível de generalidade, como ontologias de alto nível (*top-level ontologies*), ontologias de domínio, ontologias de tarefa e ontologias de aplicação. As ontologias de alto nível descrevem conceitos genéricos, que independem de um problema ou domínio particular. As ontologias de domínio e as de tarefa descrevem vocabulários relacionados a domínios genéricos ou a tarefas genéricas, a partir da especialização de termos em ontologias de alto nível. Finalmente, as ontologias classificadas como ontologias de aplicação descrevem conceitos que pertencem simultaneamente a um domínio e a uma tarefa, pela especialização de conceitos em ontologias de domínio e de tarefa.

Borst (1997) destaca que o compartilhamento e o reuso de conhecimento é mais proveitoso em domínios grandes e complexos. Uma vez que esses domínios tendem a resultar em ontologias grandes e complexas, Borst (1997) sugere que sejam divididos em subdomínios com forte coerência interna e fraco acoplamento, e que os subdomínios sejam especificados em ontologias separadas. Borst (1997) sugere a decomposição de ontologias

grandes e complexas por meio da divisão do domínio em partes com conceitos centrados em certos pontos de vista; construção de ontologias que definam conceitos abstratos que possam ser usados na definição de conceitos específicos em outros domínios; e construção de ontologias que definam como o conhecimento no domínio pode ser empregado em certas tarefas.

### 5.2.2.1 Desenvolvimento de ontologias

O desenvolvimento e uso de ontologias é um meio de entender e organizar informações em um domínio, para esclarecer a natureza de conceitos e termos; reduzir ambiguidades conceituais e terminológicas; facilitar o compartilhamento de conhecimentos e a comunicação entre pessoas, e entre sistemas; facilitar a conversão de conhecimento implícito em explícito; facilitar o entendimento de conceitos e dos seus relacionamentos (CORCHO; FERNÁNDEZ-LÓPEZ; GÓMEZ-PÉREZ, 2006; DUDDUKURI; PRABHAKAR, 2006).

Corcho, Fernández-López e Gómez-Pérez (2006) relacionam os seguintes métodos para o desenvolvimento de ontologias: Cyc; Uschold e King; Grüninger e Fox; KACTUS; METHONTOLOGY; SENSUS; On-To-Knowledge. Para esses autores, nenhum dos métodos abrange todas as atividades na construção de ontologias; a maioria deles foca nas atividades de desenvolvimento, dando pouca atenção a aspectos relacionados ao gerenciamento, evolução e avaliação de ontologias; alguns métodos foram usados apenas em um domínio; e a maioria dos métodos não tem suporte de ferramentas. Rowley e Hartley (2008) sugerem como etapas no desenvolvimento de ontologias: determinar domínio e escopo; identificar ontologias existentes; identificar conceitos e termos; definir classes, relacionamentos entre classes e propriedades; escolher linguagem de representação; criar instâncias de classes; estabelecer convenções de nomes; testar ontologia; documentar; revisar e atualizar.

Segundo Corcho, Fernández-López e Gómez-Pérez (2006), na medida em que ontologias se tornem disponíveis e aumente o reuso, ocorrerão cenários como avaliação e importação de ontologias; integração do modelo conceitual de uma ontologia ao de outra sendo desenvolvida; avaliação, importação e alinhamento de ontologias; avaliação, importação e fusão de ontologias; tradução dos termos de uma ontologia em linguagem natural; gerenciamento da evolução de ontologias colaborativamente construídas. Noy e Musen (2000) destacam que, para ontologias serem reusadas, pode ser necessário estabelecer correspondências entre ontologias, determinar sobreposições entre conceitos similares, mas com diferentes nomes ou estruturas, e determinar conceitos únicos. Segundo Noy e Musen (2000), quando o objetivo é criar uma ontologia com as informações de ontologias reusadas, ocorre o processo de fusão (*merging*); quando o objetivo é estabelecer consistência e coerência entre ontologias reusadas, mantendo-as separadas, ocorre o processo de alinhamento (*alignment*).

### 5.2.2.2 Ontologias na engenharia de *software*

Ontologias podem ser aplicadas em domínios variados. Na engenharia de *software*, as ontologias podem ser empregadas na indexação, estruturação e organização das informações, em contextos como análise e desenho; engenharia de requisitos; reuso, implementação e documentação de código; linguagens de modelagem; suporte à implantação e operação de *software*; representação de regras de negócio; manutenção corretiva e evolutiva; testes (CALERO; RUIZ; PIATTIN, 2006; ZHAO; DONG, 2009; DECKER et al., 2005; HAPPEL; SEEDORF, 2006). São potenciais benefícios do emprego de ontologias nesses contextos: reduzir ambiguidades; aumentar a probabilidade dos participantes compartilharem entendimentos acerca do domínio da aplicação; facilitar a localização e o reuso de componentes de *software* (HAPPEL; SEEDORF, 2006). Em comunidades virtuais voltadas ao desenvolvimento de *software* aberto, Ankolekar, Herbsleb e Sycara (2003) destacam a importância das ontologias na integração de informações e na descrição de estruturas de códigos e domínios. No desenvolvimento distribuído de *software*, ontologias podem contribuir na definição de conceitos, tarefas, modelos e processos (WONGTHONGTHAM et al., 2006).

### 5.2.3 Classificação social

A popularização da Internet fez com que as informações deixassem de ser organizadas apenas por profissionais especializados e passassem a ser organizadas pela população em geral (SVENONIUS, 2000). A classificação social, também chamada folksonomia (*folksonomy*) ou classificação distribuída (HAMMOND et al., 2005), consiste na construção colaborativa de uma lista de termos para indexação. Na classificação social, os usuários etiquetam os conteúdos com termos que consideram importantes. A classificação social é uma abordagem ascendente, em contraste com métodos tradicionais de organização da informação.

Na classificação social, as etiquetas têm formatos livres, não são obtidas de vocabulários controlados. Uma vez que os termos escolhidos pela comunidade refletem vocabulários dos usuários, os termos tendem a ser os empregados pelos usuários em consultas. A classificação social possibilita a organização de informações para uso pessoal e pode facilitar a comunicação e o compartilhamento de informações (MATHES, 2004). A classificação social tende a ser mais dinâmica do que outros sistemas empregados na organização de informações, pois as listas de termos criadas na classificação social são abertas e continuamente modificadas.

Entretanto, alguns autores criticam a classificação social por considerarem que ela produz espaços com estruturas planas, não hierárquicas. Além disso, mais de uma etiqueta pode ser associada a um conteúdo, não havendo, necessariamente, relacionamentos claros entre essas etiquetas (MATHES, 2004). Na classificação social, não ocorre o controle de sinônimos e homônimos (ROWLEY; HARTLEY, 2008). Ambiguidades também surgem

quando uma mesma etiqueta é empregada para identificar diferentes conceitos; quando diferentes etiquetas são usadas para identificar um mesmo conceito; e quando são empregadas siglas ou abreviações.

## 6 Frameworks de gestão de projetos

Neste capítulo, são apresentados conceitos acerca de projeto, ciclo de vida de projeto, processo, processo de negócio, *frameworks* e métodos de gestão de projetos.

### 6.1 Projeto

Segundo o PMBOK (PMI, 2013), um projeto é um empreendimento temporário para criar um produto, serviço ou resultado único. Um projeto tem uma natureza temporária, tem início e fim definidos e seu resultado pode ser, ou não, tangível (PMI, 2013). Em OGC (2009), as seguintes características associadas a projetos são relacionadas: projetos são meios pelos quais mudanças são introduzidas; projetos tem natureza temporária; projetos tipicamente envolvem equipes de pessoas com diferentes habilidades; embora possam existir projetos similares, cada projeto é único; projetos envolvem riscos. Um projeto pode envolver apenas um indivíduo, múltiplos indivíduos, apenas uma unidade de uma organização, múltiplas unidades de uma organização ou múltiplas organizações (PMI, 2013). O produto de um projeto pode ser um componente de um item, uma melhoria a um item, ou um item (PMI, 2013).

### 6.2 Ciclo de vida de projeto

O ciclo de vida de um projeto é composto por fases pelas quais o projeto passa entre o início e o final do projeto. Em PMBOK (PMI, 2013) é descrito um ciclo de vida genérico com as seguintes fases: iniciando o projeto, organizando e preparando, realizando o trabalho do projeto, encerrando o projeto. Larson e Gray (2010) sugerem um ciclo de vida genérico com as seguintes fases: definição, planejamento, execução e fechamento. Por sua vez, em OGC (2009) é descrito um ciclo de vida com as seguintes fases: pré-projeto, iniciação, fases de entrega, fase de entrega final.

Existem diferentes modelos de ciclo de vida de projetos. Os ciclos de vida de projetos podem ser caracterizados, por exemplo, por meio de modelos como preditivo, iterativo e incremental, adaptativo (PMI, 2013). No modelo de ciclo de vida iterativo e incremental, o ciclo de vida do projeto é decomposto em fases onde ocorrem iterações logicamente relacionadas que resultam em uma ou mais entregas (PMI, 2013). Em cada fase do ciclo de vida de um projeto, iterações ocorrem até ser alcançado o critério de saída da fase. No modelo de ciclo de vida iterativo e incremental, atividades podem ser intencionalmente repetidas (PMI, 2013). A figura 4 ilustra uma fase onde ocorrem iterações, algumas seriais, outras concorrentes.

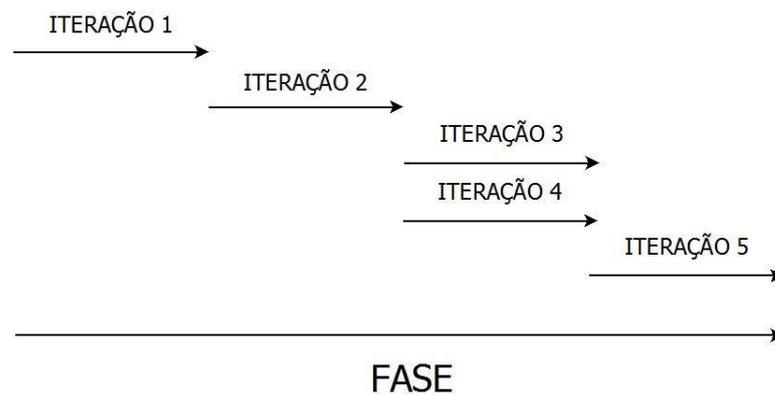


Figura 4 – Fase com iterações seriais e concorrentes.

### 6.3 Processo

Projetos são conduzidos executando-se processos. Segundo o PMBOK (PMI, 2013), um processo é uma série sistemática de atividades direcionadas a causar um resultado final, sendo cada atividade uma porção distinta e agendada de trabalho realizada no curso de um projeto. A ISACA (2011) conceitua processo como uma coleção de atividades que recebe uma ou mais entradas, e cria saídas de valor para a organização. Em OGC (2009), processo é definido como um conjunto estruturado de atividades projetado para atingir um objetivo específico. Um processo transforma entradas em saídas definidas. Segundo a norma ABNT-NBR ISO 9000 (ABNT, 2005), um processo é composto por uma atividade ou conjunto de atividades que usa recursos para transformar insumos (entradas) em produtos (saídas). Por meio da figura 5 se procura ilustrar fluxos de atividades no contexto de um processo.

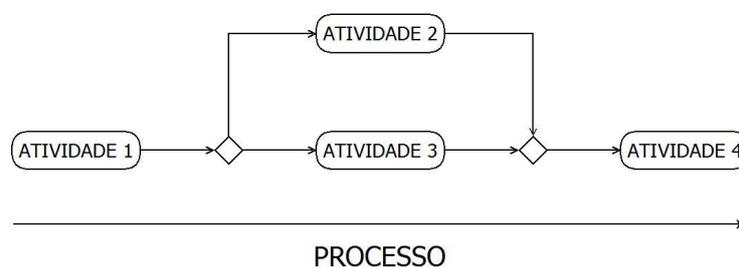


Figura 5 – Fluxo de atividades em um processo.

Em um projeto, a execução de um processo pode se concentrar em uma fase ou abranger várias fases (PMI, 2013). O PMBOK (PMI, 2013) destaca a necessidade de

processos apropriados para que os objetivos dos projetos sejam alcançados. A figura 6 ilustra que, no ciclo de vida de um projeto, atividades de diferentes processos podem ser realizadas, essa figura também ilustra que atividades de determinados processos podem se concentrar em determinadas fases.

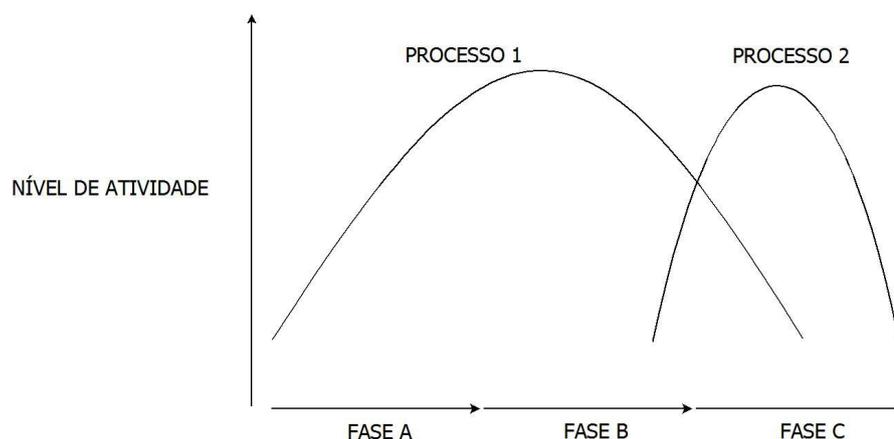


Figura 6 – Níveis de atividade de processos ao longo do ciclo de vida.

Em OGC (2009), a descrição de um processo inclui a razão para existência do processo, objetivos a serem alcançados pelo processo, contextualização do processo em relação a outros processos, atividades integrantes do processo. Aos processos podem ser associados roteiros (*procedures*), que são sequências de passos para executar os processos (PMI, 2013). Em projetos, existem processos de gerenciamento e processos orientados a produto, os processos de gerenciamento visam o fluxo efetivo do projeto durante o ciclo de vida, os processos orientados a produto visam especificar e construir produtos (PMI, 2013).

### 6.3.1 Processo de negócio

Para Berg et al. (2013), um processo de negócio é uma unidade de comportamento ou uma coleção de unidades de comportamento com relações de sequência ou dependência, com o objetivo de produzir uma coleção definida de produtos ou serviços. Ainda segundo Berg et al. (2013), um processo de negócio pode ser composto por subprocessos ou atividades, pode ser iniciado por um ou mais eventos de negócio, ou por outros processos. Um processo de negócio combina atividades, sendo cada atividade, parte de uma função de negócio. Liles e Presley (1996) sugerem as seguintes categorias de processos de negócio: processos que transformam restrições externas em restrições internas; processos que adquirem e

preparam recursos; processos que usam recursos para gerar os resultados (produtos ou serviços) da empresa.

## 6.4 Frameworks e métodos de gestão de projetos

A gestão de projeto pode ser definida como “planejamento, delegação, monitoração e controle de todos os aspectos do projeto, e a motivação dos envolvidos, de modo a atingir os objetivos do projeto dentro dos alvos de desempenho esperados para tempo, custo, qualidade, escopo, benefícios e riscos” (OGC, 2009). Existem *frameworks* e métodos para a gestão de projetos. A seguir, encontram-se descritos os *frameworks* denominados PMBOK Guide e PRINCE2.

### 6.4.1 PMBOK Guide

O PMBOK Guide é composto por descrições de princípios para o gerenciamento de projetos, definições de conceitos relacionados ao gerenciamento de projetos, descrições dos ciclos de vida de projeto e do gerenciamento de projetos, descrições de processos (PMI, 2013). Neste guia, os processos de gerenciamento de projetos são organizados nos seguintes grupos: iniciação, planejamento, execução, monitoração e controle, fechamento. O grupo de processos de iniciação é composto por processos executados com o intuito de definir novo projeto ou nova fase em projeto existente. O grupo de processos de planejamento é composto por processos que visam estabelecer o escopo, definir e refinar objetivos, estabelecer o curso de ação necessário a atingir os objetivos. O grupo de processos de execução é composto por processos executados para concluir o trabalho definido no plano de gerenciamento do projeto e satisfazer as especificações do projeto. O grupo de monitoração e controle é composto por processos necessários à monitoração, revisão e orquestração do progresso do projeto, identificação de áreas nas quais mudanças são necessárias aos planos; início de mudanças necessárias. Finalmente, o grupo de processos de fechamento é composto por processos necessários à conclusão de atividades em processos de outros grupos e à conclusão formal de projeto, fase ou obrigações contratuais (PMI, 2013).

No PMBOK Guide, os processos de gerenciamento são também agrupados em áreas de conhecimento. Essas áreas são as seguintes: Integração, Escopo, Tempo, Custos, Qualidade, Recursos Humanos, Comunicações, Riscos, Aquisições e *Stakeholders*. A área Integração inclui processos voltados à identificação, definição, combinação, unificação e coordenação de processos nos grupos de processos de gerenciamento de projetos. A área Escopo inclui processos que visam garantir que o projeto inclua o trabalho necessário à conclusão bem sucedida do projeto. A área Tempo inclui os processos necessários à conclusão do projeto em tempo hábil. A área Custos inclui os processos necessários à garantia da conclusão do projeto dentro do orçamento aprovado. A área Qualidade inclui

processos voltados a garantir que o projeto satisfaça as necessidades que levaram à sua realização. A área Recursos Humanos inclui processos voltados à organização, gestão e liderança da equipe. A área Comunicações inclui processos voltados à disponibilização de informações acerca do projeto. A área Riscos inclui processos voltados à identificação, análise, planejamento de resposta e controle de riscos no projeto. A área Aquisições inclui processos necessários à aquisição de produtos, serviços ou resultados necessários fora da equipe de projeto. A área *Stakeholders*, inclui processos para a identificação de pessoas, grupos e organizações que possam impactar o projeto ou serem impactadas pelo mesmo, analisar expectativas dos *stakeholders* e desenvolver estratégias para o engajamento deles no projeto (PMI, 2013).

#### 6.4.2 PRINCE2

PRINCE2 é um *framework*, desenvolvido pelo United Kingdom Office of Government Commerce (OGC). Esse *framework* visa auxiliar as organizações e os indivíduos no gerenciamento de projetos. Aborda o planejamento, delegação, monitoração e controle dos seguintes aspectos de desempenho em projetos: custo, tempo, qualidade, escopo, risco e benefícios (OGC, 2009). O PRINCE2 integra os seguintes elementos: princípios, temas e processos. Ao ser posto em aplicação, esse *framework* deve ser configurado ao contexto específico do projeto.

Os princípios caracterizam o emprego do *framework* e são os seguintes: existência de uma justificativa para iniciar o projeto que se mantém válida ao longo do projeto, é documentada em um caso de negócio e aprovada; equipes de projeto aprendem a partir de experiências anteriores, lições são buscadas e registradas ao longo da vida do projeto; papéis e responsabilidades encontram-se definidos no projeto; projetos são planejados, monitorados e controlados a cada etapa; a delegação de autoridade envolve a definição de tolerâncias para cada objetivo (tempo, custo, qualidade, escopo, risco, benefício); foco na definição e entrega de produtos; *framework* configurado de acordo com o projeto (OGC, 2009).

Os temas do *framework* PRINCE2 abordam os seguintes aspectos: Caso de Negócio, Organização, Qualidade, Planos, Risco, Mudança e Progresso. O tema “Caso de Negócio” aborda como desenvolver uma ideia de potencial valor em uma proposta e como manter o foco nos objetivos da organização. O tema “Organização” aborda a descrição de papéis e responsabilidades na equipe de gerenciamento de projeto. O tema “Qualidade” aborda o entendimento dos requisitos de qualidade dos produtos e o atendimento a esses requisitos. O tema “Planos” aborda passos e técnicas para o desenvolvimento de planos. O tema “Risco” aborda a gestão de incertezas. O tema “Mudança” descreve como avaliar e agir acerca de aspectos que impactem o projeto. O tema “Progresso” aborda a contínua viabilidade dos planos.

O PRINCE2 adota uma abordagem baseada em processos para a gestão de projetos. Existem sete processos compostos por atividades necessárias ao direcionamento, gerenciamento e entrega de projetos. Os processos são os seguintes: *Starting up a Project*, *Directing a Project*, *Initiating a Project*, *Managing a Stage Boundary*, *Closing a Project*, *Controlling a Stage*, *Managing Product Delivery*. As atividades integrantes dos processos podem ser executadas em série ou paralelo, e para cada atividade são recomendadas ações para se atingir os resultados.

## 7 Processos de desenvolvimento de *software*

Este capítulo apresenta conceitos acerca de comunidade virtual, comunidade virtual de prática e comunidade virtual de profissionais. Esses conceitos são importantes ao entendimento de comunidades voltadas ao desenvolvimento de *software* aberto. Neste capítulo são também apresentados conceitos acerca de *software*, ciclo de vida de *software* e processos de *software*. Finalmente, é descrito o processo tipicamente empregado no desenvolvimento de *software* aberto.

### 7.1 Desenvolvimento de *software*

Um produto de *software* é composto por programas de computador, procedimentos, documentação e dados pertinentes à operação de sistemas computacionais (IEEE, 1990). Produtos de *software* podem ser classificados como aplicativos, que suprem necessidades específicas de usuários; de suporte, que auxiliam no desenvolvimento e manutenção de outros produtos de *software*; ou de sistema, que facilitam a operação e manutenção de sistemas de computação e programas associados (IEEE, 1990).

Um produto de *software* não é fabricado no sentido clássico, é desenvolvido via um processo de engenharia (PRESSMAN, 2011). Na engenharia de *software*, os projetos são colaborativos, é necessário coordenar esforços de vários engenheiros ao longo de todo o ciclo de desenvolvimento de *software* (WHITEHEAD, 2007). Existe a necessidade de engenharia de *software* colaborativa, pois produtos de *software* frequentemente se tornam grandes e complexos, requerendo a participação de equipes de especialistas (MISTRÍK et al., 2010). Para desenvolver *software*, os membros das equipes precisam adotar processos apropriados, acompanhar o que os outros estão fazendo, compartilhar artefatos (produtos do trabalho) de baixo nível e conhecimentos de alto nível de modo controlado e consistente, serem proativamente informados sobre modificações efetuadas, e coordenar os trabalhos que realizam (MISTRÍK et al., 2010).

No desenvolvimento de *software*, parte do esforço consiste em compartilhar o entendimento acerca de variados artefatos, cada artefato incorporando um modelo próprio (WHITEHEAD, 2007). Entre engenheiros de *software*, embora ocorram comunicações não estruturadas em linguagem natural, a maior parte da colaboração é relativa a artefatos formais ou semiformais, é baseada em artefatos ou em modelos e visa produzir novos modelos, criar um entendimento compartilhado envolvendo os modelos, eliminar erros e ambiguidades nos modelos (WHITEHEAD, 2007). No desenvolvimento de *software*, informações variadas são necessárias. Muitas dessas informações podem ser classificadas como informações tecnológicas. Katia Montalli (JANNUZZI; MONTALLI, 1999) conceitua

informação tecnológica como “aquela que trata da informação necessária, utilizada e da informação gerada nos procedimentos de aquisição, inovação e transferência de tecnologia, nos procedimentos de metrologia, certificação da qualidade e normalização, e nos processos de produção”.

## 7.2 Processos de desenvolvimento de *software*

Processos de *software* envolvem aspectos organizacionais e tecnológicos, são intensos no uso de informações e na colaboração entre participantes (PRESSMAN, 2011; PERRY; STAUDENMEYER; VOTTA, 1994; SCHACH, 2008; SOMMERVILLE, 2010; BANI-SALAMEH; JEFFERY; AL-GHARAIBEH, 2010; SARMA, 2005). Os processos de desenvolvimento de *software* podem ser representados por modelos que os descrevem e que relacionam atividades, fluxos de atividades marcos e produtos. Dentre os modelos existentes, destaca-se o modelo iterativo e incremental. Nos processos que seguem esse modelo, o produto de *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 os resultados de duas iterações subsequentes é um incremento. São exemplos de processos de desenvolvimento que seguem o modelo iterativo e incremental: *IBM Rational Unified Process* e *Unified Software Development Process* (JACOBSON; BOOCH; RUMBAUGH, 1999). Nesses processos, cada ciclo de desenvolvimento é dividido em fases (concepção, elaboração, construção, e transição) e termina com o *software* pronto para produção. As fases são divididas em iterações, concluídas quando alcançados marcos definidos.

Alguns processos de desenvolvimento de *software* que são denominados ágeis. Esses processos apresentam as seguintes características: atividades não prescritas; ciclo de desenvolvimento decomposto em iterações com curta duração; iterações no planejamento do desenvolvimento do *software*; em cada iteração, encontram-se as atividades necessárias para atingir os objetivos da iteração; minimização da quantidade de atividades, para minimizar riscos e atingir objetivos; possibilidade de acréscimo ou descarte de atividades, à medida que riscos surjam ou sejam eliminados; *software* dividido em incrementos, posteriormente integrados; processo orientado às pessoas, com atribuição de poder aos desenvolvedores; ênfase na comunicação entre os participantes no processo de desenvolvimento (MILLER, 2001). Entre os processos denominados ágeis tem-se o *Extreme Programming* (XP), que sugere as seguintes práticas: cliente decide prazos e escopos das entregas a partir de estimativas providas pelos programadores; cliente participa da equipe de desenvolvimento; sistema posto em produção em curto espaço de tempo; novas entregas realizadas frequentemente; simplificação do desenho; testes de unidade realizados frequentemente; desenho evolui via transformações (*refactoring*); códigos são escritos por duplas; códigos são integrados em curtos intervalos de tempo; desenvolvedores trabalham em ambientes abertos;

qualquer programador pode melhorar qualquer código; redução da quantidade de horas extra (BECK, 1999).

## 7.3 Processos distribuídos de desenvolvimento de *software*

Mistrík et al. (2010) destacam que o desenvolvimento de *software* tem sofrido mudanças radicais, tais como: produtos de *software* aberto (*open source software*), terceirização, equipes distribuídas, processos globalizados de engenharia de *software*. Um processo de desenvolvimento de *software* pode ser distribuído em várias dimensões: geográfica, organizacional, temporal, entre grupos de stakeholders (GUMM, 2006). Por motivos variados, o desenvolvimento de *software* tem cruzado as fronteiras das organizações e dos países. São motivações para a distribuição do desenvolvimento de *software*: explorar diferentes mercados; reduzir custos de desenvolvimento; aproveitar oportunidades de aquisição de empresas; acessar pessoal técnico escasso; reduzir distâncias dos mercados para os quais os produtos de *software* se destinam; globalização da economia; competição por mercados; terceirização; avanços nas telecomunicações (AUDY; PRIKLADNICKI, 2008; HERBSLEB; MOITRA, 2001). No que tange à distribuição geográfica, os envolvidos no desenvolvimento de produtos de *software* podem apresentar diferentes níveis de dispersão. Por exemplo, podem se encontrar fisicamente em um mesmo local, em um mesmo país, em diferentes países ou em diferentes continentes. O desenvolvimento é geograficamente distribuído se pelo menos um ator se encontrar fisicamente distante dos demais (AUDY; PRIKLADNICKI, 2008).

O desenvolvimento de *software* pode ser realizado por equipes ou comunidades virtuais, com os seguintes potenciais benefícios: maior flexibilidade no balanceamento entre vida pessoal e profissional; redução da necessidade de espaço físico e de viagens; aceleração do aprendizado; oportunidade dos participantes ampliarem suas experiências trabalhando em equipes multiculturais (GILLAM; OPPENHEIM, 2006). Quanto às dificuldades, é possível destacar as seguintes: diferenças acerca de conceitos e terminologias empregadas em engenharia de *software* e gerência de projetos; diferenças no nível de entendimento acerca do domínio do problema; diferenças nos níveis de treinamento e conhecimento; falta do sentimento de propriedade em relação ao *software* desenvolvido (WONGTHONGTHAM et al., 2006).

### 7.3.1 Equipe virtual

Segundo Zenun, Loureiro e Araujo (2007), uma equipe é composta por poucas pessoas com habilidades complementares, comprometidas com propósito, metas e abordagem de trabalho em comum, pelos quais se mantêm responsáveis. Destaca que os membros de uma equipe podem compartilhar uma área física ou não. Quando a equipe é composta por

peças em diferentes localizações físicas é chamada de equipe virtual. [Lipnack e Stamps \(2000\)](#) destacam a diferença entre grupo e equipe. Segundo esses autores, um grupo é composto por indivíduos interagindo de modo interdependente com propósitos em comum ou motivações compartilhadas, enquanto equipes existem para um propósito orientado a tarefa. Uma equipe é composta por pessoas que, guiadas por um objetivo comum, interagem executando tarefas interdependentes. Quanto às equipes virtuais, [Lipnack e Stamps \(2000\)](#) destacam que os membros da equipe virtual podem cruzar fronteiras de tempo (pessoas que trabalham em diferentes horários) ou espaço (pessoas que trabalham em diferentes locais), usando tecnologias de comunicação e colaboração.

### 7.3.2 Comunidade virtual

Além de equipes virtuais, comunidades virtuais têm sido empregadas no desenvolvimento de produtos de *software*. Embora pessoas interajam usando redes de computadores há décadas, o conceito de comunidade virtual (*virtual community*) é recente. [Herring \(2002\)](#) destaca que, embora o termo “comunidade virtual” seja empregado para descrever grupos de pessoas que se comunicam primariamente, ou exclusivamente, através da Internet, há falta de consenso acerca desse termo. [Preece \(2001\)](#) sugere o termo “comunidade *online*” para designar qualquer espaço social virtual onde pessoas se encontram para obter ou prover informação, para prover suporte, para aprender, ou para encontrar companhia. Segundo [Preece \(2001\)](#), a comunidade pode ser local, nacional, internacional, pequena ou grande. [Rheingold \(2000\)](#) define comunidade virtual como um agregado social que emerge quando, utilizando uma rede de computadores, um grupo de pessoas discute, por tempo suficiente e com suficiente emoção, ao ponto de estabelecerem relações pessoais. [Ellis, Oldridge e Vasconcelos \(2004\)](#) destacam que as interações em comunidades virtuais são mediadas por computadores e se baseiam mais no compartilhamento de interesses do que em características sociais compartilhadas. Segundo [Bifulco e Santoro \(2005\)](#), as comunidades virtuais são estruturas sociais que emergem de uma sociedade tornada possível pela Internet. Essas comunidades agregam pessoas de interesses similares que visam se comunicar, compartilhar e trocar informações, se divertir ou preencher necessidades de participação social e empatia.

Nas comunidades virtuais, a comunicação envolve participantes que tipicamente estão geograficamente dispersos, apresentam variados históricos e objetivos. Os participantes podem nunca vir a se conhecer pessoalmente e as informações são compartilhadas via artefatos variados. Os membros de uma comunidade virtual disponibilizam informações para a comunidade tanto por altruísmo, sem esperar recompensa, quanto por esperarem, quando precisarem de informações, contar com o apoio da comunidade ([ELLIS; OLDRIDGE; VASCONCELOS, 2004](#)). Pesquisas demonstraram que os membros de comunidades virtuais não estão interessados apenas em participar com perguntas e respostas, apreciam também

o debate (SHIN; KIM, 2010).

Segundo Preece (2001), são características das comunidades virtuais: finalidade (foco compartilhado em um interesse, necessidade, informação, serviço ou suporte); pessoas que interagem na comunidade e têm necessidades individuais e sociais; políticas (linguagens e protocolos que guiam as interações das pessoas e contribuem para o desenvolvimento de um senso de história e normas socialmente aceitas). Preece (2001) sugere os seguintes determinantes de sucesso relacionados ao desenho das informações: velocidade de aprendizado, produtividade (quanto tempo leva para encontrar, ler e entender a informação), satisfação do usuário, retenção (quanto os usuários lembram acerca do desenho da informação na próxima visita) e erros (quantos erros os usuários cometem quando realizam uma tarefa que envolva encontrar ou usar informações). Por sua vez, relacionados à navegação, Preece (2001) sugere os seguintes determinantes de sucesso: facilidade com a qual o usuário se move e encontra o desejado, velocidade de aprendizado (quanto tempo leva para aprender a navegar) e produtividade (quanto tempo leva para chegar a um destino na comunidade ou encontrar a informação).

#### 7.3.2.1 Comunidade virtual de profissionais

Algumas comunidades virtuais são denominadas comunidades virtuais de profissionais (*professional virtual community*). Nessas comunidades, os participantes se comunicam e interagem buscando conhecimentos necessários à solução de problemas em áreas profissionais. Devido à importância dessas comunidades, seu surgimento e crescimento têm sido incentivados em várias organizações (CHEN, 2007). Segundo Bifulco e Santoro (2005), as comunidades virtuais de profissionais são arranjos organizacionais centrados no ser humano e voltados a alavancar o desenvolvimento de conhecimento e a criação de valor.

Comunidades virtuais de profissionais são comunidades de prática que buscam missões de negócio e adotam redes de computadores, práticas e ferramentas das comunidades virtuais. As comunidades de prática são comunidades onde os seus participantes compartilham interesses e aprofundam seus conhecimentos em uma área, interagindo com frequência. Segundo Wenger e Lave (1991), nessas comunidades ocorre aprendizado situado (*situated learning*) e participação periférica legítima (*legitimate peripheral participation*). Para Wenger e Lave (1991), a aquisição do conhecimento requer que os novatos movam em direção à participação integral nas práticas da comunidade. Por esse processo, os novatos aprendem gradualmente a realizar tarefas, migrando de uma participação periférica para uma situação de membros efetivos da comunidade. Com o tempo, o aprendizado coletivo resulta em práticas, que se tornam propriedade das comunidades.

Segundo Ankolekar (2004), as comunidades virtuais de profissionais podem ser orientadas a discussões, ao compartilhamento de conhecimento, ou à construção de artefatos. Para esses autores, comunidades de *software* aberto, que se formam em torno

do desenvolvimento de produtos de *software*, são comunidades virtuais de profissionais que apresentam características de comunidades orientadas a discussões e de comunidades orientadas à construção de artefatos. Como resultado das interações dos seus membros, as comunidades virtuais profissionais geram muitas informações, que geralmente não se encontram inter-relacionadas de acordo com os significados dos seus conteúdos, o que pode resultar na perda de produtividade.

## 7.4 Processo de desenvolvimento de *software* aberto

Os termos *software* livre (*free software*) e *software* aberto (*open source software*) estão frequentemente associados, entretanto, têm significados distintos. Scacchi et al. (2006) destacam diferenças entre esses dois termos. As diferenças encontram-se tanto nas licenças de tais produtos de *software*, quanto no modo como tais produtos de *software* são desenvolvidos, compartilhados, modificados, reusados e distribuídos. Os produtos de *software* livre geralmente adotam a GNU General Public License (GPL), enquanto produtos de *software* aberto podem adotar a licença GPL, ou não. *Software* livre pode ser considerado um movimento social, enquanto *software* aberto está associado a um processo de desenvolvimento. *Software* livre é *software* aberto, o inverso não necessariamente é verdade. Segundo Godfrey e TU (2000), uma característica de um *software* aberto é o fato do código fonte do *software* se encontrar disponível para qualquer um examiná-lo ou modificá-lo.

Para Crowston (2002), produtos de *software* aberto são desenvolvidos por organizações virtuais, aqueles que desenvolvem os produtos de *software* compartilham interesses e geralmente se encontram distribuídos geograficamente, sendo a Internet usada na coordenação das atividades realizadas. O desenvolvimento de *software* aberto se caracteriza por um processo com atividades de desenvolvimento visíveis; artefatos desenvolvidos disponíveis em sítios na Internet; e, em geral, sem gerência formal para o projeto, orçamento e cronograma. O processo de desenvolvimento de *software* aberto se caracteriza também pelo desenvolvimento de uma comunidade em paralelo ao desenvolvimento do produto de *software*. A construção de comunidades, a formação de alianças e a contribuição participativa, são importantes para que os projetos possam existir sem uma autoridade central (SCACCHI et al., 2006).

Pesquisas revelam que muitos projetos de *software* aberto se tornam interdependentes, formando ecossistemas de projetos. Essa interdependência pode envolver o compartilhamento de desenvolvedores, artefatos, ferramentas, e sítios. O desenvolvimento de um projeto de *software* aberto pode causar impacto sobre outros projetos a ele ligados. A existência de desenvolvedores que participam de múltiplos projetos resulta em alianças entre projetos. Pequenos projetos de *software* aberto podem se conectar formando redes

sociais grandes o suficiente ao ponto de contribuírem para o crescimento dos vários projetos interconectados (SCACCHI et al., 2006).

### 7.4.1 Participantes

Nos projetos de *software* aberto, os participantes podem ser organizados em três principais grupos. O primeiro é composto por usuários passivos, interessados apenas em usar o *software*. No segundo grupo, encontram-se participantes com variados perfis: usuários ativos que reportam defeitos e sugerem modificações, mas não leem códigos; participantes interessados no funcionamento do *software*, que leem códigos e atuam como revisores; desenvolvedores que se limitam a conhecer parte do *software* e a corrigir pequenos defeitos; desenvolvedores periféricos que contribuem esporadicamente com novos códigos; desenvolvedores ativos que regularmente contribuem com novos códigos e correções de defeitos. O terceiro grupo é o núcleo da comunidade, composto por desenvolvedores experientes. Esses desenvolvedores conhecem a arquitetura do *software*, desenvolvem a maioria dos códigos, revisam códigos de outros desenvolvedores e selecionam os códigos que farão parte do *software*.

Segundo Nakakoji et al. (2002), nem todas as comunidades têm participantes nos papéis anteriormente descritos. Além disso, há variações nos percentuais de participantes em cada papel. Quanto à estrutura da comunidade, esses autores destacam que, embora uma estrutura estritamente hierárquica não seja adotada nas comunidades de *software* aberto, as estruturas não são planas. A influência sobre o *software* e a comunidade depende do papel do participante. Nas comunidades voltadas ao desenvolvimento de *software* aberto, o papel de cada participante não é fixo e não depende de atributos como idade ou titulação; depende das contribuições do participante à comunidade. Quanto maior a quantidade de contribuições importantes, maior o reconhecimento da comunidade. Um participante pode ter se tornado membro do núcleo de um projeto após passar pelos papéis de usuário passivo, usuário ativo, desenvolvedor responsável por corrigir defeitos, desenvolvedor periférico e desenvolvedor ativo.

No núcleo das comunidades, além de desenvolvedores experientes, pode haver um líder de projeto responsável pela visão estratégica e por decidir quais contribuições serão aceitas. Em comunidades maiores, desenvolvedores ativos e de confiança do líder podem auxiliá-lo recomendando códigos a serem incorporados aos produtos de *software*. Nas comunidades em que não há um participante assumindo o papel de líder, as decisões são tomadas por grupos de desenvolvedores. Segundo Nakakoji et al. (2002), nos projetos que exploram os limites do desenvolvimento de *software*, o nível de qualidade requerido é elevado e o sucesso depende da visão e liderança do líder de projeto; nos projetos que visam preencher lacunas decorrentes da falta de produtos de *software* que supram certas necessidades, normalmente não há controle centralizado; nos projetos que visam desenvolver

produtos de *software* responsáveis por serviços estáveis e robustos, as modificações precisam ser cuidadosamente analisadas e o projeto é tipicamente controlado por um grupo de membros experientes.

Nas comunidades voltadas ao desenvolvimento de *software* aberto, a comunicação ocorre, principalmente, mediada por computadores ligados à Internet (ERENKRANTZ; TAYLOR, 2003; GERMAN, 2003; MOCKUS; FIELDING; HERBSLEB, 2002; MOODY, 2001; REIS; FORTES, 2002). Em tais projetos, portais são usados para facilitar o acesso dos desenvolvedores às fontes de informação. Esses portais são pontos de encontro para os membros das comunidades, em que são obtidas informações variadas: estado do projeto; relação de participantes; atividades por realizar e prioridades associadas às mesmas; necessidades futuras; detalhamento dos defeitos identificados; registros de comunicações trocadas entre participantes. Esses portais também proveem acesso aos códigos dos produtos de *software*.

Alguns autores consideram que certas comunidades virtuais voltadas ao desenvolvimento de *software* aberto se comportam como comunidades de prática (ANKOLEKAR; HERBSLEB; SYCARA, 2003). Nakakoji et al. (2002) destacam que as comunidades de projetos de *software* aberto encorajam e possibilitam mudanças de papéis, sendo tais comunidades consistentes com a teoria do aprendizado via participação periférica legítima descrita em Lave e Wenger (1991). Nesse processo, a presença de quem aprende é considerada legítima. Inicialmente, a participação é periférica, gradualmente expandindo-se em escopo. O aprendizado ocorre na comunidade em que o conhecimento é usado. Para participar, os novatos acessam as práticas da comunidade e realizam tarefas simples, em que os erros cometidos têm baixo custo. Nas referidas comunidades, o acesso a artefatos como os códigos dos produtos de *software* e a oportunidade de trabalhar com desenvolvedores experientes, possibilitam que os novatos conheçam práticas maduras de desenvolvimento de *software*. Para o sucesso de um projeto de *software* aberto, é necessário que evoluam *software* e comunidade. As modificações aos produtos de *software* não só alteram os produtos de *software*, mas também os papéis de quem as faz. Para o sucesso de um projeto, é importante que existam novos membros que desejem fazer parte do núcleo da comunidade e que contribuam continuamente. Nakakoji et al. (2002) destacam que, sendo a evolução do *software* e da comunidade interdependentes, é necessário, para o sucesso do projeto, atenção à criação e à manutenção da comunidade. Nos projetos de *software* aberto, é necessário criar, além de ambiente e cultura que incentivem o sentimento de participação, mecanismos que encorajem novos membros a gradualmente se moverem para o núcleo da comunidade.

## 7.4.2 Repositórios

Há várias comunidades voltadas ao desenvolvimento de *software* aberto que compartilham práticas e produtos de *software*. Em várias dessas comunidades, os códigos são armazenados em repositórios controlados por sistemas de controle de versão; os defeitos identificados, assim como novos requisitos, são registrados via produtos de *software* de suporte acessados através da Internet; as atividades de colaboração e coordenação, entre os participantes, são realizadas via canais de comunicação como correio eletrônico, listas de discussão, fóruns e chats. [Ankolekar \(2004\)](#) destaca que nas comunidades virtuais voltadas ao desenvolvimento de *software* aberto, para um desenvolvedor participar de modo efetivo, precisa entender quem está trabalhando no que, e como os outros membros da comunidade são afetados pelo seu trabalho. Para isso, os desenvolvedores empregam fontes de informação e canais de comunicação como listas de mensagens de correio; registros do serviço de *chat*; e registros de alterações nos repositórios. Esses registros são mantidos no sistema de controle de versões do projeto e tipicamente contêm informações acerca do autor da modificação, os arquivos afetados, o número de modificações e as diferenças entre a versão atual e as antigas versões do código ([GUTWIN; PENNER; SCHNEIDER, 2004](#)).

Nos repositórios também pode ser encontrada documentação do produto; ajuda a usuários finais; documentação de solicitações de alterações; documentação de defeitos; suporte à colaboração entre desenvolvedores. Nos repositórios são também armazenados artefatos em formatos variados. Mecanismos ineficazes para compartilhamento e gestão de informações, podem ser barreiras ao sucesso de comunidades de desenvolvimento de *software* geograficamente distribuídas ([HERBSLEB; MOITRA, 2001](#)). [Wu et al. \(2007\)](#) destacam que, embora os repositórios devam prover mecanismos para organização e recuperação de artefatos, em vários repositórios os mecanismos para pesquisa e recuperação são inadequados.

Segundo [Simmons e Dillon \(2006\)](#), os repositórios de comunidades de desenvolvimento de *software* aberto são vitais para essas comunidades, pois registram os conhecimentos por elas acumulados. Esses autores destacam que, em vários desses repositórios, existe pouco suporte à estruturação das informações para torná-las significativas a diferentes usuários. Nesses repositórios, parte significativa das informações é não estruturada, o que gera dificuldades aos desenvolvedores e inibe a participação de novos desenvolvedores. Para [Simmons e Dillon \(2006\)](#), é desejável um entendimento de como organizar as informações nos repositórios das comunidades virtuais voltadas ao desenvolvimento de *software* aberto. Isso reduziria dissonâncias conceituais e facilitaria o acesso às informações. [Ankolekar, Herbsleb e Sycara \(2003\)](#) mencionam que um dos desafios ao desenvolvimento de *software* aberto é a necessidade de se chegar com a informação certa à pessoa certa para a tarefa, e apresentá-la de forma compreensível e utilizável. Para atingir esse objetivo, destacam a importância do processamento de informações não estruturadas, ou semiestruturadas, a

partir de representações dos conteúdos dessas informações. Esses autores também destacam a importância da organização de informações originadas em diferentes fontes para apresentação aos usuários.

Parte III

Resultados

## 8 Proposta de modelo de *framework*

Neste capítulo, é proposto um modelo de *framework* de arquitetura da informação no contexto organizacional, baseado em roteiros. Esse modelo de *framework*, composto por modelo de ciclo de vida de projeto, processos, artefatos e roteiros, foi desenvolvido a partir de informações em TOGAF (The Open Group, 2011), PMBOK (PMI, 2013), MAIA (COSTA, 2009) e outras fontes. As atividades durante o desenvolvimento foram as seguintes:

- Selecionar *frameworks* e métodos a serem configurados;
- Definir domínio de arquitetura (*architecture domain*);
- Definir pontos de vista (*viewpoint*) de arquitetura;
- Definir modelo de ciclo de vida do projeto;
- Definir fases do modelo de ciclo de vida do projeto;
- Definir processos de gerenciamento;
- Definir processos de arquitetura;
- Definir artefatos de gerenciamento;
- Definir artefatos de arquitetura;
- Descrever roteiros.

### 8.1 Seleção de *frameworks* e métodos

O modelo de *framework* proposto nesta dissertação foi desenvolvido a partir de informações em TOGAF, PMBOK, MAIA e outras fontes. O TOGAF foi escolhido pelos seguintes motivos: *framework* adotado globalmente por várias organizações; padrão do *The Open Group*, um consórcio de várias empresas de TI; facilidade de acesso à documentação; *framework* genérico e configurável. O PMBOK foi escolhido por ser um guia globalmente reconhecido de melhores práticas para gerenciamento de projetos. Por sua vez, o MAIA foi escolhido pelos seguintes motivos: método baseado em pressupostos científicos, método adaptável, possibilidade de aplicação do método no contexto organizacional, método que enfatiza a representação de espaços de informação por meio de taxonomias e de ontologias.

A configuração do TOGAF para o seu emprego em contexto específico é prevista em *The Open Group (2011)*, onde é destacado que, embora o *framework* TOGAF possa

ser empregado com as entregas nele descritas, essas entregas podem ser substituídas ou estendidas por entregas definidas em outros *frameworks*. Um arquiteto pode configurar o TOGAF para integrá-lo a processos e estruturas de uma organização. Para configurar o TOGAF para alinhá-lo a outros processos, *The Open Group (2011)* recomenda as seguintes atividades:

- Produzir terminologia para descrever o conteúdo da arquitetura;
- Incluir e remover tarefas;
- Configurar conteúdos.

Quanto à configuração de processos de gerenciamento de projetos, essa possibilidade é prevista no PMBOK. Esse guia destaca que a existência de boas práticas para gerenciamento de projetos não significa que elas devam ser aplicadas uniformemente em todos os projetos. Para cada projeto, o referido guia sugere que sejam definidas as práticas apropriadas.

Finalmente, quanto à configuração do MAIA, em *Costa (2009)* é destacada a adaptabilidade do método, sendo afirmado que o MAIA pode ser aplicado com o apoio de processos consagrados e que o emprego de técnicas, nos mais diversos níveis, pode ser exercitado.

## 8.2 Definição de domínio e de pontos de vista de arquitetura

Em TOGAF é sugerida a existência dos seguintes domínios de arquitetura: Negócios, Dados, Aplicações e Tecnologia. O modelo de *framework* proposto nesta dissertação visa descrever a arquitetura no domínio Negócios, domínio que foca em como a organização opera para atingir os seus objetivos de negócio. A escolha desse domínio decorre da aquisição de conhecimentos acerca desse domínio de arquitetura ser considerado um pré-requisito para a aquisição de conhecimentos acerca de outros domínios de arquitetura (*The Open Group, 2011*).

Nesta dissertação, foram selecionados os seguintes pontos de vista para descrição da arquitetura da informação: organizacional (*organization viewpoint*), cooperação entre atores (*actor cooperation viewpoint*), função de negócio (*business function viewpoint*), realização de serviço (*service realization viewpoint*), processo de negócio (*business process viewpoint*). Segundo *Lankhorst (2012)*, o ponto de vista organizacional apresenta a estrutura organizacional da empresa, departamento ou outra entidade organizacional. O ponto de vista de cooperação de atores foca nos relacionamentos entre atores e seus ambientes. O ponto de vista função de negócio apresenta as principais funções de negócio e os seus relacionamentos em termos de fluxos de informação, valores e bens. O ponto de vista

realização de serviço apresenta como os serviços de negócio são realizados por processos. Finalmente, o ponto de vista de processo apresenta as estruturas dos processos de negócio.

### 8.3 Projeto de descrição de arquitetura

Nesta dissertação, a descrição de uma arquitetura, como sugerido em ISO (2011), resulta de trabalho que visa expressar a arquitetura; uma arquitetura é descrita por artefatos (produtos do trabalho) para possibilitar o entendimento da arquitetura pelos *stakeholders*; uma arquitetura é descrita por artefatos resultantes da execução de processos em um projeto. A execução de processos de arquitetura em projetos encontra respaldo nas seguintes recomendações em TOGAF: ciclos do método TOGAF ADM devem ser executados no contexto de *frameworks* de gerenciamento de projetos; atividades de arquitetura devem ser planejadas e gerenciadas seguindo-se práticas aceitas nas organizações. Em *The Open Group (2011)* também é destacado que atividades de arquitetura podem ocorrer em “projetos de desenvolvimento de arquiteturas”. A figura 7 ilustra relações propostas entre projeto de descrição de arquitetura, descrição de arquitetura, artefato, arquitetura, sistema, ambiente e *stakeholder*. Parte dos relacionamentos nessa figura se origina em ISO (2011).

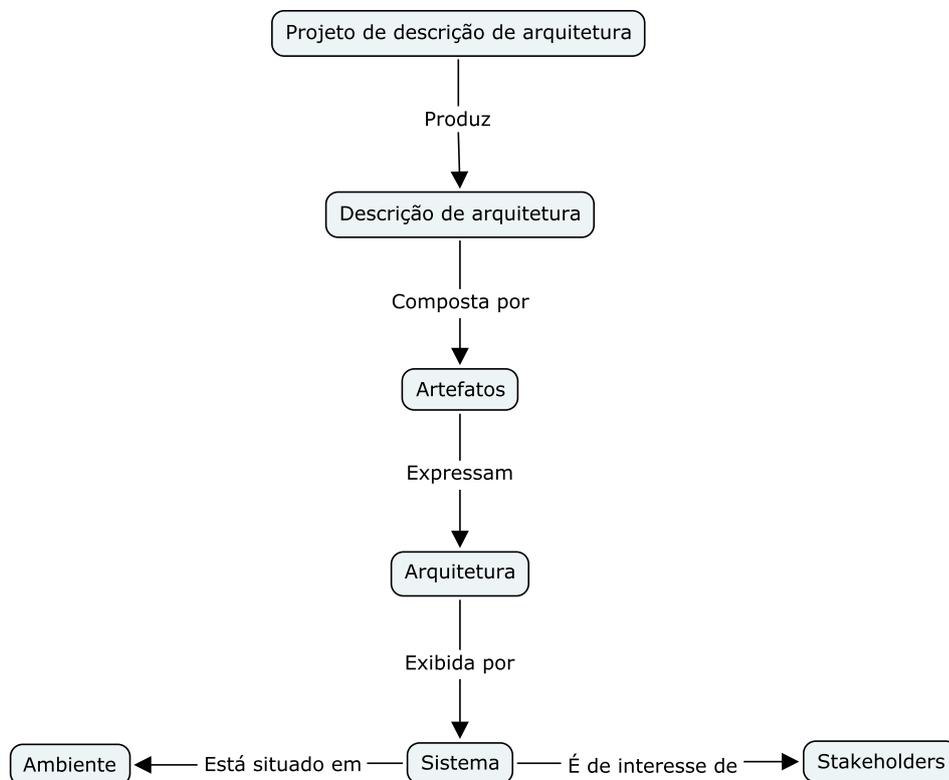


Figura 7 – Contexto de projeto de descrição de arquitetura.

## 8.4 Modelo de ciclo de vida

No modelo de *framework* proposto nesta dissertação, é adotado o modelo de ciclo de vida iterativo e incremental. Nesse modelo de ciclo de vida, ao longo do ciclo de vida, diversas iterações podem ocorrer e resultam em incrementos ao projeto. A escolha do modelo de ciclo de vida iterativo e incremental deve-se aos seguintes motivos: *The Open Group (2011)* sugere empregar iterações no desenvolvimento e gestão de arquiteturas; *The Open Group (2011)* sugere que o TOGAF ADM seja empregado em processos iterativos; o modelo iterativo contribui na gestão de mudanças, na redução da complexidade e na redução dos riscos no projeto (PMI, 2013); esse modelo facilita a incorporação de *feedbacks* e lições aprendidas (PMI, 2013); o modelo de ciclo de vida iterativo e incremental é compatível com princípios propostos em MAIA. Na figura 8 são propostos relacionamentos entre os conceitos que serão a seguir descritos nesta dissertação.

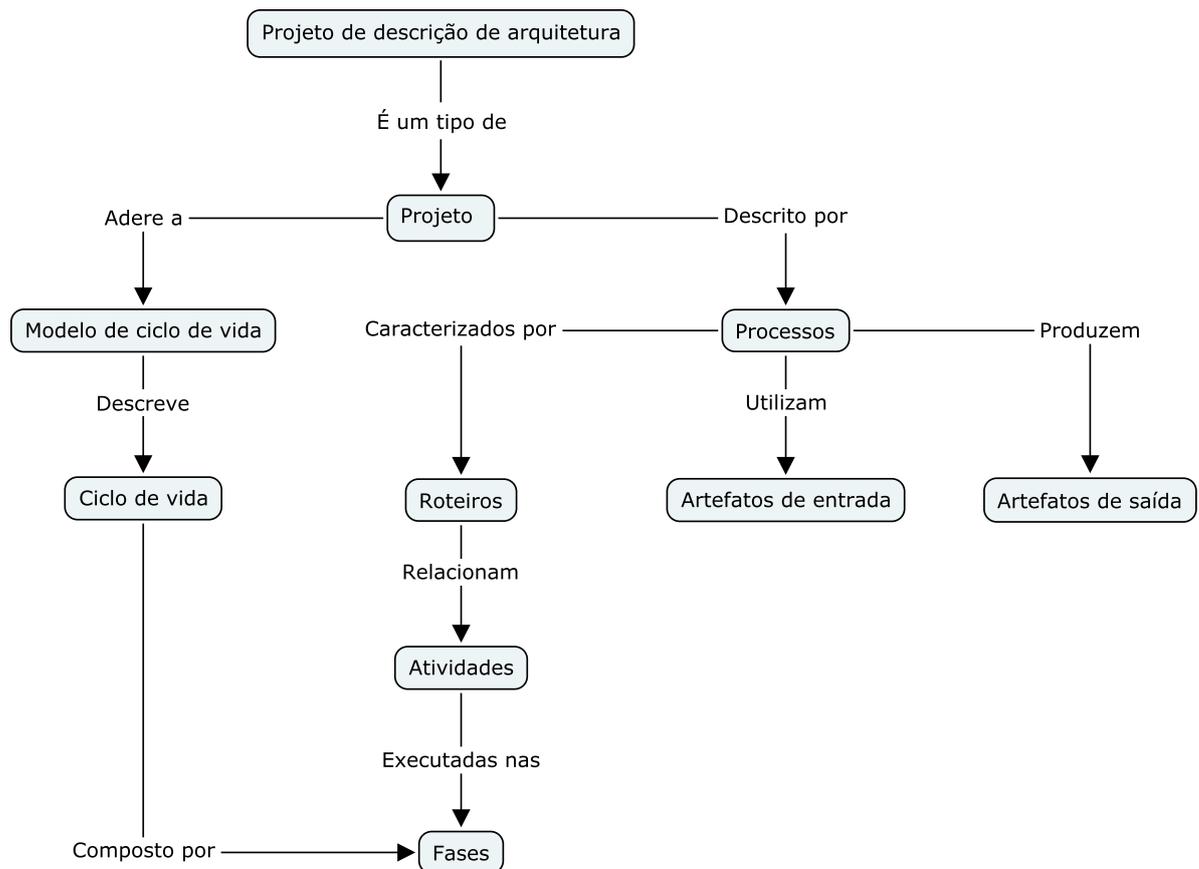


Figura 8 – Relacionamentos entre conceitos em projeto de descrição de arquitetura.

### 8.4.1 Fases no ciclo de vida

A definição das fases no ciclo de vida do modelo de *framework* proposto nesta dissertação visou contemplar os seguintes requisitos em PMBOK: cada fase ter foco que a distinga; a realização dos objetivos de cada fase requerer processos únicos; cada fase encerrar com a entrega de produtos do trabalho.

A relação de fases no ciclo de vida para o modelo de *framework* proposto nesta dissertação resultou de análise de fases em PMBOK, TOGAF e MAIA. No modelo de *framework* proposto nesta dissertação, é adotado o modelo de ciclo de vida genérico descrito em PMBOK, composto pelas seguintes fases:

- Iniciação;
- Organização e preparação;
- Execução do trabalho;
- Encerramento.

## 8.5 Processos

No modelo de *framework* proposto nesta dissertação, seguindo a abordagem em PMBOK, processos são organizados em grupos: grupo de processos de gerenciamento e grupo de processos orientados a produto.

O grupo de processos de gerenciamento proposto resulta da seleção de processos descritos em PMBOK e levou em consideração as restrições de tempo impostas ao projeto e que, nesta pesquisa, a equipe responsável pelo projeto é composta por um membro. Na tabela 1 são relacionados os processos de gerenciamento do modelo de *framework* proposto nesta dissertação.

Por sua vez, o grupo de processos orientados a produto proposto é composto por processos de arquitetura da informação e resulta da análise e integração de passos em TOGAF e MAIA. Os passos analisados fazem parte de fases consideradas importantes à descrição de um espaço de informação existente (*baseline architecture*) no domínio de arquitetura selecionado. A partir da identificação e análise desses passos, foram selecionados os processos do modelo de *framework* proposto nesta dissertação. A relação dos processos de arquitetura do modelo de *framework* proposto nesta dissertação e a relação de passos analisados, encontram-se nas tabelas 2 a 4.

Tabela 1 – Fases e processos de gerenciamento do modelo de *framework* proposto.

| FASE                     | PROCESSO  |
|--------------------------|---|
| Iniciação                | Desenvolver termo de abertura do projeto<br>Identificar partes interessadas   |
| Organização e preparação | Desenvolver plano de gerenciamento do projeto<br>Coletar requisitos<br>Definir escopo<br>Definir atividades<br>Sequenciar atividades<br>Estimar durações das atividades<br>Desenvolver cronograma |
| Execução do trabalho     | Controlar cronograma  |
| Encerramento             | Encerrar projeto ou fase  |

Tabela 2 – Fases e processos de arquitetura do modelo de *framework* proposto.

| FASE                     | PROCESSO  |
|--------------------------|---|
| Iniciação                |   |
| Organização e preparação | Confirmar <i>frameworks</i> de suporte e governança<br>Identificar e estabelecer princípios de arquitetura<br>Configurar o TOGAF e outros <i>frameworks</i> de arquitetura selecionados<br>Implementar ferramentas de arquitetura |
| Execução do trabalho     | Selecionar modelos, pontos de vista e ferramentas<br>Desenvolver descrição da arquitetura atual ( <i>baseline</i> )   |
| Encerramento             |   |

## 8.6 Artefatos

A partir da análise dos processos e artefatos em PMBOK, foram definidos os artefatos de gerenciamento resultantes dos processos de gerenciamento do modelo de *framework* proposto nesta dissertação. Na tabela 5 são relacionados os artefatos de gerenciamento do modelo de *framework* proposto nesta dissertação.

A identificação dos artefatos de arquitetura do modelo de *framework* proposto nesta dissertação foi realizada a partir da identificação e da análise de artefatos em TOGAF e MAIA. Os artefatos analisados fazem parte de fases consideradas importantes à descrição de um espaço de informação existente (*baseline architecture*) no domínio de arquitetura contemplado nesta dissertação. A relação dos artefatos de arquitetura do modelo de *framework* proposto nesta dissertação e a relação dos artefatos analisados, encontram-se nas tabelas 6 a 8.

Tabela 3 – Fases e passos analisados em MAIA.

| FASE    | PASSO   |
|---------|---|
| Escutar | Capturar registros pertencentes ao espaço de informação inicial<br>Armazenar registros pertencentes ao espaço de informação inicial<br>Definir taxonomia dos registros capturados e armazenados<br>Catalogar registros em conjuntos específicos considerando a tipologia de cada registro<br>Reunir registros em conjuntos específicos considerando a tipologia de cada registro<br>Compartilhar resultados das atividades entre integrantes do espaço de informação<br>Codificar ontologias que representem o espaço de informação inicial a partir dos registros<br>Gerar representação esquemática do espaço de informação inicial |
| Pensar  | Usar mapa de conceitos preliminar para interpretar o espaço de informação inicial<br>Definir critérios de forma, utilidade e estética<br>Codificar modelo de representação, função e estética do espaço de informação inicial   |

Tabela 4 – Fases e passos analisados em TOGAF.

| FASE                   | PASSO  |
|------------------------|--|
| Preliminar             | Delimitar organizações impactadas<br>Confirmar <i>frameworks</i> de suporte e governança<br>Definir e estabelecer organização e equipe de arquitetura<br>Identificar e estabelecer princípios de arquitetura<br>Configurar o TOGAF e outros <i>frameworks</i> de arquitetura selecionados<br>Implementar ferramentas de arquitetura  |
| Visão de Arquitetura   | Estabelecer projeto de arquitetura<br>Identificar <i>stakeholders</i> , interesses ( <i>concerns</i> ) e requisitos de negócio<br>Confirmar e elaborar metas, motores ( <i>drivers</i> ) e restrições de negócio<br>Avaliar capacidades de negócio<br>Avaliar prontidão para transformação do negócio<br>Definir escopo<br>Confirmar e elaborar princípios de arquitetura, incluindo princípios de negócio<br>Desenvolver visão de arquitetura<br>Definir proposições de valor da arquitetura alvo e KPIs<br>Identificar riscos de transformação do negócio e atividades de mitigação<br>Desenvolver enunciado de trabalho de arquitetura e garantir aprovação |
| Arquitetura de Negócio | Selecionar modelos de referência, pontos de vista e ferramentas<br>Desenvolver descrição da arquitetura atual ( <i>baseline</i> )<br>Desenvolver descrição de arquitetura alvo<br>Realizar análise de intervalo ( <i>gap analysis</i> )<br>Definir componentes candidatos para o trajeto ( <i>roadmap</i> )<br>Solucionar impactos ao longo do cenário ( <i>landscape</i> ) de arquitetura<br>Conduzir revisão formal pelas partes interessadas ( <i>stakeholders</i> )<br>Finalizar arquitetura de negócio<br>Criar documento de definição de arquitetura   |

Tabela 5 – Fases e artefatos de gerenciamento do modelo de *framework* proposto.

| FASE                     | ARTEFATO   |
|--------------------------|--|
| Iniciação                | Termo de abertura do projeto<br>Registro das partes interessadas   |
| Organização e preparação | Plano de gerenciamento do projeto<br>Documentação dos requisitos<br>Enunciado do escopo do projeto<br>Lista de atividades<br>Lista de marcos<br>Diagrama de sequência de atividades<br>Estimativas das durações das atividades<br>Cronograma |
| Execução do trabalho     | Estimativas de cronograma  |
| Encerramento             | Produto final ou resultados da fase  |

Tabela 6 – Fases e artefatos de arquitetura do modelo de *framework* proposto.

| FASE                     | ARTEFATO   |
|--------------------------|--|
| Iniciação                |  |
| Organização e preparação | Catálogo de princípios de arquitetura<br><i>Framework</i> de arquitetura configurado   |
| Execução do trabalho     | Catálogo de papéis<br>Catálogo de serviço/função de negócio<br>Catálogo de localização<br>Catálogo processo/evento/controle/produto<br>Catálogo de casos de uso de negócio<br>Catálogo organização/ator<br>Diagrama de casos de uso de negócio<br>Diagrama de fluxo de processo<br>Diagrama de eventos<br>Diagrama de decomposição funcional<br>Diagrama serviço/informação de negócio<br>Diagrama business <i>footprint</i><br>Diagrama de ciclo de vida do produto<br>Diagrama de decomposição da organização<br>Matriz ator/papel<br>Taxonomia de termos e conceitos<br>Mapa de conceitos<br>Ontologia de conceitos |
| Encerramento             | Produto final ou resultado da fase   |

Tabela 7 – Fases e artefatos analisados em MAIA.

| FASE    | ARTEFATO   |
|---------|--|
| Escutar | Taxonomia de registros coletados e armazenados<br>Mapa de conceitos preliminar representando o espaço de informação inicial<br>Ontologias representando o espaço de informação inicial |
| Pensar  | Definições de critérios de forma, utilidade e estética<br>Modelo de representação, função e estética do espaço de informação inicial   |

Tabela 8 – Fases e artefatos analisados em TOGAF.

| FASE                   | ARTEFATO  |
|------------------------|---|
| Preliminar             | Modelo organizacional para arquitetura<br>Framework de arquitetura configurado<br>Repositório de arquitetura Inicial<br>Reafirmação de princípios, metas ( <i>goals</i> ) e motores ( <i>drivers</i> ) do negócio<br>Solicitação de trabalho de arquitetura<br>Framework de governança de arquitetura<br>Catálogo de princípios   |
| Visão de arquitetura   | Enunciado do trabalho de arquitetura<br>Enunciados refinados de princípios, metas e motores do negócio<br>Princípios de arquitetura<br>Avaliação do nível de capacidade<br>Framework de arquitetura configurado<br>Visão de arquitetura<br>Esboço do documento de definição de arquitetura<br>Plano de comunicação<br>Repositório de arquitetura povoado com conteúdo adicional<br>Matriz de mapeando das partes interessadas<br>Diagrama da cadeia de valor<br>Diagrama de conceito da solução   |
| Arquitetura de negócio | Versões atualizadas e refinadas das entregas da fase Visão de Arquitetura<br>Esboço do documento de definição da arquitetura<br>Esboço do documento de especificação de requisitos de arquitetura<br>Componentes da arquitetura de negócio de um roteiro de arquitetura<br>Catálogo organização/ator<br>Catálogo motor/meta/objetivos<br>Catálogo de papéis<br>Catálogo de serviço/função de negócio<br>Catálogo de localização<br>Catálogo processo/evento/controle/produto<br>Catálogo contrato/medida<br>Matriz de interação de negócio<br>Matriz ator/papel<br>Diagrama <i>business footprint</i><br>Diagrama serviço/informação de negócio<br>Diagrama de decomposição funcional<br>Diagrama de ciclo de vida do produto<br>Diagrama meta/objetivo/serviço<br>Diagrama de casos de uso de negócio<br>Diagrama de decomposição da organização<br>Diagrama de fluxo de processo<br>Diagrama de eventos |

## 8.7 Roteiros

Roteiros para os processos do modelo de *framework* proposto nesta dissertação foram desenvolvidos a partir de informações em TOGAF, PMBOK, MAIA e outras fontes. Esses roteiros relacionam passos para executar os processos. Cada roteiro contém as seguintes informações:

- Nome do roteiro;
- Nome do processo ao qual o roteiro encontra-se associado;
- Relação de atividades;
- Nomes dos artefatos produzidos;
- Referências consultadas ou complementares ao roteiro.

Os nomes dos roteiros construídos são os seguintes: Desenvolver termo de abertura do projeto; Desenvolver plano de gerenciamento do projeto; Desenvolver documentação dos requisitos; Desenvolver enunciado do escopo do projeto; Desenvolver lista de marcos; Desenvolver lista de atividades; Desenvolver diagrama de sequência de atividades; Desenvolver cronograma; Desenvolver registro das partes interessadas; Descrever princípios de arquitetura; Descrever configuração do *framework* de arquitetura; Descrever atores e papéis; Descrever funções de negócio; Descrever localizações; Desenvolver modelo de casos de uso de negócio; Descrever *business footprint*; Descrever ciclo de vida do produto; Descrever composição da organização; Descrever processos do negócio; Desenvolver taxonomia de termos e conceitos; Desenvolver mapa de conceitos; Desenvolver ontologia de conceitos. Nesta dissertação, os roteiros anteriormente relacionados encontram-se descritos nos apêndices A e B desta dissertação. Na figura 9, a título de exemplo, encontra-se um desses roteiros.

|           |  |
|-----------|--|
| ROTEIRO   | Desenvolver taxonomia de termos e conceitos  |
| PROCESSO  | Desenvolver descrição da arquitetura atual ( <i>baseline</i> )   |
| DESCRIÇÃO | <p>Definir título do documento.</p> <p>Definir número de versão do documento.</p> <p>Definir data de elaboração do documento.</p> <p>Descrever objetivos do documento.</p> <p>Manter atualizado o histórico de alterações do documento.</p> <p>Definir objetivos (indexação, suporte à recuperação, organização, etc.) da taxonomia.</p> <p>Definir usuários da taxonomia.</p> <p>Definir conteúdo abordado, área de conhecimento e escopo da taxonomia.</p> <p>Definir tipos de relacionamentos permitidos (hierárquico, poli-hierárquico, equivalência, etc.).</p> <p>Definir como representar os relacionamentos permitidos.</p> <p>Definir como representar os níveis hierárquicos.</p> <p>Definir características estruturais (número máximo de níveis hierárquicos, etc.).</p> <p>Definir formas dos termos (número de palavras, forma gramatical, etc.).</p> <p>Definir atributos dos termos.</p> <p>Definir ferramentas para criar, administrar e usar a taxonomia.</p> <p>Definir fontes dos termos e conceitos.</p> <p>Coletar termos por meio de auditorias de conteúdos, entrevistas, etc.</p> <p>Agrupar termos em categorias.</p> <p>Identificar categorias de alto nível.</p> <p>Listar categorias de alto nível empregando ferramenta como planilha.</p> <p>Identificar categorias em níveis inferiores.</p> <p>Listar categorias em níveis inferiores empregando ferramenta como planilha.</p> <p>Representar relacionamentos hierárquicos entre termos.</p> <p>Identificar termos preferidos.</p> <p>Identificar termos não preferidos.</p> <p>Representar relacionamentos entre termos preferidos e não preferidos.</p> <p>Realizar testes com usuários.</p> <p>Revisar taxonomia.</p> <p>Selecionar normas e padrões às quais a taxonomia desenvolvida deve aderir.</p> <p>Adequar a taxonomia a normas e padrões selecionados.</p> <p>Referenciar fontes de informação consultadas.</p> |
| ARTEFATOS | Taxonomia  |
| FONTES    | <p>ANSI/NISO Z39.19-2005 (NISO, 2005).</p> <p>Hedden (2010).</p> <p>MAIA (COSTA, 2009).</p>  |

Figura 9 – Exemplo de roteiro desenvolvido.

## 8.8 Fontes de informação

O modelo de *framework* proposto nesta dissertação foi desenvolvido a partir de informações em TOGAF, PMBOK, MAIA e outras fontes. Nas tabelas 9 a 13 são identificadas as principais fontes de informação acessadas durante o desenvolvimento desse modelo.

Tabela 9 – Fontes de componentes do modelo de *framework* proposto.

| COMPONENTE                               | FONTE |       |      |
|--|-------|-------|------|
|  | PMBOK | TOGAF | MAIA |
| Modelo de ciclo de vida                  | X     | X     | X    |
| Fases no ciclo de vida                   | X     |       |      |
| Roteiros para processos de gerenciamento | X     |       |      |
| Roteiros para processos de arquitetura   |       | X     | X    |

Tabela 10 – Fontes dos processos de gerenciamento do modelo de *framework* proposto.

| PROCESSO DE GERENCIAMENTO                     | FONTE |       |      |
|---|-------|-------|------|
|   | PMBOK | TOGAF | MAIA |
| Desenvolver termo de abertura do projeto      | X     |       |      |
| Identificar partes interessadas               | X     |       |      |
| Desenvolver plano de gerenciamento do projeto | X     |       |      |
| Coletar requisitos                            | X     |       |      |
| Definir escopo                                | X     |       |      |
| Definir atividades                            | X     |       |      |
| Sequenciar atividades                         | X     |       |      |
| Estimar durações das atividades               | X     |       |      |
| Desenvolver cronograma                        | X     |       |      |
| Controlar cronograma                          | X     |       |      |
| Encerrar projeto ou fase                      | X     |       |      |

Tabela 11 – Fontes dos processos de arquitetura do modelo de *framework* proposto.

| PROCESSO DE ARQUITETURA   | FONTE |       |      |
|---|-------|-------|------|
|   | PMBOK | TOGAF | MAIA |
| Confirmar <i>frameworks</i> de suporte e governança                       |       | X     |      |
| Identificar e estabelecer princípios de arquitetura                       |       | X     | X    |
| Configurar o TOGAF e outros <i>frameworks</i> de arquitetura selecionados |       | X     |      |
| Implementar ferramentas de arquitetura                                    |       | X     |      |
| Selecionar modelos, pontos de vista e ferramentas                         |       | X     |      |
| Desenvolver descrição da arquitetura atual ( <i>baseline</i> )            |       | X     | X    |

Tabela 12 – Fontes dos artefatos de gerenciamento do modelo de *framework* proposto.

| ARTEFATO DE GERENCIAMENTO               | FONTE |       |      |
|---|-------|-------|------|
|   | PMBOK | TOGAF | MAIA |
| Termo de abertura do projeto            | X     |       |      |
| Registro das partes interessadas        | X     |       |      |
| Plano de gerenciamento do projeto       | X     |       |      |
| Documentação dos requisitos             | X     |       |      |
| Enunciado do escopo do projeto          | X     |       |      |
| Lista de atividades                     | X     |       |      |
| Lista de marcos                         | X     |       |      |
| Diagrama de sequência de atividades     | X     |       |      |
| Estimativas das durações das atividades | X     |       |      |
| Cronograma                              | X     |       |      |
| Estimativas de cronograma               | X     |       |      |

Tabela 13 – Fontes dos artefatos de arquitetura do modelo de *framework* proposto.

| ARTEFATO DE ARQUITETURA                     | FONTE |       |      |
|---|-------|-------|------|
|   | PMBOK | TOGAF | MAIA |
| Catálogo de princípios de arquitetura       |       | X     |      |
| <i>Framework</i> de arquitetura configurado |       | X     |      |
| Catálogo de papéis                          |       | X     |      |
| Catálogo de serviço/função de negócio       |       | X     |      |
| Catálogo de localização                     |       | X     |      |
| Catálogo processo/evento/controle/produto   |       | X     |      |
| Catálogo de casos de uso de negócio         |       | X     |      |
| Catálogo organização/ator                   |       | X     |      |
| Diagrama de casos de uso de negócio         |       | X     |      |
| Diagrama de fluxo de processo               |       | X     |      |
| Diagrama de eventos                         |       | X     |      |
| Diagrama de decomposição funcional          |       | X     |      |
| Diagrama serviço/informação de negócio      |       | X     |      |
| Diagrama business <i>footprint</i>          |       | X     |      |
| Diagrama de ciclo de vida do produto        |       | X     |      |
| Diagrama de decomposição da organização     |       | X     |      |
| Matriz ator/papel                           |       | X     |      |
| Taxonomia de termos e conceitos             |       |       | X    |
| Mapa de conceitos                           |       |       | X    |
| Ontologia de conceitos                      |       |       | X    |

No desenvolvimento dos roteiros, além de TOGAF, PMBOK e MAIA, as seguintes fontes de informação se destacaram: [Heldman \(2011\)](#), [Robertson e Robertson \(2006\)](#), [Jacka e Keller \(2009\)](#), [Kulak e Guiney \(2003\)](#), [Langlands e Edwards \(2009\)](#), [NISO \(2005\)](#), [Hedden \(2010\)](#), [Noy e McGuinness \(2001\)](#), [Novak e Cañas \(2008\)](#), [Cañas et al. \(2004\)](#). Além dessas fontes, outras foram acessadas. As fontes acessadas encontram-se referenciadas nos roteiros.

## 9 Exemplo de uso do modelo de *framework*

Neste capítulo, o modelo de *framework* proposto é empregado na descrição parcial da arquitetura da informação atual (*baseline architecture*) em uma comunidade voltada ao desenvolvimento de *software* aberto. Neste capítulo são descritos resultados das seguintes atividades:

- Definir os critérios para a escolha do caso a ser estudado;
- Selecionar o caso a ser estudado;
- Selecionar as fontes de informação a serem acessadas;
- Definir as visões de arquitetura adotadas;
- Seguir os roteiros associados aos processos integrantes do modelo de *framework*;
- Construir os artefatos.

### 9.1 Seleção do caso estudado

O modelo de *framework* proposto nesta dissertação foi empregado na descrição da arquitetura da informação atual (*baseline architecture*) em uma comunidade voltada ao desenvolvimento de *software* aberto para a computação em nuvem (*cloud computing*). A escolha de uma comunidade voltada ao desenvolvimento de *software* aberto para a computação em nuvem se justifica pela importância desse modelo de computação. A seguir, definição do *National Institute of Standards and Technology* (NIST) para computação em nuvem:

*“Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models and four deployment models”* (MELL; GRANCE, 2011).

O termo computação em nuvem pode designar tanto o provimento de serviços de *software* via Internet, quanto os centros de processamento usados para prover esses serviços (ARMBRUST et al., 2010). Nuvens computacionais são compostas por *hardwares*, *softwares* e serviços, os quais podem ser prestados via diferentes abordagens. Por exemplo: infraestrutura como serviço, plataforma como serviço, *software* como serviço ou dados

como serviço. Na abordagem infraestrutura como serviço, os provedores disponibilizam infraestrutura para armazenamento e processamento, enquanto na abordagem plataforma como serviço, os provedores disponibilizam ferramentas para o desenvolvimento e a execução de aplicativos. Na abordagem *software* como serviço, os provedores disponibilizam serviços que substituem aplicativos que seriam instalados nos clientes e, por fim, na abordagem dado como serviço, os provedores disponibilizam dados que serão acessados como se estivessem nos clientes.

A popularização do modelo de computação em nuvem decorre das potenciais vantagens associadas a esse modelo. Por exemplo: compartilhamento de recursos; disponibilização de recursos para contratação; sistemas gerenciados de modo transparente aos usuários; maximização do uso de recursos; possibilidade do uso de recursos por demanda; automação na alocação e liberação de recursos; custos proporcionais ao uso dos recursos; aumento da mobilidade e das oportunidades de colaboração; redução da variedade de plataformas para as quais produtos de *software* precisam ser desenvolvidos; maior flexibilidade e redução de custos para os clientes dos serviços computacionais; redução da necessidade dos clientes planejarem, com grande antecedência, o atendimento de demandas futuras; possibilidade dos clientes aumentarem ou diminuïrem dinamicamente os recursos computacionais, reduzindo a necessidade de infraestruturas dimensionadas em função de picos de demanda e minimizando desperdícios; redução da necessidade de compra de produtos; possibilidade de empresas pequenas acessarem serviços e recursos que estariam fora das suas capacidades; disponibilização de mecanismos para recuperação de desastres e continuidade dos negócios (ARMBRUST et al., 2010; HAYES, 2008; MAHMOOD, 2011; ROSENBERG; MATEOS, 2011; WANG et al., 2010).

Com o intuito de selecionar a comunidade a ser estudada, foi consultado o sítio Ohloh (OHLOH, 2013). Esse sítio, cuja página inicial é apresentada na figura 10, contém informações acerca de vários projetos de *software* aberto. As consultas ocorreram em 27/11/2012 e em 27/6/2013. Para selecionar a comunidade a ser estudada, foram definidos os seguintes critérios: projeto com mais de cem colaboradores no ano anterior às consultas; projeto identificado pelas etiquetas *cloudcomputing* ou *cloud\_computing*. A partir da análise de dados dos projetos que atenderam aos requisitos anteriormente relacionados, foi selecionada a comunidade do projeto OpenStack, por ser aquela com maior número de contribuições e colaboradores nos doze meses anteriores às datas das consultas. A comunidade do projeto OpenStack desenvolve produtos de *software* para a computação em nuvem.

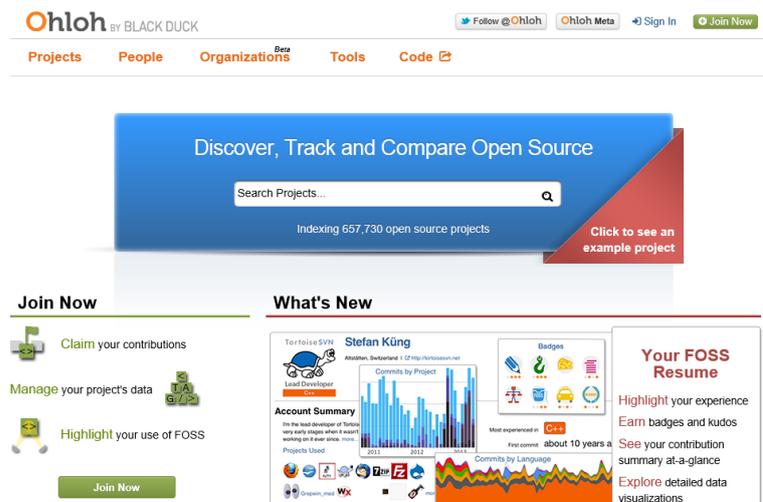


Figura 10 – Página inicial do portal do sítio Ohloh (OHLOH, 2013).

## 9.2 Fontes de informação

Uma importante fonte de informação para os membros da comunidade integrante do projeto OpenStack é um portal localizado em <http://www.openstack.org/>. A página inicial desse portal é apresentada na figura 11. As seguintes fontes de informação, acessadas pelos membros dessa comunidade, foram identificadas por meio de auditoria aos conteúdos desse portal:

- Guias, manuais, glossários, artigos, relatórios técnicos, padrões;
- Notas de distribuição (*release notes*) e boletins (*newsletters*);
- Códigos de produtos de *software*;
- Pessoas e grupos de usuários;
- Eventos como congressos e encontros;
- Vídeos;
- Páginas em portal, *wiki* e *blog*;
- Canais *Internet Relay Chat* (IRC), listas de discussão e fóruns.

## 9.3 Visões de arquitetura adotadas

Nesta dissertação, a arquitetura da informação é descrita segundo visões consideradas importantes a *stakeholders* que sejam contribuidores individuais (*individual*

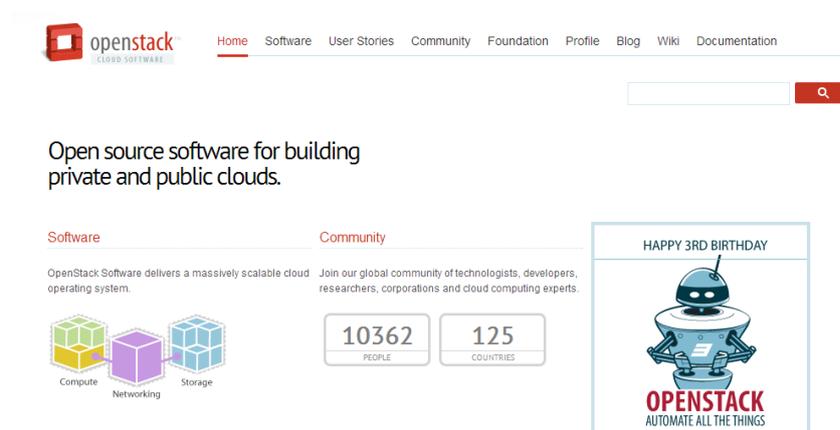


Figura 11 – Página inicial do portal do projeto OpenStack (OPENSTACK, 2013).

*contributor*) na comunidade e assumam um ou mais dos seguintes papéis identificados em OpenStack (2013):

- Desenvolvedor;
- Testador;
- Documentador;
- Tradutor.

## 9.4 Processos executados e artefatos construídos

Nesta etapa da pesquisa, processos no modelo de *framework* foram executados e artefatos construídos. Os processos executados e os artefatos construídos são relacionados nas tabelas 14 a 17. Esses artefatos foram construídos seguindo-se os roteiros nos apêndices A e B desta dissertação e acessando-se as fontes de informação referenciadas nesses roteiros.

Tabela 14 – Processos e artefatos na fase Iniciação.

| PROCESSO                        | ARTEFATO                         |
|---------------------------------|----------------------------------|
| Desenvolver termo de abertura   | Termo de abertura do projeto     |
| Identificar partes interessadas | Registro das partes interessadas |

## 9.5 Descrições dos artefatos de gerenciamento construídos

Os seguintes artefatos de gerenciamento foram construídos: Termo de abertura do projeto; Registro das partes interessadas; Plano de gerenciamento do projeto; Enunciado

Tabela 15 – Processos e artefatos na fase Organização e preparação.

| PROCESSO  | ARTEFATO                                    |
|---|---|
| Desenvolver plano de gerenciamento  | Plano de gerenciamento do projeto           |
| Coletar requisitos  | Enunciado do escopo do projeto              |
| Definir escopo  | Lista de atividades                         |
| Definir atividades  | Lista de marcos                             |
| Sequenciar atividades   | Diagrama de sequência de atividades         |
| Estimar durações das atividades   | Estimativas de durações das atividades      |
| Desenvolver cronograma  | Cronograma                                  |
| Confirmar <i>frameworks</i> de suporte e governança                       | Catálogo de princípios de arquitetura       |
| Identificar e estabelecer princípios de arquitetura                       | <i>Framework</i> de arquitetura configurado |
| Configurar o TOGAF e outros <i>frameworks</i> de arquitetura selecionados |   |
| Implementar ferramentas de arquitetura                                    |   |

Tabela 16 – Processos e artefatos na fase Execução do trabalho.

| PROCESSO   | ARTEFATO                                  |
|--|---|
| Selecionar modelos, pontos de vista e ferramentas              | Catálogo de papéis                        |
| Desenvolver descrição da arquitetura atual ( <i>baseline</i> ) | Catálogo de serviço/função de negócio     |
| Controlar cronograma   | Catálogo de localização                   |
|  | Catálogo processo/evento/controle/produto |
|  | Catálogo de casos de uso de negócio       |
|  | Diagrama de casos de uso de negócio       |
|  | Diagrama de fluxo de processo             |
|  | Diagrama de eventos                       |
|  | Taxonomia                                 |

Tabela 17 – Processos e artefatos na fase Encerramento.

| PROCESSO                 | ARTEFATO      |
|--------------------------|---------------|
| Encerrar Projeto ou Fase | Produto final |

do escopo do projeto; Lista de atividades; Lista de marcos; Diagrama de sequência de atividades; Estimativas de durações das atividades; Cronograma. Esses artefatos, exceto o “Diagrama de sequência de atividades”, contém as seguintes informações: título do documento, nome do projeto, descrições dos objetivos do documento, histórico de alterações, referências a fontes consultadas. Além dessas informações, cada artefato contém as informações descritas a seguir.

O artefato “Termo de abertura do projeto” contém as seguintes informações: descrição resumida do projeto, objetivos do projeto, problemas e oportunidades, escopo, premissas e restrições, partes interessadas, nome do responsável pelo projeto, papéis e responsabilidades dos membros da equipe do projeto, marcos do projeto, recursos necessários ao projeto, riscos e estratégias de mitigação, medidas de sucesso, domínios e pontos de vista de arquitetura. O artefato “Registro das partes interessadas” contém as seguintes informações: partes interessadas, expectativas das partes interessadas, classificações das partes interessadas, artefatos importantes a cada parte interessada. O artefato “Plano de gerenciamento do projeto” contém as seguintes informações: descrição resumida do projeto, objetivos do projeto, escopo do projeto, papéis e responsabilidades dos membros da equipe, marcos, recursos necessários, ciclo de vida do projeto, processos e artefatos, ferramentas, métodos, premissas e restrições. O artefato “Enunciado do escopo do projeto” contém as seguintes informações: descrição resumida do projeto, partes interessadas, escopo, requisitos fora do escopo do projeto, relação de entregas. O artefato “Lista de atividades” contém, para cada atividade, as seguintes informações: nome, tempo estimado, responsável, recursos, premissas e restrições, localização. O artefato “Lista de marcos” contém, para cada marco, as seguintes informações: nome, data, entregas. O artefato “Diagrama de sequência de atividades” contém, para cada atividade, as seguintes informações: nome, data de início e data de término. O artefato “Estimativas de durações das atividades” contém, para cada atividade, as seguintes informações: nome e duração. O artefato “Cronograma” contém: para cada atividade, a data de início, a data de término e a duração; para cada marco, nome e data.

## 9.6 Descrições dos artefatos de arquitetura construídos

Nesta pesquisa, os artefatos de arquitetura foram construídos a partir da análise do modelo de processo de desenvolvimento de *software* aberto descrito em [Lonchamp \(2005\)](#) e de informações em páginas e documentos no portal e *wiki* da comunidade do projeto OpenStack. Na tabela 18, encontram-se as macroatividades e atividades descritas por [Lonchamp \(2005\)](#) e na tabela 19 estão relacionados os principais documentos acessados (os endereços das páginas acessadas não encontram-se listados devido ao seu elevado número).

Tabela 18 – Desenvolvimento de *software* aberto segundo Lonchamp (2005).

| Macroatividade                | Atividade   |
|-------------------------------|---|
| Desenvolver código            | Realizar <i>check-out</i> e transferir ( <i>download</i> ) código<br>Editar<br>Compilar<br>Depurar<br>Realizar teste de unidade<br>Submeter código ( <i>patch</i> )   |
| Testar código                 | Transferir ( <i>download</i> ) versão integrada do código em desenvolvimento ( <i>build</i> )<br>Realizar teste<br>Analisar resultados de testes<br>Reportar defeito ( <i>bug</i> ) ou problema ( <i>issue</i> )  |
| Realizar outras contribuições | Comunicar requisitos, problemas e aspectos de projeto<br>Reportar defeito<br>Solicitar característica ( <i>feature</i> )<br>Participar de tomada de decisão<br>Analisar defeito ( <i>bug</i> )<br>Escrever documentação<br>Atuar como mentor de outros participantes ( <i>mentoring</i> ) |

Tabela 19 – Endereços de documentos acessados.

| Documento                | Endereço  | Data de acesso       |
|--------------------------|---|----------------------|
| How To Contribute        | <a href="https://wiki.openstack.org/wiki/How_To_Contribute">https://wiki.openstack.org/wiki/How_To_Contribute</a>             | 14 de agosto de 2013 |
| Release Cycle            | <a href="https://wiki.openstack.org/wiki/ReleaseCycle">https://wiki.openstack.org/wiki/ReleaseCycle</a>                       | 14 de agosto de 2013 |
| Blueprints               | <a href="https://wiki.openstack.org/wiki/Blueprints">https://wiki.openstack.org/wiki/Blueprints</a>                           | 14 de agosto de 2013 |
| Bugs                     | <a href="https://wiki.openstack.org/wiki/Bugs">https://wiki.openstack.org/wiki/Bugs</a>                                       | 14 de agosto de 2013 |
| Branch Model             | <a href="https://wiki.openstack.org/wiki/BranchModel">https://wiki.openstack.org/wiki/BranchModel</a>                         | 14 de agosto de 2013 |
| BugTriage                | <a href="https://wiki.openstack.org/wiki/BugTriage">https://wiki.openstack.org/wiki/BugTriage</a>                             | 14 de agosto de 2013 |
| Gerrit Workflow          | <a href="https://wiki.openstack.org/wiki/GerritWorkflow">https://wiki.openstack.org/wiki/GerritWorkflow</a>                   | 14 de agosto de 2013 |
| Documentation            | <a href="https://wiki.openstack.org/wiki/Documentation">https://wiki.openstack.org/wiki/Documentation</a>                     | 14 de agosto de 2013 |
| Documentaion How/To      | <a href="https://wiki.openstack.org/wiki/Documentation/HowTo">https://wiki.openstack.org/wiki/Documentation/HowTo</a>         | 14 de agosto de 2013 |
| Summit                   | <a href="https://wiki.openstack.org/wiki/Summit">https://wiki.openstack.org/wiki/Summit</a>                                   | 14 de agosto de 2013 |
| Website                  | <a href="https://wiki.openstack.org/wiki/Website">https://wiki.openstack.org/wiki/Website</a>                                 | 14 de agosto de 2013 |
| Translations             | <a href="https://wiki.openstack.org/wiki/Translations">https://wiki.openstack.org/wiki/Translations</a>                       | 14 de agosto de 2013 |
| Vulnerability Management | <a href="https://wiki.openstack.org/wiki/VulnerabilityManagement">https://wiki.openstack.org/wiki/VulnerabilityManagement</a> | 15 de agosto de 2013 |

### 9.6.1 Catálogo de princípios de arquitetura

O artefato “Catálogo de princípios de arquitetura” contém descrições da missão e dos princípios de negócio. A seguir encontra-se a descrição da missão de negócio localizada na página principal de *wiki* em [OpenStack \(2013\)](#):

*“OpenStack project mission is to produce the ubiquitous Open Source Cloud Computing platform that will meet the needs of public and private clouds regardless of size, by being simple to implement and massively scalable.”*

Quanto aos princípios de negócio relacionados no artefato “Catálogo de princípios de arquitetura”, os seguintes foram identificados em [OpenStack \(2013\)](#): Modelo de desenvolvimento aberto; Comunidade aberta; Processo de projeto aberto; Autogerência da comunidade. Na figura 12, como exemplo, encontra-se descrição, presente no artefato “Catálogo de princípios de arquitetura”, de um dos princípios de negócio anteriormente relacionados.

|                     |  |
|---------------------|--|
| Nome                | Modelo de desenvolvimento aberto   |
| Enunciado           | Todo código OpenStack é livremente disponível de acordo com a licença Apache 2.0.  |
| Motivações          |  |
| Implicações         | Adesão a princípios de uso, reprodução e distribuição definidos na licença.  |
| Fonte de informação | OpenStack Community Welcome Guide<br>(Em: < <a href="http://www.openstack.org/assets/openstack-welcome-guide.pdf">http://www.openstack.org/assets/openstack-welcome-guide.pdf</a> >, Acessado em 15 de julho de 2013). |

Figura 12 – Exemplo de descrição de princípio.

### 9.6.2 Framework de arquitetura configurado

O artefato “*Framework* de arquitetura configurado” descreve o modelo de *framework* proposto. Esse artefato contém as seguintes informações: título do documento, nome do projeto, descrições dos objetivos do documento, histórico de alterações do documento, descrição do ciclo de vida do projeto, relação de processos e artefatos, descrições de roteiros.

### 9.6.3 Catálogo de papéis

O artefato “Catálogo de papéis” contém descrições dos papéis dos membros da comunidade do projeto OpenStack. Os seguintes papéis foram identificados em [OpenStack \(2013\)](#): Membro de conselho consultivo (*advisory board*); Líder técnico de projeto (*project technical lead*); Membro de comitê técnico de projeto (*technical committee*); Gerente de distribuição (*release manager*); Gerente de desenvolvimento (*development manager*); Gerente de comunidade (*community manager*); Líder de componente (*component lead*);

Desenvolvedor central (*core developer*); Contribuidor (*contributor*); Diretor (*board of directors*); Membro de comitê de usuários (*user committee*). Na figura 13, como exemplo, encontra-se a descrição presente no artefato “Catálogo de papéis”, de um desses papéis.

|                     |  |
|---------------------|--|
| Nome                | Líder Técnico de Projeto ( <i>Project Technical Lead</i> )   |
| Descrição           | Responsável pela liderança técnica de um projeto. Entre as suas responsabilidades se pode destacar: definir o roteiro ( <i>roadmap</i> ) para o projeto; participar de encontros visando interagir com os membros da equipe; decidir conflitos em debates técnicos; gerenciar <i>release</i> ; definir tema e plano para o próximo release; definir prioridades; aprovar desenhos ( <i>design</i> ) propostos. |
| Fonte de informação | PTLguide (Em:< <a href="https://wiki.openstack.org/wiki/PTLguide">https://wiki.openstack.org/wiki/PTLguide</a> >, Acessado em 16 de julho de 2013).<br>Gov (Em:< <a href="https://wiki.openstack.org/wiki/Gov">https://wiki.openstack.org/wiki/Gov</a> >, Acessado em 18 de julho de 2013).  |

Figura 13 – Exemplo de descrição de papel.

#### 9.6.4 Catálogo de serviço/função de negócio

O artefato “Catálogo de serviço/função de negócio” contém as descrições de serviços e funções de negócio na comunidade. Um serviço de negócio representa o valor adicionado que uma organização entrega ao seu ambiente, enquanto uma função de negócio é uma área à qual a organização deseja dar atenção (BERG et al., 2013). Foram identificados os seguintes serviços de negócio: Adesão, Desenvolvimento de código, Desenvolvimento de sítio, Divulgação, Documentação, Suporte, Teste, Tradução. Por sua vez, as seguintes funções de negócio foram identificadas: Administração, Desenvolvimento, *Marketing*, Recursos humanos. Esses serviços e funções de negócio foram identificados por meio de acessos a páginas no portal da comunidade. Nas figuras 14 e 15, encontram-se exemplos de descrições de serviço e função de negócio, como presentes no artefato “Catálogo de serviço/função de negócio”.

|                 |  |
|-----------------|--|
| Nome do serviço | Divulgação de eventos e oportunidades  |
| Descrição       | Divulgação de eventos e oportunidades de emprego para os membros da comunidade.  |
| Fonte           | OpenStack Event Listing (Em:< <a href="http://www.openstack.org/community/events/">http://www.openstack.org/community/events/</a> >, Acessado em 19 de julho de 2013). |

Figura 14 – Exemplo de descrição de serviço de negócio.

|                |   |
|----------------|---|
| Nome da função | Administração   |
| Descrição      | Administração técnica, estratégica e financeira de recursos físicos, humanos e financeiros empregados nos projetos.   |
| Fonte          | The OpenStack Foundation (Em:< <a href="http://www.openstack.org/foundation/">http://www.openstack.org/foundation/</a> >, Acessado em 19 de julho de 2013). |

Figura 15 – Exemplo de descrição de função de negócio.

### 9.6.5 Catálogo de localização

O artefato “Catálogo de localização” contém relação das localizações das operações do negócio ou onde são mantidos artefatos importantes ao negócio. As seguintes localizações foram identificadas: portal do projeto OpenStack, portais de grupos de usuários, sítio do serviço *wiki* do projeto OpenStack, sítios dos serviços Twitter, Flickr, YouTube, Facebook, LinkedIn, Ohloh, Launchpad, Freenode, Transifex e GitHub. Na figura 16 encontra-se exemplo de descrição de localização, como presente no artefato “Catálogo de localização”.

|                     |  |
|---------------------|--|
| Nome da localização | Sítio da plataforma Transifex  |
| Descrição           | Sítio da plataforma colaborativa de tradução Transifex. A comunidade de tradutores do projeto OpenStack pode ser localizada em <a href="https://www.transifex.com/projects/p/openstack/">https://www.transifex.com/projects/p/openstack/</a> . |

Figura 16 – Exemplo de descrição de localização.

### 9.6.6 Modelo de casos de uso de negócio

Modelos de caso de uso são compostos por atores, casos de uso, relacionamentos entre atores, relacionamentos entre casos de uso, relacionamentos entre atores e casos de uso (KULAK; GUINEY, 2003). Casos de uso podem ser de negócio ou de sistema. Segundo Langlands e Edwards (2009), um caso de uso de negócio (*business use case*) descreve como um ator usa o negócio para obter resultados desejados. Esses autores destacam que a execução de um caso de uso de negócio incorpora todas as atividades necessárias à realização do que o ator deseja. Um caso de uso de negócio descreve uma interação entre o ator, partes importantes do negócio e atores de suporte. A execução de um caso de uso de negócio pode envolver várias unidades organizacionais e ter uma duração breve ou longa.

Nesta pesquisa, o modelo de casos de uso de negócio encontra-se descrito nos artefatos “Catálogo de casos de uso de negócio” e “Diagrama de casos de uso de negócio”. Os nomes dos atores identificados, assim como descrições dos mesmos, encontram-se na tabela 20. Na tabela 21, os atores são relacionados aos casos de uso identificados. O artefato

“Diagrama de casos de uso” foi construído com a ferramenta StarUML (STARUML, 2013). Na figura 17 encontra-se um dos diagramas construídos com essa ferramenta. O artefato “Catálogo de casos de uso de negócio” contém as descrições dos casos de uso. Na figura 18 encontra-se a descrição de um dos casos de uso.

Tabela 20 – Nomes e descrições dos atores.

| Ator                  | Descrição   |
|-----------------------|---|
| Usuário               | Usuário de serviços do negócio.   |
| Contribuidor          | Agrega valor aos componentes do projeto.  |
| Desenvolvedor         | Codifica novas características ( <i>features</i> ), corrige defeitos melhora a qualidade de código. |
| Testador              | Testa software, reporta defeito, confirma defeito, atribui prioridade a defeito reportado.          |
| Documentador          | Elabora e revisa documentação.  |
| Tradutor              | Converte materiais de uma linguagem para outra.   |
| Contribuidor do sítio | Provê <i>feedback</i> acerca do sítio, cria conteúdo para o sítio e posta no <i>blog</i> .          |

Tabela 21 – Relações entre atores e casos de uso.

| ATOR                  | CASO DE USO  |
|-----------------------|--|
| Usuário               | Aderir à fundação  |
| Contribuidor          | Requerer suporte<br>Prover suporte   |
| Desenvolvedor         | Instruir correção de <i>bug</i><br>Corrigir <i>bug</i><br>Melhorar a qualidade de código<br>Desenvolver nova característica ( <i>feature</i> ) |
| Testador              | Notificar bug<br>Notificar vulnerabilidade<br>Realizar triagem de bug  |
| Documentador          | Revisar documento<br>Submeter documento  |
| Tradutor              | Traduzir documento<br>Traduzir código<br>Internacionalizar código<br>Localizar código  |
| Contribuidor do sítio | Criar conteúdo para o sítio<br>Postar em <i>blog</i><br>Prover <i>feedback</i> acerca do sítio   |

### 9.6.7 Taxonomia

No artefato “Taxonomia” encontra-se taxonomia hierárquica composta por conceitos e termos identificados em páginas do portal e *wiki* da comunidade. Os conceitos e termos foram identificados por meio de auditoria de conteúdos. Essa auditoria foi realizada de

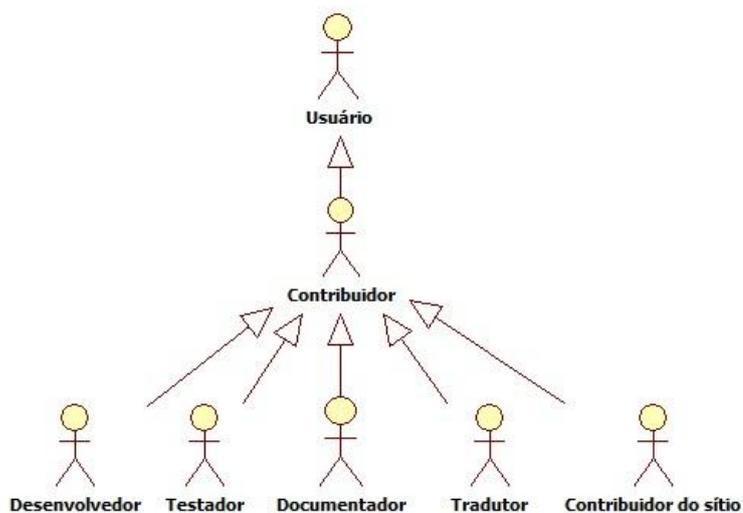


Figura 17 – Exemplo de diagrama do modelo de caso de uso.

|                          |  |
|--------------------------|--|
| Nome do caso de uso      | Requerer suporte   |
| Descrição do caso de uso | O contribuidor emprega serviço do negócio com o intuito de obter suporte de membros da comunidade na resposta a questionamento acerca de produtos ou serviços do projeto.  |
| Atores                   | Contribuidor   |
| Fluxo principal          | <ol style="list-style-type: none"> <li>Contribuidor acessa informações disponíveis com o intuito de responder ao seu questionamento acerca de produtos ou serviços providos pelo projeto.</li> <li>Se localizar informações que respondam ao seu questionamento, o contribuidor acessa as informações e não submete um novo questionamento.</li> </ol>             |
| Fluxos alternativos      | <ol style="list-style-type: none"> <li>Se o contribuidor não localizar informações que respondam ao seu questionamento, o contribuidor submete questionamento no qual descreve a sua dúvida.</li> <li>As informações acerca do questionamento do contribuidor são armazenadas para que possam ser posteriormente consultadas por membros da comunidade.</li> </ol> |
| Pré-condições            |  |

Figura 18 – Exemplo de descrição de caso de uso.

30 de julho a 12 de agosto de 2013. Para armazenar os dados resultantes da auditoria, foi construída planilha com uma linha para cada página e colunas para as seguintes informações:

- Identificador de linha;
- Identificador de nível hierárquico;
- URL;
- Metadados;

- Tipo do documento;
- Termos identificados.

Para identificar conceitos e termos, além dos conteúdos em páginas de portal e *wiki* da comunidade, metadados associados a essas páginas foram acessados. Para facilitar o acesso a esses metadados, foi empregada a ferramenta *Meta Tag Analyzer* (SEOCENTRO, 2013). A figura 19 ilustra resultados do uso dessa ferramenta. A estrutura da planilha segue sugestões presentes em Hedden (2010). A figura 20 ilustra a estrutura da planilha empregada para a coleta de dados durante a auditoria.

**Meta Tag Analyzer**

Menu

- » Home
- » SEO Archives
- » SEO Articles
- » SEO Services

SEO Tools

- » Meta Tag Analyzer
- » Rank Checker
- » PageRank Check
- » Keyword Density
- » Keyword Analyzer
- » Social Link Popularity
- » Keyword Suggestion

Promotion Tools

- » QR Code Generator
- new
- » Bookmark Buttons
- » RSS Subscribe Buttons

URL:

Spider Robots (optional)

User Agent (optional):

color = Error color = Warn color = Good

**Status.**

|                 |                            |
|-----------------|----------------------------|
| Status:         | 200 OK                     |
| Web Server:     | Apache/2.2                 |
| Content Type:   | text/html; charset="utf-8" |
| Charset:        | "UTF-8"                    |
| Content Length: | 13331                      |

**Meta tags report for: http://www.openstack.org/**

Figura 19 – Ferramenta para coleta de metadados.

Devido ao elevado número de páginas e documentos no portal e *wiki* da comunidade, apenas parte dos mesmos foi auditada. Segundo Hedden (2010), quando o número de páginas em um sítio é elevado, uma alternativa é auditar páginas representativas. Nesta pesquisa foram auditadas as informações acessadas por meio de 120 distintos endereços eletrônicos (distintos *uniform resource locators*). Uma dos documentos era o “*OpenStack Glossary*” em OpenStack (2013), um glossário com definições para vários termos empregados na comunidade.

Na taxonomia construída nesta pesquisa, existem relacionamentos hierárquicos e de equivalência entre termos. Como sugerido em NISO (2005), os relacionamentos de equivalência são identificados pelos códigos USE e USED FOR. Os relacionamentos hierárquicos abrangem os seguintes relacionamentos: generalização, instância e todo-parte. Como recomendado em NISO (2005), os relacionamentos hierárquicos são identificados com os seguintes códigos: BTG (*Broader Term Generic*), NTG (*Narrow Term Generic*), BTI (*Broader Term Instance*), NTI (*Narrow Term Instance*), BTP (*Broader Term Partitive*) e NTP (*Narrow Term Partitive*).

| IDENTIFICADOR | NÍVEL 1  | NÍVEL 2         | NÍVEL 3                               | NÍVEL 4 | NÍVEL 5 | URL   |
|---------------|----------|-----------------|---------------------------------------|---------|---------|---|
| 1.            | Home     |                 |                                       |         |         | <a href="http://www.openstack.org/">http://www.openstack.org/</a>   |
| 2.            | Software |                 |                                       |         |         | <a href="http://www.openstack.org/software/">http://www.openstack.org/software/</a>   |
| 2.1.          |          | Compute         |                                       |         |         | <a href="http://www.openstack.org/software/openstack-compute/">http://www.openstack.org/software/openstack-compute/</a>   |
| 2.1.1.        |          |                 | storage                               |         |         | <a href="http://www.openstack.org/software/openstack-storage/">http://www.openstack.org/software/openstack-storage/</a>   |
| 2.1.2.        |          |                 | networking                            |         |         | <a href="http://www.openstack.org/software/openstack-networking/">http://www.openstack.org/software/openstack-networking/</a>                                     |
| 2.1.3.        |          |                 | dashboard                             |         |         | <a href="http://www.openstack.org/software/openstack-dashboard/">http://www.openstack.org/software/openstack-dashboard/</a>                                       |
| 2.1.4.        |          |                 | Flexible networking models            |         |         | <a href="http://www.openstack.org/software/openstack-networking/">http://www.openstack.org/software/openstack-networking/</a>                                     |
| 2.1.5.        |          |                 | store, import, share and query images |         |         | <a href="http://www.openstack.org/software/openstack-shared-services/">http://www.openstack.org/software/openstack-shared-services/</a>                           |
| 2.1.6.        |          |                 | OpenStack dashboard                   |         |         | <a href="http://www.openstack.org/software/openstack-dashboard/">http://www.openstack.org/software/openstack-dashboard/</a>                                       |
| 2.2.          |          | Storage         |                                       |         |         | <a href="http://www.openstack.org/software/openstack-storage/">http://www.openstack.org/software/openstack-storage/</a>   |
| 2.3.          |          | Networking      |                                       |         |         | <a href="http://www.openstack.org/software/openstack-networking/">http://www.openstack.org/software/openstack-networking/</a>                                     |
| 2.4.          |          | Dashboard       |                                       |         |         | <a href="http://www.openstack.org/software/openstack-dashboard/">http://www.openstack.org/software/openstack-dashboard/</a>                                       |
| 2.5.          |          | Shared Services |                                       |         |         | <a href="http://www.openstack.org/software/openstack-shared-services/">http://www.openstack.org/software/openstack-shared-services/</a>                           |
| 2.6.          |          | Getting Started |                                       |         |         | <a href="http://www.openstack.org/software/start/">http://www.openstack.org/software/start/</a>   |
| 2.6.1.        |          |                 | User Stories                          |         |         | <a href="http://www.openstack.org/user-stories/">http://www.openstack.org/user-stories/</a>   |
| 2.6.2.        |          |                 | OpenStack Summit Videos               |         |         | <a href="http://www.openstack.org/summit/san-diego-2012/openstack-summit-sessions/">http://www.openstack.org/summit/san-diego-2012/openstack-summit-sessions/</a> |
| 2.7.          |          | Latest Release  |                                       |         |         | <a href="http://www.openstack.org/software/grizzly/">http://www.openstack.org/software/grizzly/</a>   |
| 2.7.1.        |          |                 | Press release                         |         |         | <a href="http://www.openstack.org/grizzly/press-release/">http://www.openstack.org/grizzly/press-release/</a>   |
| 2.7.1.1.      |          |                 | OpenStack® Grizzly                    |         |         | <a href="http://www.openstack.org/software/grizzly/">http://www.openstack.org/software/grizzly/</a>   |
| 2.7.1.2.      |          |                 | OpenStack Summit                      |         |         | <a href="http://www.openstack.org/summit/portland-2013/">http://www.openstack.org/summit/portland-2013/</a>   |
| 2.7.2.        |          |                 | Take the User Survey                  |         |         | <a href="https://www.openstack.org/user-survey/Login">https://www.openstack.org/user-survey/Login</a>   |
| 2.8.          |          | Roadmap         |                                       |         |         | <a href="http://www.openstack.org/software/roadmap/">http://www.openstack.org/software/roadmap/</a>   |

| META TAG | VALOR  | TIPO      | TERMOS  |
|----------|--|-----------|---|
| Title    | Home » OpenStack Open Source Cloud Computing Software  | text/html | open source, cloud, cloud computing, private c    |
| Title    | Software » OpenStack Open Source Cloud Computing Software  | text/html | open source cloud operating system, compute       |
| Title    | OpenStack Compute » OpenStack Open Source Cloud Computing Software   | text/html | virtual machine, network of virtual machine, cl   |
|          |  |           |   |
|          |  |           |   |
|          |  |           |   |
|          |  |           |   |
|          |  |           |   |
| Title    | OpenStack Storage » OpenStack Open Source Cloud Computing Software   | text/html | storage, object storage, block storage, server, s |
| Title    | OpenStack Networking » OpenStack Open Source Cloud Computing Software  | text/html | networking, data center network, network equ      |
| Title    | OpenStack Dashboard » OpenStack Open Source Cloud Computing Software   | text/html | dashboard, administrator, user, service provid    |
| Title    | OpenStack Shared Services » OpenStack Open Source Cloud Computing Software   | text/html | compute, storage, networking, cloud, shared s     |
| Title    | How To Get Started With OpenStack » OpenStack Open Source Cloud Computing Software                                 | text/html | public cloud, VM, user, dashboard, admin, dev     |
|          |  |           |   |
| Title    | OpenStack Summit Sessions » OpenStack Open Source Cloud Computing Software   | text/html |   |
| Title    | OpenStack Grizzly Release » OpenStack Open Source Cloud Computing Software   | text/html | open source software, public cloud, private clo   |
| Title    | Grizzly Release Positions OpenStack at Center of Cloud Innovation » OpenStack Open Source Cloud Computing Software | text/html | networking technology, enterprise technolog       |
|          |  |           |   |
| Title    | OpenStack Summit Portland 2013 » OpenStack Open Source Cloud Computing Software                                    | text/html |   |
| Title    | User Survey » OpenStack Open Source Cloud Computing Software   | text/html |   |
| Title    | OpenStack Roadmap » OpenStack Open Source Cloud Computing Software   | text/html | roadmap, design process, development proces       |

Figura 20 – Estrutura da planilha empregada na auditoria.

Os termos integrantes da taxonomia foram armazenados em uma planilha com uma coluna para cada nível hierárquico e colunas para termos não preferidos. Como sugerido em NISO (2005), os níveis hierárquicos são identificados por dígitos (NTG1, NTG2, etc.). A figura 21 ilustra pequeno trecho da planilha contendo a taxonomia construída. Na escolha dos termos foram observadas as seguintes recomendações em Hedden (2010): o conceito deve estar dentro do escopo da área da taxonomia; o conceito deve ser importante aos usuários; o conceito deve conter informação suficiente; os usuários devem esperar que o conceito seja abordado. Os termos estão em inglês, por ser esse o idioma dos conteúdos.

### 9.6.8 Descrições de processos do negócio

Segundo o *framework* proposto nesta dissertação, processos são descritos via “Catálogo processo/evento/controle/produtos”, “Diagrama de fluxo de processo” e “Diagrama de eventos”. No catálogo, são relacionados eventos e processos. Nesta pesquisa, eventos e processos foram identificados executando-se as seguintes atividades recomendadas em Jacka e Keller (2009):

| NÍVEL 0 | NÍVEL 1               | NÍVEL 2                | NÍVEL 3                           | NÍVEL 4                         | NÍVEL 5                           | TERMO 1    |
|---------|-----------------------|------------------------|-----------------------------------|---------------------------------|-----------------------------------|------------|
| member  | NTG1 community member |                        |                                   |                                 |                                   |            |
|         |                       | NTG2 foundation member |                                   |                                 |                                   |            |
|         |                       |                        | NTG3 board of directors member    |                                 |                                   |            |
|         |                       |                        |                                   | NTG4 chairman of the board      |                                   |            |
|         |                       |                        |                                   | NTG4 vice chairman of the board | USED FOR                          | vice chair |
|         |                       |                        | NTG3 committee member             |                                 |                                   |            |
|         |                       |                        |                                   | NTG4 technical committee member |                                   |            |
|         |                       |                        |                                   | NTG4 user committee member      |                                   |            |
|         |                       |                        | NTG3 contributor                  |                                 |                                   |            |
|         |                       |                        |                                   | NTG4 active contributor         |                                   |            |
|         |                       |                        |                                   |                                 | NTG5 active technical contributor |            |
|         |                       |                        | NTG3 individual foundation member |                                 |                                   |            |
|         |                       |                        |                                   | NTG4 foundation staff member    |                                   |            |
|         |                       |                        |                                   | NTG4 supporting cast member     |                                   |            |
|         |                       |                        | NTG3 sponsor                      |                                 |                                   |            |
|         |                       |                        |                                   | NTG4 corporate sponsor          |                                   |            |
|         |                       |                        |                                   | NTG4 gold member                |                                   |            |
|         |                       |                        |                                   | NTG4 platinum member            |                                   |            |
|         |                       | NTG2 user              |                                   |                                 |                                   |            |

Figura 21 – Planilha com alguns poucos elementos da taxonomia.

- Identificar eventos que provocam a execução dos processos;
- Identificar processos críticos;
- Identificar processos de suporte;
- Atribuir nomes aos processos.

Os processos identificados são os seguintes: Adesão à comunidade, Desenvolvimento de código, Desenvolvimento de sítio, Divulgação, Documentação, Suporte, Teste, Tradução. Nesta etapa da pesquisa, diagramas de processos e eventos foram construídos com a linguagem de modelagem ArchiMate ([The Open Group, 2012](#)) e a ferramenta Archi ([ARCHI, 2012](#)).

## 9.7 Considerações acerca dos artefatos de arquitetura construídos

Os artefatos de arquitetura construídos ao longo desta pesquisa não foram testados por usuários. Portanto, se encontram em um estado intermediário de desenvolvimento. Para que o desenvolvimento dos mesmos seja concluído, os artefatos precisam ser testados, os resultados dos testes analisados e os artefatos revisados. A importância de testes por usuários, por exemplo, no desenvolvimento de taxonomias, é destacado em [Hedden \(2010\)](#) e em [NISO \(2005\)](#). Em [\(NISO, 2005\)](#) é destacada a importância da realização de testes e avaliações dos vocabulários controlados, uma vez que vocabulários controlados existem para servir aos seus usuários, tanto aos usuários que empregam os vocabulários para descrever conteúdos, quanto aos que empregam os vocabulários para localizar informações.

## Considerações finais

Nesta dissertação foram descritos resultados de pesquisa que propôs um modelo de *framework* de arquitetura da informação no contexto organizacional. Para atingir esse objetivo, inicialmente foi realizada pesquisa bibliográfica que possibilitou as seguintes conclusões:

- Existem variados *frameworks* e métodos de arquitetura da informação, e existem *frameworks* e métodos de arquitetura da informação propostos para diversos domínios de aplicação;
- Entre documentações de *frameworks* de arquitetura da informação, um mesmo conceito pode ser identificado por diferentes termos, e nas documentações de alguns *frameworks* de arquitetura da informação existem conceitos com definições imprecisas;
- Alguns *frameworks* de arquitetura da informação são coleções de boas práticas com poucas referências para trabalhos científicos, e o nível de detalhamento das suas documentações é variável.

Após a pesquisa bibliográfica, foram realizadas atividades para propor um modelo de *framework* de arquitetura da informação no contexto organizacional, e o emprego desse modelo de *framework* na descrição da arquitetura da informação em uma comunidade voltada ao desenvolvimento de *software* aberto. Essas atividades possibilitaram as seguintes conclusões:

- A construção de artefatos prescritos em *frameworks* de arquitetura da informação pode requerer variadas ferramentas, e o esforço para a construção desses artefatos pode ser significativo;
- O emprego de *frameworks* de arquitetura da informação pode melhorar a organização da informação em desenvolvimento de *software*.

Quanto às contribuições da pesquisa, é possível destacar as seguintes:

- Descrição de conceitos acerca de *frameworks* de arquitetura da informação, método de arquitetura da informação, modelagem de arquitetura, linguagem para modelagem de arquitetura, *frameworks* e métodos para gerenciamento de projetos, sistemas para organização da informação, processos de desenvolvimento de *software*;
- Modelo de *framework* de arquitetura da informação no contexto organizacional, desenvolvido a partir de informações em TOGAF, PMBOK, MAIA e outras fontes.
- Exemplo de uso de um modelo de *framework* de arquitetura da informação.

Quanto às limitações da pesquisa, é possível destacar as seguintes:

- O modelo de *framework* de arquitetura da informação proposto foi desenvolvido considerando-se um contexto específico;
- Apenas uma comunidade voltada ao desenvolvimento de *software* aberto teve a arquitetura da informação descrita e essa arquitetura foi descrita em apenas um domínio;
- Não foram construídos todos os artefatos sugeridos no modelo de *framework* proposto nesta dissertação, e os artefatos construídos não foram testados por seus potenciais usuários.

Finalmente, algumas sugestões de pesquisas futuras:

- Ampliar o modelo de *framework* proposto para descrever a arquitetura da informação em domínios não contemplados nesta pesquisa;
- Ampliar o modelo de *framework* proposto com o intuito de contemplar o desenvolvimento de arquiteturas;
- Testar com usuários os artefatos resultantes desta pesquisa.

## Referências

- ABNT. *ABNT-NBR ISO 9000:2005 Sistemas de gestão da qualidade – Fundamentos e vocabulário*. 2005. Citado na página 64.
- ALBUQUERQUE, R.; 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 na página 30.
- ANDERSON, J. Organization of knowledge. In: FEATHER J. & STURGES, P. (Ed.). *International encyclopedia of information and library science*. 2nd edition. ed. London, UK: Routledge, 2003. p. 471–490. Citado 2 vezes nas páginas 19 e 52.
- ANKOLEKAR, A. Supporting online problem-solving communities with the semantic web. In: *Proceedings of the 15th International Conference on World Wide Web*. [S.l.]: ACM, 2004. p. p.575–584. Citado 2 vezes nas páginas 73 e 77.
- 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 4 vezes nas páginas 22, 61, 76 e 77.
- ARCHI. *Archi ArchiMate modelling*. 2012. Acessado em 30 de setembro de 2013. Disponível em: <<http://archi.cetis.ac.uk/>>. Citado na página 108.
- ARMBRUST, M. et al. A view of cloud computing. *Communications of the ACM*, v. 53, n. 4, p. 50–58, April 2010. Citado 2 vezes nas páginas 94 e 95.
- AUDY, J.; PRIKLADNICKI, R. *Desenvolvimento Distribuído de Software: Desenvolvimento de software com equipes distribuídas*. Rio de Janeiro: Elsevier, 2008. Citado na página 71.
- BACON, J. et al. Generic support for distributed applications. *Computer*, v. 33, n. 3, p. 2–10, March 2000. Citado na página 28.
- 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: [s.n.], 2010. p. 46 – 55. Citado 4 vezes nas páginas 19, 22, 23 e 70.
- BASS, L.; CLEMENTS, P.; KAZMAN, R. *Software Architecture in Practice*. 3rd. ed. Boston: Addison Wesley, 2012. 640 p. Citado 3 vezes nas páginas 28, 45 e 49.
- BECK, K. Embracing change with extreme programming. *Computer*, v. 32, n. 10, p. 70–77, October 1999. Citado na página 71.
- BERG, H. et al. *ArchiMate Made Practical: Modeling according to ArchiMate guided by a collection of good practices*. 4.0. ed. [S.l.], 2013. Citado 2 vezes nas páginas 65 e 102.

- BERNARD, S. *An Introduction to Enterprise Architecture*. 3rd. ed. [S.l.]: AuthorHouse, 2012. 340 p. Citado 6 vezes nas páginas 16, 32, 33, 34, 35 e 42.
- BIFULCO, A.; SANTORO, R. Collaborative networks and their breeding environments. In: \_\_\_\_\_. [S.l.]: The International Federation for Information Processing, 2005. cap. A Conceptual Framework for “Professional Virtual Communities”, p. 417–412. Citado 2 vezes nas páginas 72 e 73.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *The Unified Modeling Language User Guide*. 2nd. ed. Upper Saddle River, NJ: Addison-Wesley Professional, 2005. 496 p. Citado na página 50.
- BORKO, H. Information science: What is it ? *American Documentation*, v. 19, n. 1, p. 3–5, 1968. Citado na página 19.
- BORST, W. *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. Tese (Doutorado) — Centre for Telematics and Information Technology, The Netherlands, 1997. Citado 2 vezes nas páginas 58 e 59.
- BRINK, T.; GERGLE, D.; WOOD, S. *Designing web sites that work: Usability for the web*. 1st. ed. San Francisco: Morgan Kaufmann Publishers, 2002. 481 p. Citado na página 35.
- BUSCHMANN, F. et al. *Pattern-Oriented Software Architecture: A System of Patterns*. 1st. ed. New York: John Wiley & Sons, 1996. 457 p. Citado na página 47.
- CALERO, C.; RUIZ, F.; PIATTIN, M. *Ontologies for Software Engineering and Software Technology*. New York: Springer, 2006. 339 p. Citado na página 61.
- CAÑAS, A. et al. Cmaptools: A knowledge modeling and sharing environment. In: UNIVERSIDAD PÚBLICA DE NAVARRA. *Proceedings of the first international conference on concept mapping*. Pamplona, Spain, 2004. v. 1, p. 125–133. Citado 2 vezes nas páginas 93 e 132.
- 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 29.
- CHEN, I. The factors influencing members’ continuance intentions in professional virtual communities – a longitudinal study. *Journal of Information Science*, v. 33, n. 4, p. 451–467, 2007. Citado na página 73.
- CHOO, C. *Information Management for the Intelligent Organization: The art of scanning the environment*. 3rd. ed. [S.l.]: Information Today, Inc., 2002. 325 p. (ASIST Monograph Series). Citado na página 52.
- CLEMENTS, P. Understanding architectural influences and decisions in large-system projects. In: *First International Workshop on Architectures for Software Systems*. [S.l.: s.n.], 1995. Citado na página 49.
- CORCHO, O.; FERNÁNDEZ-LÓPEZ, M.; GÓMEZ-PÉREZ, A. Ontologies for software engineering and software technology. In: \_\_\_\_\_. [S.l.]: Springer, 2006. cap. Ontological Engineering: Principles, Methods, Tools and Languages, p. 1–48. Citado 3 vezes nas páginas 58, 59 e 60.

- 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 5 vezes nas páginas 39, 40, 80, 81 e 132.
- CROWSTON, K. Open source software projects as virtual organizations: Competency rallying for software development. *IEE Proceedings – Software*, v. 149, n. 1, p. 3–17, 2002. Citado na página 74.
- DECKER, B. et al. Self-organized reuse of software engineering knowledge supported by semantic wikis. In: *Proc. First Workshop on Semantic Web Enabled Software Engineering*. [S.l.: s.n.], 2005. Citado na página 61.
- DETLOR, B. Information management. *International Journal of Information Management*, v. 30, p. 103–108, 2010. Citado na página 52.
- 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 na página 29.
- 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 29.
- DTF. *Developing and managing an information architecture*. 2. ed. Melbourne, 2010. Citado 3 vezes nas páginas 15, 37 e 38.
- DUARTE, J. *Uma Arquitetura Ágil da Informação Organizacional*. Tese (Doutorado) — Universidade de Brasília, Faculdade de Ciência da Informação, Brasília, 2011. Citado 3 vezes nas páginas 31, 41 e 48.
- DUDDUKURI, R.; PRABHAKAR, T. Helping architects in retrieving architecture documents: A semantic based approach. In: *1st International Workshop on Semantic Matchmaking and Resource Retrieval*. Seoul, Korea: [s.n.], 2006. Citado na página 60.
- ELLIS, D.; OLDRIDGE, R.; VASCONCELOS, A. Community and virtual community. *Annual Review of Information Science and Technology*, v. 38, n. 1, p. 145–186, 2004. Citado na página 72.
- EOP. *The Common Approach to Federal Enterprise Architecture*. [S.l.], 2012. Acessado em 28 de janeiro de 2014. Disponível em: <[http://www.whitehouse.gov/sites/default/files/omb/assets/egov\\_docs/common\\_approach\\_to\\_federal\\_ea.pdf](http://www.whitehouse.gov/sites/default/files/omb/assets/egov_docs/common_approach_to_federal_ea.pdf)>. Citado na página 34.
- 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: [s.n.], 2003. Citado na página 76.
- FINKELSTEIN, C.; AIKEN, P. *Building Corporate Portals with XML*. [S.l.]: McGraw-Hill, 2000. 512 p. Citado na página 54.

- FOX, M.; GRUNINGER, M. On ontologies and enterprise modeling. In: *Proceedings of the International Conference on Enterprise Integration Modelling Technology*. [S.l.]: Springer Verlag, 1997. Citado na página 59.
- FRASER, S. et al. Meeting the challenge of software engineering education for working professionals in the 21st century. In: *Panel in OOPSLA '03*. Anaheim, California: [s.n.], 2003. Citado na página 23.
- FREIRE, I. M. Barreiras na comunicação da informação tecnológica. *Ciência da Informação*, v. 20, n. 1, p. 51–54, 1991. Citado na página 23.
- FUJITA, M. A identificação de conceitos no processo de análise de assuntos para indexação. *Revista Digital de Biblioteconomia e Ciência da Informação*, v. 1, n. 1, p. 60–90, 2003. Citado na página 53.
- GAIL, H. *Systems of Knowledge Organization for Digital Libraries: Beyond Traditional Authority Files*. Washington, DC, 2000. Acessado em 28 de janeiro de 2014. Disponível em: <<http://www.clir.org/pubs/reports/pub91/pub91.pdf>>. Citado na página 57.
- GARLAN, D. Software architecture: a roadmap. In: *ICSE,00 Proceedings of the Conference on the Future of Software Engineering*. [S.l.: s.n.], 2000. p. 91–101. Citado na página 45.
- GARLAN, D.; KOMPANEK, A. Reconciling the needs of architectural description with object-modeling notations. *Science of Computer Programming Journal*, v. 44, n. 1, p. 23–49, 2002. Citado na página 49.
- GERMAN, D. Gnome: A case of open source global software development. In: *Proceedings of the International Workshop on Global Software Development*. Portland, OR, USA: [s.n.], 2003. Citado na página 76.
- GIL, A. *Como Elaborar Projetos de Pesquisa*. Quinta edição. São Paulo: Editora Atlas, 2010. 184 p. Citado na página 24.
- GILLAM, C.; OPPENHEIM, C. Review article: Reviewing the impact of virtual teams in the information age. *Journal of Information Science*, v. 32, n. 2, p. 160–175, 2006. Citado na página 71.
- 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 74.
- GREEFHORST, D.; KONING, H.; VLIET, H. The many faces of architectural descriptions. *Information Systems Frontiers*, v. 8, p. 103–113, 2006. Citado 2 vezes nas páginas 32 e 44.
- GREEFHORST, D.; PROPER, E. *Architecture Principles: The Cornerstones of Enterprise Architecture*. New York: Springer, 2011. 197 p. Citado 2 vezes nas páginas 28 e 47.
- GRUBER, T. A translation approach to portable ontology specifications. *Knowledge Acquisition*, v. 5, n. 1, p. 199–220, 1993. Citado na página 58.

- GUARINO, N. Formal ontology and information system. In: *Proceedings of FOIS'98*. Trento, Italy: IOS Press, 1998. Citado 2 vezes nas páginas 58 e 59.
- GUMM, D. Distribution dimensions in software development projects: A taxonomy. *IEEE Software*, v. 23, n. 5, p. 45–51, September/October 2006. Citado na página 71.
- GUTWIN, C.; PENNER, R.; SCHNEIDER, K. Group awareness in distributed software development. In: *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*. [S.l.]: ACM, 2004. p. 72–81. Citado na página 77.
- 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](http://argus-acia.com/white_papers/iaglossary.html)>. Citado na página 29.
- HALVORSON, K.; RACH, M. *Content Strategy for the Web*. 2nd. ed. Berkeley, CA: New Riders, 2012. 216 p. Citado 2 vezes nas páginas 29 e 37.
- HAMMOND, T. et al. Social bookmarking tools: A general review. *D-Lib Magazine*, v. 11, n. 4, April 2005. Citado na página 61.
- 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 61.
- HAYES, B. Cloud computing. *Communications of the ACM*, v. 51, n. 7, p. 9–11, July 2008. Citado na página 95.
- HEDDEN, H. *The Accidental Taxonomist*. Medford, N.J.: Information Today, Inc., 2010. 442 p. Citado 9 vezes nas páginas 55, 56, 57, 58, 93, 106, 107, 108 e 132.
- HELDMAN, K. *PMP Project Management Professional Exam Study Guide*. 6th. ed. Indianapolis, Ind.: Sybex, 2011. 598 p. Citado 3 vezes nas páginas 43, 93 e 125.
- HERBSLEB, J.; MOITRA, D. Guest editors' introduction: Global software development. *IEEE Software*, v. 18, n. 2, p. 16–20, 2001. Citado 2 vezes nas páginas 71 e 77.
- HERRING, S. Computer-mediated communication on the internet. *Annual Review of Information Science and Technology*, v. 36, p. 109–168, 2002. Citado na página 72.
- HOFMEISTER, C.; NORD, S.; SONI, D. *Applied Software Architecture*. Reading, Mass.: Addison Wesley, 1999. 397 p. Citado na página 47.
- IEEE. *IEEE std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology*. New York, NY, 1990. Citado na página 69.
- INGWERSEN, P. Information and information science in context. *Libri*, v. 42, n. 2, p. 99–135, 1992. Citado na página 19.
- ISACA. *COBIT 5: Process Reference Guide Exposure Draft*. 2011. Citado na página 64.
- ISO. *ISO 5963: Documentation – Methods for examining documents, determining their subjects, and selecting indexing terms*. Switzerland, 1985. Citado 2 vezes nas páginas 53 e 55.

- ISO. *ISO/IEC/IEEE 42010 Systems and software engineering – Architecture description*. Switzerland, 2011. Citado 11 vezes nas páginas 8, 28, 32, 43, 44, 45, 46, 47, 48, 49 e 82.
- JACKA, J.; KELLER, P. *Business Process Mapping: Improving Customer Satisfaction*. 2nd. ed. Hoboken, N.J.: Wiley, 2009. 336 p. Citado 3 vezes nas páginas 93, 107 e 132.
- JACOB, E.; LOEHRLEIN, A. Information architecture. *Annual Review of Information Science and Technology*, v. 43, n. 1, p. 1–64, 2009. Citado na página 29.
- JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. *The Unified Software Development Process*. Reading, Mass.: Addison Wesley, 1999. 463 p. Citado na página 70.
- 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 na página 69.
- KAZMAN, R. et al. Saam: a method for analyzing the properties of software architectures. In: *ICSE'94 Proceedings of the 16th international conference on software engineering*. [S.l.]: IEEE Computer Society Press, 1994. p. 81–90. Citado na página 47.
- KHOURY, G. *A Unified Approach to Enterprise Architecture Modeling*. Tese (Doutorado) — University of Technology, Faculty of Information Technology, Sidney, 2007. Citado na página 49.
- KRUCHTEN, P. Architectural blueprints - the “4+1” view model of software architecture. *IEEE Software*, v. 12, n. 6, p. 42–50, 1995. Citado na página 47.
- KULAK, D.; GUINEY, E. *Use Cases Requirements in Context*. 2nd. ed. Boston: Addison Wesley, 2003. Citado 3 vezes nas páginas 93, 103 e 132.
- LANGLANDS, M.; EDWARDS, C. Business vs. system use cases. 2009. Citado 3 vezes nas páginas 93, 103 e 132.
- LANKHORST, M. *Enterprise Architecture at Work: Modeling, Communication and Analysis*. 3rd. ed. [S.l.]: Springer, 2012. 364 p. (The Enterprise Engineering Series). Citado 10 vezes nas páginas 28, 32, 35, 43, 46, 47, 48, 49, 51 e 81.
- LARSON, E.; GRAY, C. *Project management: The managerial process*. 5th. ed. [S.l.]: McGraw Hill, 2010. 608 p. Citado na página 63.
- LAVE, J.; WENGER, E. *Situated Learning: Legitimate peripheral participation*. Cambridge, U.K.: Cambridge University Press, 1991. 138 p. Citado na página 76.
- LILES, D.; PRESLEY, A. Enterprise modeling within an enterprise engineering framework. In: *Proceedings of the 1996 Winter Simulation Conference*. San Diego, CA: [s.n.], 1996. Citado na página 65.
- 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 na página 30.
- LIPNACK, J.; STAMPS, J. *Virtual Teams: People Working Across Boundaries with Technology*. 2nd. ed. [S.l.]: Wiley, 2000. 352 p. Citado na página 72.

- LONCHAMP, J. Software process modeling. In: \_\_\_\_\_. [S.l.]: Springer, 2005. (International Series in Software Engineering), cap. Open Source Software Development Process Modeling, p. 208. Citado 3 vezes nas páginas 10, 99 e 100.
- LUNG, C.; JALNAPURKAR, A.; EL-RAYESS, A. Performance-oriented software architecture engineering: an experience report. In: *WOSP'98 Proceedings of the 1st international workshop on software and performance*. [S.l.]: ACM, 1998. p. 191–196. Citado na página 45.
- MAHMOOD, Z. Cloud computing for enterprise architectures. In: \_\_\_\_\_. [S.l.]: Springer, 2011. (Computer Communications and Networks), cap. Cloud Computing for Enterprise Architectures: Concepts, Principles and Approaches, p. 343. Citado na página 95.
- MALLOY, M. et al. *An Information Architecture Framework for the USAF: Managing Information from an Enterprise Architecture*. [S.l.], 2010. Citado 3 vezes nas páginas 19, 32 e 33.
- MARTIN, A.; DMITRIEV, D.; AKEROYD, J. A resurgence of interest in information architecture. *International Journal of Information Management*, v. 30, p. 6–12, 2010. Citado na página 52.
- MATHES, A. *Folksonomies: Cooperative Classification and Communication Through Shared Metadata*. [S.l.], 2004. Citado na página 61.
- MEDVIDOVIC, N. et al. Modeling software architecture in the unified modeling language. *ACM Transactions on Software Engineering and Methodology*, v. 11, n. 1, p. 2–57, January 2002. Citado na página 50.
- MELL, P.; GRANCE, T. *The NIST Definition of Cloud Computing*. 2011. Citado na página 94.
- MELO, A. *Um modelo de Arquitetura da Informação para processos de investigação científica*. Dissertação (Mestrado) — Universidade de Brasília, Faculdade de Ciência da Informação, 2010. Citado na página 24.
- 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 70.
- MISTRÍK, I. et al. *Collaborative Software Engineering*. [S.l.]: Springer, 2010. 480 p. Citado 2 vezes nas páginas 69 e 71.
- 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 76.
- MOODY, G. *Rebel code: Linux and the open source revolution*. New York: Allen Lane, 2001. Citado na página 76.
- 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 6 vezes nas páginas 15, 29, 35, 36, 55 e 57.

- MOSHIRI, S.; HILL, R. Cloud computing for enterprise architectures. In: \_\_\_\_\_. [S.l.]: Springer, 2011. cap. Enterprise Architecture Fundamentals, p. 343. Citado na página 33.
- 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 2 vezes nas páginas 75 e 76.
- NISO. *Understanding Metadata*. Bethesda, MD, USA, 2004. Citado 2 vezes nas páginas 53 e 54.
- NISO. *ANSI/NISO Z39.19-2005 Guidelines for the Construction, Format, and management of Monolingual Controlled Vocabularies*. Bethesda, MD, USA, 2005. Citado 10 vezes nas páginas 53, 55, 56, 57, 58, 93, 106, 107, 108 e 132.
- NISO. *A Framework of Guidance for Building Good Digital Collections*. 2007. Citado na página 54.
- NOVAK, J.; CAÑAS, A. *The Theory Underlying Concept Maps and How to Construct and Use Them*. Pensacola, Fl, 2008. Citado 2 vezes nas páginas 93 e 132.
- NOY, N.; MCGUINNESS, D. *Ontology Development 101: A Guide to Creating Your First Ontology*. [S.l.], 2001. Citado 4 vezes nas páginas 58, 59, 93 e 132.
- NOY, N.; MUSEN, M. Prompt: Algorithm and tool for automated ontology merging and alignment. In: *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*. [S.l.: s.n.], 2000. p. 450–455. Citado na página 60.
- OGC. *Managing Successful Projects with PRINCE2: 2009 Edition*. [S.l.]: Stationery Office Books, 2009. 342 p. Citado 5 vezes nas páginas 63, 64, 65, 66 e 67.
- OHLOH. *Ohloh, the open source network*. 2013. Acessado em 28 de janeiro de 2014. Disponível em: <<http://www.ohloh.net/>>. Citado 4 vezes nas páginas 8, 25, 95 e 96.
- OMG. *OMG Unified Modeling Language (OMG UML) Infrastructure, version 2.4.1*. 2011. Citado na página 50.
- OPENSTACK. *OpenStack Cloud Software*. 2013. Acessado em 28 de janeiro de 2014. Disponível em: <<http://www.openstack.org/>>. Citado 4 vezes nas páginas 8, 97, 101 e 106.
- PERRY, D.; STAUDENMEYER, N.; VOTTA, L. People, organizations, and process improvement. *IEEE Software*, v. 11, n. 4, p. 36–45, 1994. Citado 2 vezes nas páginas 23 e 70.
- PMI. *A Guide to the Project Management Body of Knowledge (PMBOK GUIDE)*. 5th. ed. [S.l.]: Project Management Institute, 2013. 589 p. Citado 10 vezes nas páginas 43, 44, 63, 64, 65, 66, 67, 80, 83 e 125.
- PREECE, J. Sociability and usability in online communities: Determining and measuring success. *Behavior and Information Technology*, v. 20, p. 347–356, 2001. Citado 2 vezes nas páginas 72 e 73.

- PRESSMAN, R. *Engenharia de Software: Uma Abordagem Profissional*. [S.l.]: McGraw-Hill, 2011. 776 p. Citado 3 vezes nas páginas 23, 69 e 70.
- Princeton University. *Guide to Creating Website Information Architecture and Content*. 2.2. ed. [S.l.], 2008. Acessado em 28 de janeiro de 2014. Disponível em: <<http://www.princeton.edu/communications/services/docs/IAguide2.pdf>>. Citado 2 vezes nas páginas 15 e 39.
- RAGAN, E.; FREZZA, S.; CANNELL, J. Product-based learning in software engineering education. In: *39th IEEE Frontiers in Education Conference*. San Antonio, USA: [s.n.], 2009. Citado na página 23.
- 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 76.
- RHEINGOLD, H. *The Virtual Community: Homesteading on the Electronic Frontier*. [S.l.]: The MIT Press, 2000. 447 p. Citado na página 72.
- ROBERTSON, J.; ROBERTSON, S. *Mastering the Requirements Process*. 2nd. ed. [S.l.]: Addison Wesley, 2006. 592 p. Citado 2 vezes nas páginas 93 e 125.
- ROCKLEY, A.; COOPER, C. *Managing Enterprise Content: A Unified Content Strategy*. 2nd. ed. [S.l.]: New Riders, 2012. Citado na página 37.
- ROSENBERG, J.; MATEOS, A. *The cloud at your service: The when, how and why of enterprise cloud computing*. Greenwich, CT: Manning Publications, 2011. 247 p. Citado na página 95.
- ROWLEY, J.; HARTLEY, R. *Organizing Knowledge: An Introduction to Managing Access to Information*. 4th. ed. [S.l.]: Ashgate Publishing Limited, 2008. Citado 6 vezes nas páginas 52, 55, 56, 59, 60 e 61.
- ROZANSKI, N.; WOODS, E. *Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives*. 2nd. ed. [S.l.]: Addison Wesley, 2011. Citado 8 vezes nas páginas 28, 43, 44, 45, 47, 48, 49 e 50.
- SAA. A glossary of archival and records terminology. Acessado em 28 de janeiro de 2014. 2014. Disponível em: <<http://www2.archivists.org/glossary>>. Citado na página 57.
- SARACEVIC, T. Information science. *Journal of the American Society for Information Science*, v. 50, n. 12, p. 1051–1063, 1999. Citado na página 19.
- SARMA, A. *A survey of collaborative tools in software development*. Irvine, 2005. Citado 4 vezes nas páginas 19, 22, 23 e 70.
- SCACCHI, W. et al. Understanding free/open source software development processes. *Software Process Improvement and Practice*, v. 11, p. 95–105, 2006. Citado 2 vezes nas páginas 74 e 75.
- SCHACH, S. *Engenharia de Software: Os Paradigmas Clássico e Orientado a Objetos*. Sétima. [S.l.]: McGraw Hill, 2008. 618 p. Citado 2 vezes nas páginas 23 e 70.

SEOCENTRO. *SEOCentro web marketing resources*. 2013. Acessado em 30 de julho de 2013. Disponível em: <<http://www.seocentro.com/>>. Citado na página 106.

SESSIONS, R. *A Comparison of the Top Four Enterprise Architecture Methodologies*. 2007. Acessado em 28 de janeiro de 2014. Disponível em: <<http://msdn.microsoft.com/en-us/library/bb466232.aspx>>. Citado na página 33.

SHAW, M. Abstractions for software architecture and tools to support them. *IEEE Transactions on Software Engineering*, v. 21, p. 314–335, April 1995. Citado na página 44.

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 23.

SHIN, H.; KIM, K. Examining identity and organizational citizenship behavior in computer-mediated communication. *Journal of Information Science*, v. 36, n. 1, p. 114–126, 2010. Citado na página 73.

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 na página 77.

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. Citado na página 30.

SMITH, B. Blackwell guide to the philosophy of computing and information. In: \_\_\_\_\_. Oxford: Blackwell, 2003. cap. Ontology, p. 155–166. Citado na página 58.

SOMMERVILLE, I. *Software Engineering*. 9th. ed. [S.l.]: Addison Wesley, 2010. Citado 2 vezes nas páginas 23 e 70.

SOUZA, G.; AKEN, E. V.; GROESBECK, R. Applying an enterprise engineering approach to engineering work: a focus on business process modeling. *Engineering Management Journal*, v. 14, n. 3, p. 15–24, 2002. Citado na página 48.

SOWA, J.; ZACHMAN, J. Extending and formalizing the framework for information systems architecture. *IBM Systems Journal*, v. 31, n. 3, p. 590–616, 1992. Citado na página 33.

SPENCER, D.; FEATHERSTONE, D. *A Practical Guide to Information Architecture*. Penarth, UK: Five Simple Steps, 2010. 310 p. Citado 5 vezes nas páginas 15, 29, 36, 37 e 55.

STARUML. *StarUML The Open Source UML/MDA Platform*. 2013. Acessado em 14 de agosto de 201. Disponível em: <<http://staruml.sourceforge.net/en/index.php>>. Citado na página 104.

STEEN, M. et al. Supporting viewpoint-oriented enterprise architecture. In: *Enterprise Distributed Object Computing Conference (EDOC 2004)*. [S.l.: s.n.], 2004. p. 201–211. Citado na página 47.

STUDER, R.; BENJAMINS, V.; FENSEL, D. Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, v. 25, p. 161–197, 1998. Citado na página 59.

- SVENONIUS, E. *The Intellectual Foundation of Information Organization*. [S.l.]: The MIT Press, 2000. 280 p. Citado na página 61.
- TAYLOR, A. G.; JOUNDREY, D. N. *The Organization of Information*. 3rd. ed. West Port, Conn.: Libraries Unlimited, 2009. Citado 2 vezes nas páginas 29 e 54.
- 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](http://iainstitute.org/en/learn-/resources/what_is_ia.php)>. Citado na página 29.
- The Open Group. *TOGAF Version 9.1*. [S.l.]: Van Haren Publishing, 2011. 654 p. Citado 13 vezes nas páginas 28, 32, 34, 41, 43, 46, 48, 80, 81, 82, 83, 125 e 132.
- The Open Group. *ArchiMate 2.0 Specification*. [S.l.]: Van Haren Publishing, 2012. 216 p. Citado 5 vezes nas páginas 35, 47, 50, 51 e 108.
- The Open Group. *ArchiMate 2.0 Translation Glossary: English – Brazilian Portuguese*. 2013. Citado 3 vezes nas páginas 28, 35 e 48.
- 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 29.
- USCHOLD, M. Knowledge level modeling: concepts and terminology. *The Knowledge Engineering Review*, v. 13, n. 1, p. 5–29, 1998. Citado 2 vezes nas páginas 58 e 59.
- WANG, L. et al. Cloud computing a perspective study. *New Generation Computing*, v. 29, n. 2, p. 137–146, 2010. Citado na página 95.
- 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 na página 56.
- WENGER, E.; LAVE, J. *Situated Learning: Legitimate peripheral participation*. Cambridge, U.K.: Cambridge University Press, 1991. 138 p. Citado na página 73.
- WG42. *Survey of Architecture Frameworks*. 2013. Acessado em 28 de janeiro de 2014. Disponível em: <<http://www.iso-architecture.org/ieee-1471/afs/frameworks-table.html>>. Citado 2 vezes nas páginas 32 e 33.
- WHITEHEAD, J. Collaboration in software engineering: A roadmap. In: *Proceedings of FOSE'07 Future of Software Engineering*. [S.l.: s.n.], 2007. p. 214–215. Citado na página 69.
- WONGTHONGTHAM, P. et al. Ontology-based multi-site software methodology and tools. *Journal of Systems Architecture*, v. 52, p. 640–653, 2006. Citado 2 vezes nas páginas 61 e 71.
- WU, Y. et al. Construction of ontology-based software repositories by text mining. *Lecture Notes in Computer Science*, v. 4489, p. 790–797, 2007. Citado na página 77.
- YIN, R. K. *Case study research: Design and Methods*. 2nd. ed. Thousand Oaks: SAGE Publications, 1994. Citado na página 24.

ZACHMAN, J. A framework for information systems architecture. *IBM Systems Journal*, v. 26, n. 3, p. 276–292, 1987. Citado 2 vezes nas páginas 33 e 47.

ZACHMAN, J. *John Zachman's Concise Definition of the Zachman Framework*. 2008. Acessado em 28 de janeiro de 2014. Disponível em: <<http://www.zachman.com/about-the-zachman-framework>>. Citado na página 33.

ZENUN, M.; LOUREIRO, G.; ARAUJO, C. The effects of teams' co-location on project performance. In: LOUREIRO G. & CURRAN, R. (Ed.). *Complex Systems Concurrent Engineering-Collaboration, Technology Innovation and Sustainability*. London: Springer, 2007. p. 717–726. Citado na página 71.

ZHAO, Y.; DONG, J. Ontology classification for semantic-web based software engineering. *IEEE Transactions on Services Computing*, v. 2, n. 4, p. 303–317, 2009. Citado na página 61.

# Apêndices

# APÊNDICE A – Roteiros para processos de gerenciamento

Neste apêndice são descritos roteiros para processos de gerenciamento do modelo de *framework* proposto. Esses roteiros foram desenvolvidos por meio de consultas às seguintes fontes: Heldman (2011), PMI (2013), The Open Group (2011), Robertson e Robertson (2006) e documentos referenciados nos próprios roteiros.

|           |  |
|-----------|--|
| ROTEIRO   | Desenvolver termo de abertura do projeto   |
| PROCESSO  | Desenvolver termo de abertura do projeto   |
| DESCRIÇÃO | <p>Definir título do documento.</p> <p>Definir número de versão do documento.</p> <p>Definir data de elaboração do documento.</p> <p>Descrever objetivos do documento.</p> <p>Manter atualizado o histórico de alterações do documento.</p> <p>Descrever resumidamente o projeto.</p> <p>Descrever objetivos do projeto.</p> <p>Relacionar problemas que justificam o projeto.</p> <p>Relacionar oportunidades que justificam o projeto.</p> <p>Descrever escopo do projeto.</p> <p>Relacionar requisitos excluídos do escopo do projeto.</p> <p>Relacionar partes interessadas (<i>stakeholders</i>) no projeto.</p> <p>Informar nome do responsável pelo projeto.</p> <p>Relacionar papéis dos membros do projeto.</p> <p>Relacionar responsabilidades dos membros do projeto.</p> <p>Relacionar marcos do projeto.</p> <p>Relacionar potenciais impactos decorrentes do projeto.</p> <p>Relacionar unidades impactadas pelo projeto.</p> <p>Estimar recursos necessários ao projeto.</p> <p>Relacionar premissas associadas ao projeto.</p> <p>Relacionar restrições associadas ao projeto.</p> <p>Relacionar riscos associados ao projeto.</p> <p>Relacionar medidas de sucesso do projeto.</p> <p>Descrever critérios de aceitação do projeto.</p> <p>Descrever domínios de arquitetura abordados pelo projeto.</p> <p>Relacionar pontos de vista de arquitetura abordados pelo projeto.</p> <p>Referenciar fontes de informação consultadas.</p> |
| ARTEFATOS | Termo de abertura do projeto   |
| FONTES    | <p>Heldman (2011).</p> <p>PMBOK (PMI, 2013).</p> <p>Project Managment Docs (Em: &lt;<a href="http://www.projectmanagementdocs.com/">http://www.projectmanagementdocs.com/</a>&gt;. Acessado em 30 de maio de 2013).</p>  |

Figura 22 – Roteiro Desenvolver termo de abertura do projeto.

|           |   |
|-----------|---|
| ROTEIRO   | Desenvolver registro das partes interessadas  |
| PROCESSO  | Identificar partes interessadas   |
| DESCRIÇÃO | Definir título do documento.<br>Definir número de versão do documento.<br>Definir data de elaboração do documento.<br>Descrever objetivos do documento.<br>Manter atualizado o histórico de alterações do documento.<br>Relacionar identificadores das partes interessadas ( <i>stakeholders</i> ) no projeto.<br>Descrever expectativas das partes interessadas no projeto.<br>Descrever interesses das partes interessadas no projeto.<br>Classificar as partes interessadas no projeto.<br>Relacionar artefatos relevantes a cada parte interessada no projeto.<br>Referenciar fontes de informação consultadas. |
| ARTEFATOS | Registro das partes interessadas  |
| FONTES    | PMBOK (PMI, 2013).<br>TOGAF (THE OPEN GROUP, 2011).   |

Figura 23 – Roteiro Desenvolver registro das partes interessadas.

|           |  |
|-----------|--|
| ROTEIRO   | Desenvolver plano de gerenciamento do projeto  |
| PROCESSO  | Desenvolver plano de gerenciamento do projeto  |
| DESCRIÇÃO | <p>Definir título do documento.</p> <p>Definir número de versão do documento.</p> <p>Definir data de elaboração do documento.</p> <p>Descrever objetivos do documento.</p> <p>Manter atualizado o histórico de alterações do documento.</p> <p>Descrever resumidamente o projeto.</p> <p>Relacionar objetivos do projeto.</p> <p>Descrever escopo do projeto.</p> <p>Relacionar papéis dos membros do projeto.</p> <p>Relacionar responsabilidades dos membros do projeto.</p> <p>Relacionar marcos do projeto.</p> <p>Relacionar recursos necessários ao projeto.</p> <p>Descrever ciclo de vida do projeto.</p> <p>Relacionar processos empregados no projeto.</p> <p>Relacionar artefatos resultantes dos processos no projeto.</p> <p>Relacionar ferramentas empregadas no projeto.</p> <p>Relacionar métodos empregados no projeto.</p> <p>Relacionar premissas relevantes ao projeto.</p> <p>Relacionar restrições relevantes ao projeto.</p> <p>Referenciar fontes de informação consultadas.</p> |
| ARTEFATOS | Plano de gerenciamento do projeto  |
| FONTES    | <p>Heldman (2011).</p> <p>PMBOK (PMI, 2013).</p> <p>Project Managment Docs (Em: &lt;<a href="http://www.projectmanagementdocs.com/project-planning-templates/project-management-plan.html">http://www.projectmanagementdocs.com/project-planning-templates/project-management-plan.html</a>&gt;. Acessado em 30 de maio de 2013).</p> <p>Project Managment Docs (Em: &lt;<a href="http://www.projectmanagementdocs.com/project-planning-templates/scope-management-plan.html">http://www.projectmanagementdocs.com/project-planning-templates/scope-management-plan.html</a>&gt;. Acessado em 30 de maio de 2013).</p> <p>Project Managment Docs (Em: &lt;<a href="http://www.projectmanagementdocs.com/project-planning-templates/requirements-management-plan.html">http://www.projectmanagementdocs.com/project-planning-templates/requirements-management-plan.html</a>&gt;. Acessado em 30 de maio de 2013).</p>  |

Figura 24 – Roteiro Desenvolver plano de gerenciamento do projeto.

|           |  |
|-----------|--|
| ROTEIRO   | Desenvolver documentação dos requisitos  |
| PROCESSO  | Coletar requisitos   |
| DESCRIÇÃO | <p>Definir título do documento.</p> <p>Definir número de versão do documento.</p> <p>Definir data de elaboração do documento.</p> <p>Descrever objetivos do documento.</p> <p>Manter atualizado o histórico de alterações do documento.</p> <p>Relacionar identificadores para os requisitos do projeto.</p> <p>Relacionar nomes dos requisitos do projeto.</p> <p>Descrever requisitos do projeto.</p> <p>Relacionar motivações para a existência dos requisitos do projeto.</p> <p>Relacionar prioridades associadas aos requisitos do projeto.</p> <p>Relacionar critérios de seleção dos requisitos do projeto.</p> <p>Relacionar identificadores das origens das solicitações dos requisitos.</p> <p>Relacionar estados dos requisitos (aprovado, etc.) do projeto.</p> <p>Descrever relações existentes entre os requisitos do projeto.</p> <p>Referenciar fontes de informação consultadas.</p> |
| ARTEFATOS | Documentação dos requisitos  |
| FONTES    | Robertson & Robertson (2006).<br>PMBOK (PMI, 2013).  |

Figura 25 – Roteiro Desenvolver documentação dos requisitos.

|           |   |
|-----------|---|
| ROTEIRO   | Desenvolver enunciado do escopo do projeto  |
| PROCESSO  | Definir escopo  |
| DESCRIÇÃO | <p>Definir título do documento.</p> <p>Definir número de versão do documento.</p> <p>Definir data de elaboração do documento.</p> <p>Descrever objetivos do documento.</p> <p>Manter atualizado o histórico de alterações do documento.</p> <p>Descrever resumidamente o projeto.</p> <p>Relacionar partes interessadas (<i>stakeholders</i>) no projeto.</p> <p>Descrever escopo do projeto.</p> <p>Relacionar requisitos fora do escopo do projeto.</p> <p>Relacionar entregas do projeto.</p> <p>Relacionar critérios de aceitação das entregas.</p> <p>Relacionar fronteiras do projeto.</p> <p>Relacionar premissas e associadas ao projeto.</p> <p>Relacionar restrições associadas ao projeto.</p> <p>Referenciar fontes de informação consultadas</p> |
| ARTEFATOS | Enunciado do escopo do projeto  |
| FONTES    | Heldman (2011).<br>PMBOK (PMI, 2013).   |

Figura 26 – Roteiro Desenvolver enunciado do escopo do projeto.

|           |  |
|-----------|--|
| ROTEIRO   | Desenvolver lista de marcos  |
| PROCESSO  | Definir atividades   |
| DESCRIÇÃO | Definir título do documento.<br>Definir número de versão do documento.<br>Definir data de elaboração do documento.<br>Descrever objetivos do documento.<br>Manter atualizado o histórico de alterações do documento.<br>Relacionar identificadores dos marcos do projeto.<br>Relacionar nomes dos marcos.<br>Para cada marco do projeto, informar se o marco é obrigatório ou opcional.<br>Relacionar datas dos marcos do projeto.<br>Descrever como verificar marcos do projeto.<br>Relacionar entregas que caracterizam cada marco do projeto.<br>Referenciar fontes de informação consultadas |
| ARTEFATOS | Lista de marcos  |
| FONTES    | Heldman (2011).<br>PMBOK (PMI, 2013).<br>Project Managment Docs (Em: < <a href="http://www.projectmanagementdocs.com/project-documents/milestone-list.html">http://www.projectmanagementdocs.com/project-documents/milestone-list.html</a> >. Acessado em 30 de maio de 2013).   |

Figura 27 – Roteiro Desenvolver lista de marcos.

|           |  |
|-----------|--|
| ROTEIRO   | Desenvolver lista de atividades  |
| PROCESSO  | Definir atividades   |
| DESCRIÇÃO | <p>Definir título do documento.</p> <p>Definir número de versão do documento.</p> <p>Definir data de elaboração do documento.</p> <p>Descrever objetivos do documento.</p> <p>Manter atualizado o histórico de alterações do documento.</p> <p>Relacionar identificadores das atividades no projeto.</p> <p>Relacionar nomes das atividades no projeto.</p> <p>Relacionar responsáveis pelas atividades no projeto.</p> <p>Descrever atividades no projeto.</p> <p>Relacionar recursos necessários às atividades no projeto.</p> <p>Estabelecer relacionamentos entre atividades antecessoras e sucessoras.</p> <p>Relacionar localizações das atividades no projeto.</p> <p>Relacionar premissas associadas às atividades no projeto.</p> <p>Relacionar restrições associadas às atividades no projeto.</p> <p>Descrever os escopos das atividades no projeto.</p> <p>Referenciar fontes de informação consultadas.</p> |
| ARTEFATOS | Lista de atividades  |
| FONTES    | <p>Heldman (2011).</p> <p>PMBOK (PMI, 2013).</p> <p>Project Managment Docs (Em: &lt;<a href="http://www.projectmanagementdocs.com/project-documents/activity-list.html">http://www.projectmanagementdocs.com/project-documents/activity-list.html</a>. Acessado em 30 de maio de 2013).</p> <p>Project Managment Docs (Em: &lt;<a href="http://www.projectmanagementdocs.com/project-documents/activity-attributes.html">http://www.projectmanagementdocs.com/project-documents/activity-attributes.html</a>. Acessado em 30 de maio de 2013).</p>   |

Figura 28 – Roteiro Desenvolver lista de atividades.

|           |  |
|-----------|--|
| ROTEIRO   | Desenvolver diagrama de sequência de atividades  |
| PROCESSO  | Sequenciar atividades  |
| DESCRIÇÃO | <p>Definir título do documento.</p> <p>Definir número de versão do documento.</p> <p>Definir data de elaboração do documento.</p> <p>Descrever objetivos do documento.</p> <p>Manter atualizado o histórico de alterações do documento.</p> <p>Identificar dependências entre as atividades no projeto.</p> <p>Construir diagrama representando as dependências entre as atividades no projeto.</p> <p>Referenciar fontes de informação consultadas.</p> |
| ARTEFATOS | Diagrama de sequência de atividades  |
| FONTES    | Heldman (2011).  |

Figura 29 – Roteiro Desenvolver diagrama de sequência de atividades.

|           |  |
|-----------|--|
| ROTEIRO   | Desenvolver cronograma   |
| PROCESSO  | Desenvolver cronograma   |
| DESCRIÇÃO | Definir título do documento.<br>Definir número de versão do documento.<br>Definir data de elaboração do documento.<br>Descrever objetivos do documento.<br>Manter atualizado o histórico de alterações do documento.<br>Relacionar identificadores das atividades no projeto.<br>Relacionar nomes das atividades no projeto.<br>Relacionar os identificadores dos responsáveis pelas atividades no projeto.<br>Relacionar as durações das atividades no projeto.<br>Relacionar as datas de início de cada atividade no projeto.<br>Relacionar as datas de término de cada atividade no projeto.<br>Referenciar fontes de informação consultadas. |
| ARTEFATOS | Cronograma   |
| FONTES    | Heldman (2011).<br>PMBOK (PMI, 2013).  |

Figura 30 – Roteiro Desenvolver cronograma.

## APÊNDICE B – Roteiros para processos de arquitetura

Neste apêndice são descritos roteiros para processos de arquitetura do modelo de *framework* proposto. Esses roteiros foram desenvolvidos por meio de consultas às seguintes fontes: Jacka e Keller (2009), Kulak e Guiney (2003), Langlands e Edwards (2009), NISO (2005), Hedden (2010), Noy e McGuinness (2001), Costa (2009), Novak e Cañas (2008), Cañas et al. (2004), The Open Group (2011) e documentos referenciados nos próprios roteiros.

|           |  |
|-----------|--|
| ROTEIRO   | Descrever princípios de arquitetura  |
| PROCESSO  | Identificar e estabelecer princípios de arquitetura  |
| DESCRIÇÃO | Definir título do documento.<br>Definir número de versão do documento.<br>Definir data de elaboração do documento.<br>Descrever objetivos do documento.<br>Manter atualizado o histórico de alterações do documento.<br>Descrever princípios associados ao negócio.<br>Referenciar fontes de informação consultadas. |
| ARTEFATOS | Catálogo de princípios de arquitetura  |
| FONTES    | The Open Group. (Em: < <a href="http://www.togaf.info/togaf9/togafSlides9/TOGAF-V9-Sample-Catalogs-Matrices-Diagrams-v2.pdf">http://www.togaf.info/togaf9/togafSlides9/TOGAF-V9-Sample-Catalogs-Matrices-Diagrams-v2.pdf</a> >. Acessado em 31 de maio de 2013).   |

Figura 31 – Roteiro Descrever princípios de arquitetura.

|           |   |
|-----------|---|
| ROTEIRO   | Descrever configuração do <i>framework</i> de arquitetura   |
| PROCESSO  | Configurar o TOGAF e outros <i>frameworks</i> de arquitetura selecionados   |
| DESCRIÇÃO | <p>Definir título do documento.</p> <p>Definir número de versão do documento.</p> <p>Definir data de elaboração do documento.</p> <p>Descrever objetivos do documento.</p> <p>Manter atualizado o histórico de alterações do documento.</p> <p>Relacionar <i>frameworks</i> e métodos a serem configurados e integrados.</p> <p>Descrever domínio de arquitetura adotado.</p> <p>Descrever ponto de vista de arquitetura adotado.</p> <p>Descrever modelo de ciclo de vida adotado para o projeto.</p> <p>Descrever fases no modelo de ciclo de vida adotado para o projeto.</p> <p>Descrever processos integrantes do <i>framework</i>.</p> <p>Descrever artefatos integrantes do <i>framework</i>.</p> <p>Referenciar fontes de informação consultadas.</p> |
| ARTEFATOS | Framework de arquitetura configurado  |
| FONTES    | TOGAF (THE OPEN GROUP, 2011).   |

Figura 32 – Roteiro Descrever configuração do *framework* de arquitetura.

|           |   |
|-----------|---|
| ROTEIRO   | Descrever atores e papéis   |
| PROCESSO  | Desenvolver descrição da arquitetura atual ( <i>baseline</i> )  |
| DESCRIÇÃO | <p>Definir título do documento.</p> <p>Definir número de versão do documento.</p> <p>Definir data de elaboração do documento.</p> <p>Descrever objetivos do documento.</p> <p>Manter atualizado o histórico de alterações do documento.</p> <p>Relacionar identificadores dos atores.</p> <p>Relacionar nomes dos atores.</p> <p>Descrever os atores.</p> <p>Relacionar categorias dos atores.</p> <p>Relacionar localizações dos atores.</p> <p>Relacionar identificadores dos papéis.</p> <p>Relacionar nomes dos papéis.</p> <p>Descrever papéis.</p> <p>Descrever relacionamentos entre os atores e os papéis assumidos.</p> <p>Referenciar fontes de informação consultadas.</p> |
| ARTEFATOS | <p>Catálogo organização/ator</p> <p>Catálogo de papéis</p> <p>Matriz ator/papel</p>   |
| FONTES    | The Open Group (Em: < <a href="http://www.togaf.info/togaf9/togafSlides9/TOGAF-V9-Sample-Catalogs-Matrices-Diagrams-v2.pdf">http://www.togaf.info/togaf9/togafSlides9/TOGAF-V9-Sample-Catalogs-Matrices-Diagrams-v2.pdf</a> >. Acessado em 31 de maio de 2013).   |

Figura 33 – Roteiro Descrever atores e papéis.

|           |   |
|-----------|---|
| ROTEIRO   | Descrever serviços e funções de negócio   |
| PROCESSO  | Desenvolver descrição da arquitetura atual ( <i>baseline</i> )  |
| DESCRIÇÃO | <p>Definir título do documento.</p> <p>Definir número de versão do documento.</p> <p>Definir data de elaboração do documento.</p> <p>Descrever objetivos do documento.</p> <p>Manter atualizado o histórico de alterações do documento.</p> <p>Identificar as funções do negócio.</p> <p>Atribuir identificadores e nomes às funções do negócio.</p> <p>Descrever funções de negócio.</p> <p>Identificar os serviços do negócio.</p> <p>Atribuir identificadores e nomes aos serviços do negócio.</p> <p>Descrever serviços do negócio.</p> <p>Identificar informações necessárias aos serviços.</p> <p>Descrever informações necessárias aos serviços.</p> <p>Representar funções do negócio.</p> <p>Representar serviços do negócio.</p> <p>Representar relacionamentos entre serviços.</p> <p>Representar informações necessárias à prestação dos serviços.</p> <p>Referenciar fontes de informação consultadas.</p> |
| ARTEFATOS | <p>Catálogo de serviço/função de negócio</p> <p>Diagrama de decomposição funcional</p> <p>Diagrama serviço/informação de negócio</p>  |
| FONTES    | <p>The Open Group (Em: &lt;<a href="http://www.togaf.info/togaf9/togafSlides9/TOGAF-V9-Sample-Catalogs-Matrices-Diagrams-v2.pdf">http://www.togaf.info/togaf9/togafSlides9/TOGAF-V9-Sample-Catalogs-Matrices-Diagrams-v2.pdf</a>&gt;. Acessado em 26 de junho de 2013).</p> <p>Togaf-Modeling.org (Em: &lt;<a href="http://www.togaf-modeling.org/models/business-architecture-menu/functional-decomposition-diagrams-menu.html">http://www.togaf-modeling.org/models/business-architecture-menu/functional-decomposition-diagrams-menu.html</a>&gt;. Acessado em 25 de junho de 2013).</p>   |

Figura 34 – Roteiro Descrever serviços e funções de negócio.

|           |  |
|-----------|--|
| ROTEIRO   | Descrever localizações   |
| PROCESSO  | Desenvolver descrição da arquitetura atual ( <i>baseline</i> )   |
| DESCRIÇÃO | <p>Definir título do documento.</p> <p>Definir número de versão do documento.</p> <p>Definir data de elaboração do documento.</p> <p>Descrever objetivos do documento.</p> <p>Manter atualizado o histórico de alterações do documento.</p> <p>Relacionar identificadores das localizações da empresa.</p> <p>Relacionar nomes das localizações.</p> <p>Descrever localizações.</p> <p>Referenciar fontes de informação consultadas.</p> |
| ARTEFATOS | Catálogo de localização  |
| FONTES    | The Open Group (Em: < <a href="http://www.togaf.info/togaf9/togafSlides9/TOGAF-V9-Sample-Catalogs-Matrices-Diagrams-v2.pdf">http://www.togaf.info/togaf9/togafSlides9/TOGAF-V9-Sample-Catalogs-Matrices-Diagrams-v2.pdf</a> >. Acessado em 31 de maio de 2013).  |

Figura 35 – Roteiro Descrever localizações.

|           |   |
|-----------|---|
| ROTEIRO   | Desenvolver modelo de casos de uso de negócio   |
| PROCESSO  | Desenvolver descrição da arquitetura atual ( <i>baseline</i> )  |
| DESCRIÇÃO | <p>Definir título do documento.</p> <p>Definir número de versão do documento.</p> <p>Definir data de elaboração do documento.</p> <p>Descrever objetivos do documento.</p> <p>Manter atualizado o histórico de alterações do documento.</p> <p>Identificar e acessar fontes de informação (usuários, <i>stakeholders</i>, documentos, etc.).</p> <p>Criar catálogo de casos de uso de negócio.</p> <p>Identificar atores.</p> <p>Atribuir nomes aos atores.</p> <p>Descrever atores.</p> <p>Identificar casos de uso.</p> <p>Atribuir nomes aos casos de uso.</p> <p>Descrever resumidamente casos de uso.</p> <p>Refinar nomes de atores e de casos de uso.</p> <p>Identificar e descrever fluxo principal de cada caso de uso.</p> <p>Identificar e descrever fluxos alternativos que existam nos casos de uso.</p> <p>Identificar e descrever fluxos de exceção que existam nos casos de uso.</p> <p>Identificar e descrever gatilhos (<i>trigger</i>) dos casos de uso.</p> <p>Relacionar pré-condições dos casos de uso.</p> <p>Relacionar pós-condições dos casos de uso.</p> <p>Identificar relacionamentos entre atores (herança, etc.).</p> <p>Identificar relacionamentos entre casos de uso (inclusão, extensão, etc.).</p> <p>Identificar pontos de extensão existentes nas documentações dos casos de uso.</p> <p>Criar diagramas de casos de uso de negócio.</p> <p>Representar atores.</p> <p>Representar casos de uso.</p> <p>Representar relacionamentos entre atores.</p> <p>Representar relacionamentos entre casos de uso.</p> <p>Representar relacionamentos entre atores e casos de uso.</p> <p>Testar casos de uso com cenários.</p> <p>Revisar catálogo de casos de uso de negócio.</p> <p>Revisar diagramas de casos de uso de negócio.</p> <p>Referenciar fontes de informação consultadas.</p> |
| ARTEFATOS | <p>Catálogo de casos de uso de negócio</p> <p>Diagrama de casos de uso de negócio</p>   |
| FONTES    | <p>Kulak e Guiney (2003).</p> <p>Langlands e Edwards (2009).</p> <p>The Open Group (Em: &lt;<a href="http://www.togaf.info/togaf9/togafSlides9/TOGAF-V9-Sample-Catalogs-Matrices-Diagrams-v2.pdf">http://www.togaf.info/togaf9/togafSlides9/TOGAF-V9-Sample-Catalogs-Matrices-Diagrams-v2.pdf</a>&gt;. Acessado em 26 de junho de 2013).</p>  |

Figura 36 – Roteiro Desenvolver modelo de casos de uso de negócio.

|           |   |
|-----------|---|
| ROTEIRO   | Descrever <i>business footprint</i>   |
| PROCESSO  | Desenvolver descrição da arquitetura atual ( <i>baseline</i> )  |
| DESCRIÇÃO | <p>Definir título do documento.</p> <p>Definir número de versão do documento.</p> <p>Definir data de elaboração do documento.</p> <p>Descrever objetivos do documento.</p> <p>Manter atualizado o histórico de alterações do documento.</p> <p>Representar metas do negócio.</p> <p>Representar unidades do negócio.</p> <p>Representar funções do negócio.</p> <p>Representar serviços do negócio.</p> <p>Descrever relacionamentos entre os elementos no diagrama.</p> <p>Referenciar fontes de informação consultadas.</p>   |
| ARTEFATOS | Diagrama <i>business footprint</i>  |
| FONTES    | <p>The Open Group (Em: &lt;<a href="http://www.togaf.info/togaf9/togafSlides9/TOGAF-V9-Sample-Catalogs-Matrices-Diagrams-v2.pdf">http://www.togaf.info/togaf9/togafSlides9/TOGAF-V9-Sample-Catalogs-Matrices-Diagrams-v2.pdf</a>&gt;. Acessado em 26 de junho de 2013).</p> <p>Togaf-Modeling.org (Em: &lt;<a href="http://www.togaf-modeling.org/models/business-architecture-menu/business-footprint-diagrams-menu.html">http://www.togaf-modeling.org/models/business-architecture-menu/business-footprint-diagrams-menu.html</a>&gt;. Acessado em 25 de junho de 2013).</p> |

Figura 37 – Roteiro Descrever *business footprint*.

|           |   |
|-----------|---|
| ROTEIRO   | Descrever ciclo de vida do produto  |
| PROCESSO  | Desenvolver descrição da arquitetura atual ( <i>baseline</i> )  |
| DESCRIÇÃO | <p>Definir título do documento.</p> <p>Definir número de versão do documento.</p> <p>Definir data de elaboração do documento.</p> <p>Descrever objetivos do documento.</p> <p>Manter atualizado o histórico de alterações do documento.</p> <p>Descrever fases no ciclo de vida do produto.</p> <p>Informar vendas do produto em cada fase.</p> <p>Informar lucros das vendas do produto em cada fase.</p> <p>Referenciar fontes de informação consultadas.</p>                                 |
| ARTEFATOS | Diagrama de ciclo de vida do produto  |
| FONTES    | <p>MarketingTeacher.com (Em: &lt;<a href="http://www.marketingteacher.com/lesson-store/lesson-plc.html">http://www.marketingteacher.com/lesson-store/lesson-plc.html</a>&gt;. Acessado em 26 de junho de 2013).</p> <p>The Open Group (Em: &lt;<a href="http://www.togaf.info/togaf9/togafSlides9/TOGAF-V9-Sample-Catalogs-Matrices-Diagrams-v2.pdf">http://www.togaf.info/togaf9/togafSlides9/TOGAF-V9-Sample-Catalogs-Matrices-Diagrams-v2.pdf</a>&gt;. Acessado em 26 de junho de 2013).</p> |

Figura 38 – Roteiro Descrever ciclo de vida do produto.

|           |   |
|-----------|---|
| ROTEIRO   | Descrever composição da organização   |
| PROCESSO  | Desenvolver descrição da arquitetura atual ( <i>baseline</i> )  |
| DESCRIÇÃO | <p>Definir título do documento.</p> <p>Definir número de versão do documento.</p> <p>Definir data de elaboração do documento.</p> <p>Descrever objetivos do documento.</p> <p>Manter atualizado o histórico de alterações do documento.</p> <p>Representar os atores.</p> <p>Construir representações dos papéis.</p> <p>Representar localizações.</p> <p>Representar unidades da organização.</p> <p>Representar relacionamentos entre localizações dos elementos na organização.</p> <p>Referenciar fontes de informação consultadas</p>  |
| ARTEFATOS | Diagrama de decomposição da organização   |
| FONTES    | <p>The Open Group (Em: &lt;<a href="http://www.togaf.info/togaf9/togafSlides9/TOGAF-V9-Sample-Catalogs-Matrices-Diagrams-v2.pdf">http://www.togaf.info/togaf9/togafSlides9/TOGAF-V9-Sample-Catalogs-Matrices-Diagrams-v2.pdf</a>&gt;. Acessado em 26 de junho de 2013).</p> <p>Togaf-Modeling.org (Em: &lt;<a href="http://www.togaf-modeling.org/models/business-architecture-menu/organization-decomposition-diagrams-menu.html">http://www.togaf-modeling.org/models/business-architecture-menu/organization-decomposition-diagrams-menu.html</a>&gt;. Acessado em 25 de junho de 2013).</p> |

Figura 39 – Roteiro Descrever composição da organização.

|           |  |
|-----------|--|
| ROTEIRO   | Descrever processos do negócio   |
| PROCESSO  | Desenvolver descrição da arquitetura atual ( <i>baseline</i> )   |
| DESCRIÇÃO | <p>Definir título do documento.</p> <p>Definir número de versão do documento.</p> <p>Definir data de elaboração do documento.</p> <p>Descrever objetivos do documento.</p> <p>Manter atualizado o histórico de alterações do documento.</p> <p>Coletar informações necessárias à identificação dos processos (relatórios, entrevistas, etc.).</p> <p>Analisar informações coletadas.</p> <p>Identificar eventos que desencadeiam (<i>trigger</i>) as execuções de processos.</p> <p>Identificar eventos que caracterizam os términos de processos.</p> <p>Atribuir nome e identificador a cada evento.</p> <p>Descrever eventos.</p> <p>Identificar processos.</p> <p>Atribuir nome e identificador a cada processo.</p> <p>Descrever processos.</p> <p>Identificar relacionamentos entre os processos.</p> <p>Identificar entradas e saídas associadas a cada processo.</p> <p>Atribuir nome e identificador a cada produto de processo.</p> <p>Descrever produtos de processos.</p> <p>Decompor processos em atividades.</p> <p>Identificar relacionamentos entre atividades.</p> <p>Identificar e entrevistar donos dos processos (pessoas, unidades na organização, etc.).</p> <p>Revisar descrições dos processos.</p> <p>Escolher modo de representação (diagramas de processos, mapas de processos, etc.).</p> <p>Representar eventos, processos, produtos, atividades e unidades da organização.</p> <p>Representar responsáveis pelas atividades.</p> <p>Representar relacionamentos entre eventos e processos.</p> <p>Representar relacionamentos entre atividades.</p> <p>Representar relacionamentos entre eventos e atividades.</p> <p>Representar relacionamentos entre unidades da organização e processos.</p> <p>Representar fluxos de informação entre processos e entre atividades.</p> <p>Analisar e revisar artefatos construídos.</p> <p>Referenciar fontes de informação consultadas.</p> |
| ARTEFATOS | <p>Catálogo processo/evento/controle/produto</p> <p>Diagrama de fluxo de processo</p> <p>Diagrama de eventos</p>   |
| FONTES    | <p>Jacka e Keller (2009).</p> <p>The Open Group (Em:&lt;<a href="http://www.togaf.info/togaf9/togafSlides9/TOGAF-V9-Sample-Catalogs-Matrices-Diagrams-v2.pdf">http://www.togaf.info/togaf9/togafSlides9/TOGAF-V9-Sample-Catalogs-Matrices-Diagrams-v2.pdf</a>&gt;. Acessado em 26 de junho de 2013).</p> <p>Togaf-Modeling.org (Em:&lt;<a href="http://www.togaf-modeling.org/models/business-architecture-menu/process-flow-diagrams-menu.html">http://www.togaf-modeling.org/models/business-architecture-menu/process-flow-diagrams-menu.html</a>&gt;. Acessado em 25 de junho de 2013).</p> <p>Togaf-Modeling.org (Em:&lt;<a href="http://www.togaf-modeling.org/models/business-architecture-menu/event-diagrams-menu.html">http://www.togaf-modeling.org/models/business-architecture-menu/event-diagrams-menu.html</a>&gt;. Acessado em 25 de junho de 2013).</p>   |

Figura 40 – Roteiro Descrever processos do negócio.

|           |  |
|-----------|--|
| ROTEIRO   | Desenvolver taxonomia de termos e conceitos  |
| PROCESSO  | Desenvolver descrição da arquitetura atual ( <i>baseline</i> )   |
| DESCRIÇÃO | <p>Definir título do documento.</p> <p>Definir número de versão do documento.</p> <p>Definir data de elaboração do documento.</p> <p>Descrever objetivos do documento.</p> <p>Manter atualizado o histórico de alterações do documento.</p> <p>Definir objetivos (indexação, suporte à recuperação, organização, etc.) da taxonomia.</p> <p>Definir usuários da taxonomia.</p> <p>Definir conteúdo abordado, área de conhecimento e escopo da taxonomia.</p> <p>Definir tipos de relacionamentos permitidos (hierárquico, poli-hierárquico, equivalência, etc.).</p> <p>Definir como representar os relacionamentos permitidos.</p> <p>Definir como representar os níveis hierárquicos.</p> <p>Definir características estruturais (número máximo de níveis hierárquicos, etc.).</p> <p>Definir formas dos termos (número de palavras, forma gramatical, etc.).</p> <p>Definir atributos dos termos.</p> <p>Definir ferramentas para criar, administrar e usar a taxonomia.</p> <p>Definir fontes dos termos e conceitos.</p> <p>Coletar termos por meio de auditorias de conteúdos, entrevistas, etc.</p> <p>Agrupar termos em categorias.</p> <p>Identificar categorias de alto nível.</p> <p>Listar categorias de alto nível empregando ferramenta como planilha.</p> <p>Identificar categorias em níveis inferiores.</p> <p>Listar categorias em níveis inferiores empregando ferramenta como planilha.</p> <p>Representar relacionamentos hierárquicos entre termos.</p> <p>Identificar termos preferidos.</p> <p>Identificar termos não preferidos.</p> <p>Representar relacionamentos entre termos preferidos e não preferidos.</p> <p>Realizar testes com usuários.</p> <p>Revisar taxonomia.</p> <p>Selecionar normas e padrões às quais a taxonomia desenvolvida deve aderir.</p> <p>Adequar a taxonomia a normas e padrões selecionados.</p> <p>Referenciar fontes de informação consultadas.</p> |
| ARTEFATOS | Taxonomia  |
| FONTES    | <p>ANSI/NISO Z39.19-2005 (NISO, 2005).</p> <p>Hedden (2010).</p> <p>MAIA (COSTA, 2009).</p>  |

Figura 41 – Roteiro Desenvolver taxonomia de termos e conceitos.

|           |  |
|-----------|--|
| ROTEIRO   | Desenvolver mapa de conceitos  |
| PROCESSO  | Desenvolver descrição da arquitetura atual ( <i>baseline</i> )   |
| DESCRIÇÃO | Definir título do documento.<br>Definir número de versão do documento.<br>Definir data de elaboração do documento.<br>Descrever objetivos do documento.<br>Manter atualizado o histórico de alterações do documento.<br>Determinar domínio no qual o mapa de conceito será construído.<br>Identificar problema a ser resolvido pelo mapa de conceito.<br>Construir questão a ser respondida pelo mapa de conceito ( <i>focus question</i> ).<br>Identificar conceitos chave no domínio.<br>Listar os conceitos do mais genérico para o mais específico.<br>Selecionar ferramenta para a construção do mapa de conceitos.<br>Criar mapa de conceitos.<br>Posicionar conceitos identificados.<br>Criar ligações ( <i>links</i> ) para identificar relacionamentos entre conceitos.<br>Definir textos para identificar relacionamentos entre conceitos.<br>Revisar mapa de conceitos.<br>Reposicionar conceitos visando maior clareza e melhor estrutura.<br>Preparar versão final do mapa de conceitos.<br>Referenciar fontes de informação consultadas. |
| ARTEFATO  | Mapa de conceitos  |
| FONTES    | MAIA (COSTA, 2009).<br>Cañas, et al. (2004).<br>Novak & Cañas (2008).  |

Figura 42 – Roteiro Desenvolver mapa de conceitos.

|           |   |
|-----------|---|
| ROTEIRO   | Desenvolver ontologia de conceitos  |
| PROCESSO  | Desenvolver descrição da arquitetura atual ( <i>baseline</i> )  |
| DESCRIÇÃO | <p>Definir título do documento.</p> <p>Definir número de versão do documento.</p> <p>Definir data de elaboração do documento.</p> <p>Descrever objetivos do documento.</p> <p>Manter atualizado o histórico de alterações do documento.</p> <p>Definir ferramentas para criar, administrar e usar a ontologia.</p> <p>Definir domínio no qual a ontologia será construída.</p> <p>Delimitar escopo da ontologia considerando a aplicação da mesma.</p> <p>Definir perguntas que devem poder ser respondidas empregando-se a ontologia.</p> <p>Definir os responsáveis por usar a ontologia.</p> <p>Definir os responsáveis por prover a manutenção da ontologia.</p> <p>Definir convenção para os nomes de classes e <i>slots</i> (uso de maiúsculas, prefixos, etc.).</p> <p>Avaliar a possibilidade de reuso (refino e extensão) de ontologias existentes.</p> <p>Construir lista com os termos integrantes da ontologia.</p> <p>Definir informações relevantes acerca dos termos integrantes da ontologia.</p> <p>Escolher abordagem para construir a hierarquia de classes (<i>top-down, bottom-up, etc.</i>).</p> <p>Identificar as classes (conceitos) a partir da lista de termos.</p> <p>Atribuir nomes às classes considerando a convenção de nomes adotada.</p> <p>Descrever as classes.</p> <p>Organizar as classes em uma hierarquia (relacionamento é um tipo de).</p> <p>Avaliar quantidade de níveis hierárquicos.</p> <p>Definir propriedades das classes (atributos e relacionamentos) a partir da lista de termos.</p> <p>Definir <i>slots</i> das classes a partir das propriedades identificadas.</p> <p>Atribuir nomes aos <i>slots</i> considerando a convenção de nomes adotada.</p> <p>Definir características dos <i>slots</i> (tipo, faixa de valores, cardinalidade, etc.) das classes.</p> <p>Instanciar as classes.</p> <p>Atribuir valores aos <i>slots</i> das instâncias.</p> <p>Testar a ontologia.</p> <p>Revisar a ontologia.</p> <p>Referenciar fontes de informação consultadas.</p> |
| ARTEFATO  | Ontologia   |
| FONTES    | <p>MAIA (COSTA, 2009).</p> <p>Noy e McGuinness (2001).</p>  |

Figura 43 – Roteiro Desenvolver ontologia de conceitos.

## APÊNDICE C – Bibliometria

Neste apêndice são apresentados dados acerca das fontes de informação referenciadas ao longo desta dissertação e resultados de consultas realizadas à fonte de informação Web of Science. Na fase da pesquisa bibliográfica descrita nesta dissertação, variadas fontes de informação foram consultadas. A seguir, são relacionadas as fontes mais consultadas:

- ABNT Coleção;
- ACM *Digital Library*;
- Base de dados ProQuest;
- Base de dados *Web of Science*;
- Biblioteca Central da Universidade de Brasília (BCE);
- Biblioteca do Instituto Brasileiro de Informação em Ciência e Tecnologia (IBICT);
- Biblioteca Digital Brasileira de Computação (BDBComp);
- Biblioteca digital EBRARY;
- IEEE Xplore *Digital Library*;
- Portal de periódicos da CAPES;
- Repositório Institucional da Universidade de Brasília;
- *Scientific Electronic Library Online* (SciELO);
- Sítios das editoras Prentice Hall, Addison-Wesley e John Wiley and Sons;
- Sítios das livrarias Amazon.com, Barnes & Noble, Livraria Cultura e Siciliano;
- Sítios dos sistemas de busca Yahoo! e Google;
- *The Information Architecture Institute*.

Quanto à atualidade das informações acessadas ao longo desta dissertação, na figura 44 encontram-se dados acerca da distribuição dos documentos referenciados por ano de publicação.

| ANO  | %   | ANO  | %   | ANO  | %   |
|------|-----|------|-----|------|-----|
| 2013 | 5.1 | 2012 | 7.9 | 2011 | 8.5 |
| 2010 | 8.0 | 2009 | 5.1 | 2008 | 4.0 |
| 2007 | 5.7 | 2006 | 7.4 | 2005 | 4.0 |
| 2004 | 3.4 | 2003 | 5.7 | 2002 | 7.4 |
| 2001 | 4.0 | 2000 | 5.7 | 1999 | 2.3 |
| 1998 | 2.3 | 1997 | 1.7 | 1996 | 1.2 |
| 1995 | 2.3 | 1994 | 1.7 | 1993 | 0.6 |
| 1992 | 1.2 | 1991 | 2.8 | 1990 | 0.6 |
| 1987 | 0.6 | 1985 | 0.6 | 1984 | 0.6 |

Figura 44 – Distribuição dos documentos referenciados por ano de publicação.

Com o intuito de possibilitar uma avaliação acerca da atualidade e ineditismo da pesquisa realizada, neste apêndice encontram-se dados resultantes de consultas à fonte de informação Web of Science e dados acerca das fontes de informação referenciadas nesta dissertação. A primeira consulta à fonte de informação Web of Science visou localizar documentos que satisfizessem os seguintes critérios: tópico (título, *abstract*, palavras-chave) com o termo “*information architecture*”, “arquitetura da informação” ou “*arquitectura de la información*”; documento em inglês, português ou espanhol; documento do tipo “*article*” ou “*proceedings paper*”; documento publicado entre 1994 e 2013. Na na figura 45 encontram-se as quantidades de documentos por ano de publicação, enquanto na figura 46 encontram-se informações acerca da distribuição desses documentos por áreas de pesquisa.

A segunda consulta à fonte de informação Web of Science visou localizar documentos que satisfizessem os seguintes critérios: tópico com os termos “*information architecture*” e “*software engineering*”, ou “arquitetura da informação” e “engenharia de *software*” ou “*arquitectura de la información*” e “*ingeniería de software*”; documento em inglês, português ou espanhol; documento do tipo “*article*” ou “*proceedings paper*”; documento publicado entre 1994 e 2013. Na figura 47 encontram-se as quantidades de documentos por ano de publicação, enquanto na figura 48 encontram-se informações acerca da distribuição desses documentos por áreas de pesquisa.

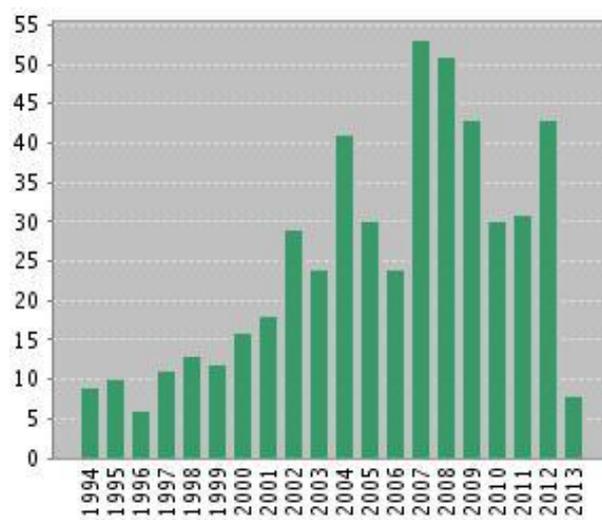


Figura 45 – Documentos por ano de publicação (Fonte: Web of Science).

| Campo: Áreas de pesquisa               | Contagem do registro | % de 502 | Gráfico de barras |
|--|----------------------|----------|-------------------|
| COMPUTER SCIENCE                       | 297                  | 59.163 % |                   |
| ENGINEERING                            | 136                  | 27.092 % |                   |
| INFORMATION SCIENCE<br>LIBRARY SCIENCE | 120                  | 23.904 % |                   |
| BUSINESS ECONOMICS                     | 53                   | 10.558 % |                   |
| TELECOMMUNICATIONS                     | 36                   | 7.171 %  |                   |

Figura 46 – Documentos por áreas de pesquisa (Fonte: Web of Science).

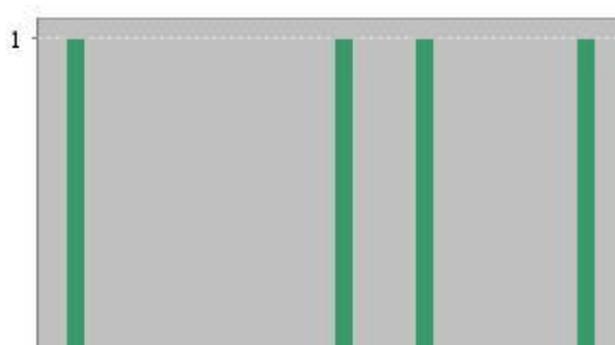


Figura 47 – Documentos por ano de publicação (Fonte: Web of Science).

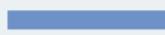
| Campo: Áreas de pesquisa | Contagem do registro | % de 4   | Gráfico de barras   |
|--------------------------|----------------------|----------|---|
| COMPUTER SCIENCE         | 3                    | 75.000 % |  |
| ENGINEERING              | 2                    | 50.000 % |  |

Figura 48 – Documentos por áreas de pesquisa (Fonte: Web of Science).

A terceira consulta, realizada em 20/11/2013 à fonte de informação Web of Science, visou localizar documentos que satisfizessem os seguintes critérios: tópico com os termos “*information architecture*” e “*software development process*”, ou “arquitetura da informação” e “processo de desenvolvimento de *software*” ou “*arquitectura de la información*” e “*proceso de desarrollo de software*”; documento em inglês, português ou espanhol; documento do tipo “*article*” ou “*proceedings paper*”; documento publicado entre 1994 e 2013. Essa consulta localizou apenas um artigo.