

UNIVERSIDADE FEDERAL DA PARAÍBA - UFPB  
CENTRO DE INFORMÁTICA - CI  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA – PPGI

*FRAMEWORK* PARA SUPORTE À EVOLUÇÃO DE  
ONTOLOGIAS BIOMÉDICAS

ANA PATRÍCIA DE SOUSA

JOÃO PESSOA – PB  
FEVEREIRO 2019

**ANA PATRÍCIA DE SOUSA**

***FRAMEWORK* PARA SUPORTE À EVOLUÇÃO DE  
ONTOLOGIAS BIOMÉDICAS**

JOÃO PESSOA – PB  
FEVEREIRO 2019

**ANA PATRÍCIA DE SOUSA**

***FRAMEWORK* PARA SUPORTE À EVOLUÇÃO DE  
ONTOLOGIAS BIOMÉDICAS**

DISSERTAÇÃO APRESENTADA AO CENTRO DE INFORMÁTICA DA  
UNIVERSIDADE FEDERAL DA PARAÍBA, COMO REQUISITO PARCIAL  
PARA OBTENÇÃO DO TÍTULO DE MESTRE EM INFORMÁTICA  
(SISTEMAS DE COMPUTAÇÃO)

**Orientadora:** Prof.<sup>a</sup>. Dr.<sup>a</sup>. Natasha Correia Queiroz Lino

JOÃO PESSOA – PB  
FEVEREIRO 2019

**Catálogo na publicação**  
**Seção de Catalogação e Classificação**

S725f Sousa, Ana Patricia de.

FRAMEWORK PARA SUPORTE À EVOLUÇÃO DE ONTOLOGIAS  
BIOMÉDICAS / Ana Patricia de Sousa. - João Pessoa,  
2019.

199 f. : il.

Orientação: Natasha Correia Queiroz Lino.

Coorientação: Claurton de Albuquerque Siebra.

Dissertação (Mestrado) - UFPB/CI.

1. Evolução de Ontologias. 2. Ontologias Biomédicas. 3.  
Framework. I. Lino, Natasha Correia Queiroz. II. Título.

UFPB/BC



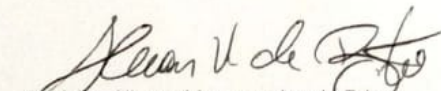


UNIVERSIDADE FEDERAL DA PARAÍBA  
CENTRO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA


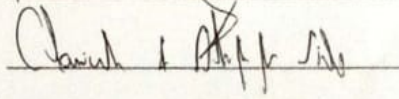


Ata da Sessão Pública de Defesa de Dissertação de Mestrado de Ana Patrícia de Sousa, candidata ao título de Mestre em Informática na Área de Sistemas de Computação, realizada em 31 de janeiro de 2019.

1 Aos trinta e um dias do mês de janeiro, do ano de dois mil e dezanove, às oito horas, no  
2 Centro de Informática da Universidade Federal da Paraíba, em Mangabeira, reuniram-se os  
3 membros da Banca Examinadora constituída para julgar o Trabalho Final da Sr<sup>a</sup>. Ana  
4 Patrícia de Sousa, vinculada a esta Universidade sob a matrícula nº 20161020962, candidata  
5 ao grau de Mestre em Informática, na área de "Sistemas de Computação", na linha de  
6 pesquisa "Computação Distribuída", do Programa de Pós-Graduação em Informática, da  
7 Universidade Federal da Paraíba. A comissão examinadora foi composta pelos professores:  
8 Natasha Correia Queiroz Lino (PPGI-UFPB) Orientadora e Presidente da Banca,  
9 Claurton de Albuquerque Siebra (PPGI-UFPB), Examinador Interno, Cecir Barbosa de  
10 Almeida Farias (UFCG), Examinadora Externa à Instituição, Cecília Neta Alves Pegado  
11 Gomes (FAMENE), Examinadora Externa à Instituição. Dando início aos trabalhos, a  
12 Presidente da Banca, cumprimentou os presentes, comunicou aos mesmos a finalidade da  
13 reunião e passou a palavra à candidata para que a mesma fizesse a exposição oral do  
14 trabalho de dissertação intitulado: "Framework para suporte a evolução de ontologias  
15 biomédicas". Concluída a exposição, a candidata foi arguida pela Banca Examinadora que  
16 emitiu o seguinte parecer: "**aprovada**". Do ocorrido, eu, Alisson Vasconcelos de Brito, Vice-  
17 Coordenador do Programa de Pós-Graduação em Informática, lavrei a presente ata que vai  
18 assinada por mim e pelos membros da banca examinadora. João Pessoa, 31 de janeiro de  
19 2019.

  
Prof. Dr. Alisson Vasconcelos de Brito

Prof. Dr<sup>a</sup>. Natasha Correia Queiroz Lino  
Orientadora (PPGI-UFPB)

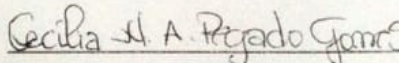
  


Prof. Dr. Claurton de Albuquerque Siebra  
Examinador Interno (PPGI-UFPB)

Prof. Dr<sup>a</sup>. Cecir Barbosa de Almeida Farias  
Examinadora Externa ao Programa (UFCG)



Cecília Neta Alves Pegado Gomes  
Examinadora Externa ao Programa (FAMENE)



## **AGRADECIMENTOS**

Agradeço à minha orientadora, professora Dr<sup>a</sup> Natasha C. Q. Lino, aos meus amigos e à minha família.

## Resumo

As ontologias Biomédicas crescem continuamente e por isso necessitam ser refinadas ao longo do seu ciclo de vida. Nas ontologias Biomédicas, pequenas alterações podem causar efeitos inesperados em grandes partes da ontologia, isso pode ser explicado pela sua alta complexidade e pelo fato de serem, em geral, maiores que as ontologias de outros domínios. O processo de refinamento para que as ontologias sejam adaptadas às mudanças ocorridas em suas fontes de conhecimento se chama evolução de ontologias. Nesse contexto, no presente trabalho foi construído um *framework* conceitual para auxiliar desenvolvedores de ontologias Biomédicas durante seu processo de evolução. Seu objetivo principal é oferecer sugestões validadas de atividades e ferramentas para apoiar a evolução de ontologias Biomédicas. O *framework* definido está dividido em 5 fases compreensivas e bem definidas denominadas Planejamento da Evolução, Implementação de Mudanças, Detecção de Mudanças, Tratamento de Inconsistências e Auditoria de Mudanças. O *framework* foi validado por meio de um estudo de caso onde o mesmo foi aplicado a uma ontologia sobre a Doença Renal Crônica submetida a evolução em três perspectivas baseadas em três motivos de evolução: a mudança do foco da ontologia, mudanças para refletir alterações em suas diretrizes de domínio e mudanças relativas à terminologia de referência da ontologia. A consistência da ontologia utilizada no estudo de caso foi verificada por meio da aplicação de consultas *SPARQL* baseadas em Questões de Competência pré-definidas pelo especialista do domínio da ontologia. O *framework* desenvolvido possui como principal benefício o provimento de recursos para apoiar todo o processo de evolução, desde a representação da mudança até a verificação da conformidade da ontologia evoluída com o plano de evolução. Com base nos resultados dos experimentos realizados pode-se concluir que o *framework* permitiu manter organizado e compreensivo o processo evolutivo da ontologia utilizada. A organização e compreensividade fornecida pelo *framework* são importantes pois facilitam a recuperação de informações caso haja a necessidade de refazer alguma mudança. Os relatórios fornecidos como saída das atividades que compõem os subprocessos do *framework*, além de subsidiar a realização das atividades às quais são dados como entradas, podem ser utilizados como ferramentas de predição de mudanças, por meio das quais as mudanças recorrentes podem ser identificadas, o que pode ajudar na definição prévia de estratégias de aplicação de mudanças bem como de correção de erros identificados durante a validação da ontologia.

**Palavras-chave:** Evolução de Ontologias, Ontologias Biomédicas, *Framework*.

## **Abstract**

Biomedical ontologies grow continuously and therefore need to be refined throughout their life cycle. In Biomedical ontologies, small changes can cause unexpected effects on large parts of the ontology, which can be explained by their high complexity and the fact that they are, in general, larger than the ontologies of other domains. The process of refinement in order for ontologies to be adapted to the changes occurring in their knowledge sources is called evolution of ontologies. In this context, in the present work a conceptual framework was created to assist Biomedical ontology developers during their evolution process. Its main objective is to offer validated suggestions of activities and tools to support the evolution of Biomedical ontologies. The defined framework is divided into 5 comprehensive and well-defined phases called Evolution Planning, Implementation of Changes, Detection of Changes, Treatment of Inconsistencies and Audit of Changes. The framework was validated through a case study where it was applied to an ontology on Chronic Kidney Disease submitted to evolution in three perspectives based on three reasons for evolution: the change of the focus of the ontology, changes to reflect changes in their domain guidelines and changes related to ontology reference terminology. The consistency of the ontology used in the case study was verified through the application of SPARQL queries based on Competency Issues predefined by the ontology domain expert. The developed framework has as its main benefit the provision of resources to support the whole process of evolution, from the representation of the change to the verification of the conformity of the evolved ontology with the evolution plan. Based on the results of the experiments carried out, it is possible to conclude that the framework allowed to keep organized and understanding the evolutionary process of the ontology used. The organization and comprehensiveness provided by the framework are important because it facilitates the retrieval of information if there is a need to redo any changes. In addition to subsidizing the activities for which they are provided as inputs, the reports provided as a result of the activities that make up the subprocesses of the framework can be used as tools for predicting changes. This can be useful in the prior definition of strategies for applying changes and correcting errors identified during ontology validation.

**Keywords:** Ontology Evolution, Biomedical Ontologies, Framework.

# Sumário

1	Introdução.....	14
1.1	Motivação.....	14
1.2	Objetivos.....	19
1.2.1	Objetivo Geral.....	19
1.2.2	Objetivos Específicos.....	19
1.3	Organização do documento.....	20
2	Fundamentação Teórica.....	22
2.1	Representação do Conhecimento.....	22
2.2	ontologias.....	23
2.2.1	Definições Filosóficas.....	23
2.2.2	Definições na esfera da Inteligência Artificial.....	24
2.2.3	Conceitualização.....	26
2.2.4	Critérios para projeto de ontologias.....	27
2.2.4.1	Clareza.....	28
2.2.4.2	Coerência.....	28
2.2.4.3	Extensibilidade.....	28
2.2.4.4	Mínima codificação parcial.....	28
2.2.4.5	Mínimo compromisso ontológico.....	29
2.3	Ontologias Biomédicas.....	29
2.4	Evolução de ontologias.....	32
3	Trabalhos Relacionados.....	43
4	Materiais e Métodos.....	64
4.1	Concepção do <i>framework</i> .....	65
4.1.1	Análise Bibliográfica.....	65

4.1.2 Estudo de caso para apoio à construção do <i>framework</i> .....	65
4.1.2.1 Levantamento de Requisitos.....	68
4.1.2.2 Definição do Escopo.....	68
4.1.2.3 Padronização.....	69
4.1.2.4 Validação.....	70
4.1.2.5 Resultados.....	70
4.1.2.6 Conclusão.....	72
4.1.3 Construção do <i>framework</i> para suporte à evolução de ontologias Biomédicas.....	73
4.2 Validação do <i>Framework</i> .....	77
5 Framework para suporte à evolução de ontologias Biomédicas.....	79
5.1 Definições Preliminares e Anotações.....	79
5.2 Processos do Framework.....	83
5.2.1 Planejamento da Evolução - PE.....	87
5.2.1.1 Captura das Mudanças.....	87
5.2.1.2 Representação das Mudanças.....	88
5.2.2 Implementação das mudanças - IM.....	106
5.2.2.1 Classificação das Mudanças – CM.....	107
5.2.2.2 Edição da ontologia - EO.....	109
5.2.3 Detecção de mudanças - DM.....	117
5.2.3.1 Comparação entre Versões (CV).....	119
5.2.3.2 Aceitação/Rejeição de Mudanças.....	121
5.2.4 Tratamento de Inconsistências - TI.....	122
5.2.4.1 Detecção de Inconsistências.....	124
5.2.4.2 Classificação de Inconsistências – CI.....	126
5.2.4.3 Resolução de inconsistências.....	129
5.2.5 Auditoria de Mudanças - AM.....	133
5.2.5.1 Análise da conformidade.....	134

5.2.5.2 Testes de consistência.....	135
5.3 Conclusão do Capítulo.....	137
5.4 Notas sobre Capítulo.....	142
6 Estudo de Caso.....	143
6.1 Uso do <i>framework</i> em uma ontologia em evolução por mudança de foco.....	156
6.2 Uso do <i>framework</i> em uma ontologia em evolução por mudanças em suas diretrizes de domínio.....	165
6.3 Uso do <i>framework</i> em uma ontologia em evolução por mudanças em sua terminologia de referência.....	173
6.4 Conclusão do capítulo.....	176
6.5 Notas sobre o capítulo.....	178
7 Conclusões.....	179
7.1 Contribuições.....	179
7.2 Trabalhos Futuros.....	182
Bibliografia.....	184
Anexo A.....	192
Anexo B.....	197

## Lista de Tabelas

Tabela 1. Estatísticas de três importantes ontologias Biomédicas.....	31
Tabela 2. Exemplos de mudanças compostas.....	38
Tabela 3. Resultado das buscas por temas do processo de evolução de ontologias em 19 trabalhos relacionados.....	63
Tabela 4. Exemplo de organização do relatório de mudanças a serem aplicadas em uma ontologia em evolução por mudança de foco.....	91
Tabela 5. Exemplo de organização do relatório de mudanças a serem aplicadas em uma ontologia em evolução por mudanças em suas diretrizes de domínio.....	94
Tabela 6. Mudanças sobre anatomia implementadas no release de janeiro de 2019 da SNOMED CT edição Internacional.....	96
Tabela 7. Resumo dos conceitos adicionados no release de janeiro de 2019 da SNOMED CT edição internacional.....	97
Tabela 8. Resumo dos conceitos modificados no release de janeiro de 2019 da SNOMED CT edição internacional.....	97
Tabela 9. Exemplo de organização do relatório de mudanças a serem aplicadas em uma ontologia em evolução por alterações em sua terminologia de referência.....	100
Tabela 10. Exemplo de organização do relatório de mudanças a serem aplicadas em uma ontologia em evolução para padronização terminológica total.....	104
Tabela 11. Exemplo de relatório de mudanças classificadas para serem implementadas em uma ontologia em evolução por mudanças em suas diretrizes de domínio.....	108
Tabela 12. Exemplo de relatório de mudanças detectadas em uma ontologia sobre a DRC em evolução por mudança de foco.....	125
Tabela 13. Exemplo de relatório de inconsistências classificadas para uma ontologia sobre a DRC em evolução por mudança de foco.....	128



Tabela 14. Exemplos de conceitos de nível superior em SNOMED CT.....	151
Tabela 15. Lista de mudanças especificadas para aplicação na versão da ontologia ontoDecideDRC em evolução por mudança de foco.....	158
Tabela 16. Diferenças entre a versão anterior e a versão alterada da ontoDecideDRC em evolução por mudança de foco.....	159
Tabela 17. Inconsistências identificadas na versão da ontoDecideDRC em evolução por mudança de foco.....	161
Tabela 18. Questões de competência da ontoDecideDRC após evolução por mudança de foco .....	162
Tabela 19. Consultas SPARQL criadas a partir das questões de competência da versão da ontologia ontoDecideDRC em evolução por mudança de foco.....	162
Tabela 20. Lista de mudanças especificadas para aplicação na versão da ontologia ontoDecideDRC em evolução por mudanças em suas diretrizes de domínio.....	166
Tabela 21. Lista de mudanças especificadas e classificadas para aplicação na versão da ontologia ontoDecideDRC em evolução por mudanças em suas diretrizes de domínio.....	167
Tabela 22. Inconsistências identificadas na versão da ontoDecideDRC em evolução por mudanças em suas diretrizes de domínio.....	168
Tabela 23. Questões de competência da ontoDecideDRC.....	169
Tabela 24. Consultas SPARQL criadas a partir das questões de competência da versão da ontologia ontoDecideDRC em evolução por mudanças em suas diretrizes de domínio.....	170
Tabela 25. Relatório de mudanças a serem aplicadas na ontologia ONTODRC em evolução para padronização terminológica total.....	174

## **Lista de Siglas**

***AM:** Aprendizagem de Máquina*

***CBN:** Causal Bayesian Network*

***CIS:** Clinical Information Systems*

***CDL:** Changes Definition Language*

***CHAO:** Change and Annotation ontology*

***CHL:** Change History Log*

***CHO:** Change History ontology*

***CTV3:** Clinical Terms Version 3*

***DL:** Description Logic*

***DRC:** Doença Renal Crônica*

***FMA:** Foundational Model of Anatomy*

***FSN:** Fully Specified Name*

***GO:** Gene ontology*

***IA:** Inteligência Artificial*

***ICD:** International Classification of disease*

***IHTSDO:** International Health Terminology Standards Development Organization*

***KAON:** Kaulshure ontology on Semantic Web framework*

***KOS:** Knowledge Organization Systems*

***LAL:** Léxico Ampliado da Linguagem*

***LOINC:** Logical Observation Identifiers Names and Codes*

***NCIt:** National Cancer Institute Thesaurus*

***NCBO:** National Center for Biomedical ontology*

**OBO:** *Open Biological and Biomedical ontologies Foundry*

**OWL:** *Web ontology Language*

**OWL DL:** *Web ontology Language Description Logic*

**QC:** *Questão de Competência*

**RDF:** *Resource Description Framework*

**RES:** *Registro Eletrônico de Saúde*

**SCTID:** *SNOMED CT identifier*

**SNOMED CT:** *Systematized Nomenclature of Medicine - Clinical Terms*

**SNOMED RT:** *SNOMED Reference Terminology*

**SPARQL:** *SPARQL Protocol and RDF Query Language*

**TFG:** *Taxa de Filtração Glomerular*

# Capítulo 1

## 1 Introdução

### 1.1 Motivação

Os Sistemas de Organização do Conhecimento, do inglês, *Knowledge Organization Systems* (KOS), englobam todos os tipos de esquemas de organização de informações, por exemplo: terminologias, taxonomias, classificação e ontologias. Os KOSs são amplamente utilizados no domínio biomédico para apoiar o compartilhamento de informações entre aplicações de *software*. Nesse domínio, tais sistemas compreendem propriedades para tornar explícita a semântica dos dados, ou seja, tornar claro o sentido dos dados por meio do uso de propriedades que auxiliem nesse processo (REIS *et al.*, 2014).

Podem ser citados como exemplos de KOSs biomédicos a Nomenclatura Sistematizada de Medicina - Termos Clínicos (SNOMED CT ou SCT, do inglês, *Systematized Nomenclature of Medicine - Clinical Terms*) e a Classificação Internacional de Doenças (ICD, do inglês *International Classification of disease*) que são amplamente utilizados atualmente (REIS *et. al*, 2014).

Hartung, Groß e Rahm (2013) afirmam que as ontologias tornaram-se cada vez mais importantes para anotar e categorizar as informações de maneira semântica e consistente. Nesse sentido, ontologias são usadas como vocabulários compartilhados para aprimorar a recuperação de informações ou dar suporte à integração de dados.

Por ser um processo dinâmico, o desenvolvimento de ontologias deve passar por refinamento de modo que ao longo do tempo possa mudar e se adaptar às necessidades do seu ambiente de aplicação bem como dos seus usuários e evitar a obsolescência do conhecimento nelas contidos. A evolução de ontologias é o processo no qual o sistema de gerenciamento de ontologias de um ambiente permite a realização de mudanças sem comprometer a consistência do conteúdo da ontologia submetida à evolução (STOJANOVIC, 2004). Análogo à definição de Stojanovic (2004) para evolução de ontologias, o conceito dado por

Gawich et al. (2016) considera evolução de ontologias um processo de adaptação da ontologia às mudanças surgidas em seu domínio, mantendo simultaneamente a consistência da mesma, bem como a consistência dos artefatos que dela dependem. Outras ontologias, sites e aplicações *web* são exemplos de artefatos dependentes de ontologias.

Para Flouris, Plexousakis e Antoniou (2006), a evolução de uma ontologia pode ser causada por uma mudança em seu domínio, conceituação ou especificação. O número de ontologias sobre as ciências da vida cresce continuamente, conforme destacam Hartung, Groß e Rahm (2013). Um exemplo disso é o fato de a Fundação de Ontologias Biológicas e Biomédicas Abertas (OBO, do inglês, *Open Biological and Biomedical Ontologies Foundry*)<sup>1</sup> e o BioPortal<sup>2</sup> terem oferecido em 2013 acesso a cerca de 250 ontologias cada, abrangendo conhecimentos de diferentes domínios em ciências da vida, como anatomia, fenótipo, função biológica ou bioquímica.

A maioria das ontologias voltadas para as ciências da vida evoluem para atender às novas exigências da área em estudo, corrigir erros de projeto prévios ou melhor incorporar conhecimentos de domínio recém disponibilizados. Assim, há o lançamento contínuo de novas versões da ontologia refletindo suas últimas alterações. Um exemplo desse tipo de ontologia é a *Gene ontology* (GO)<sup>3</sup> ou *National Cancer Institute Thesaurus*<sup>4</sup> (NCIt) que teve seu tamanho duplicado desde 2004 (HARTUNG, GROß e RAHM, 2013).

As ciências da vida são um domínio altamente dinâmico por natureza. Em consequência disso novas descobertas conduzem a constantes renovações do conhecimento do domínio, tornando-o mais rico ao longo do tempo. Tal evolução afeta profundamente as ontologias de domínio, forçando os especialistas a revisarem regularmente seu conteúdo. Dessa forma, a evolução de ontologias mostra-se como um campo de pesquisa que tem ganhado maior importância com o passar dos anos por meio da junção de esforços da comunidade biomédica e da *Web Semântica* (GROß, PRUSKI, e RAHM, 2016).

A partir de um trabalho realizado pela autora e colegas para aquisição de requisitos que auxiliassem na construção do *framework* proposto, três motivos relevantes para a

1<http://obofoundry.org/>

2<https://bioportal.bioontology.org/>

3<http://www.geneontology.org/>

4<https://ncit.nci.nih.gov/ncitbrowser/>

evolução de ontologias do domínio Biomédico foram identificados. Os motivos identificados compreendem: (1) a mudança de foco da ontologia, (2) a necessidade de incorporar mudanças realizadas em suas diretrizes de domínio e (3) a implementação necessária de mudanças relativas à terminologia de referência da ontologia.

A mudança de foco de uma ontologia ocorre quando é necessário representar conceitos pertencentes ao mesmo domínio da ontologia, porém, relativos a um conteúdo distinto do que está sendo abordado atualmente. Por exemplo, estará em evolução por mudança de foco, uma ontologia sobre o cuidado especializado da Doença Renal Crônica (DRC) que deverá passar a representar apenas conceitos relacionados aos cuidados realizados durante o acompanhamento do paciente antes de ser encaminhado para o Nefrologista, que é o profissional responsável pelo tratamento da DRC. Conforme o exposto nesse exemplo, ainda que o domínio da ontologia continue sendo a DRC, o foco da mesma passará a ser a atenção primária do paciente.

Em seu sentido mais amplo, diretrizes contituem-se de documentos que têm como finalidade informar e fornecer recomendações. Voltando esse termo para a área médica, as diretrizes são utilizadas com a finalidade de melhorar os cuidados prestados a determinado paciente. Os documentos por meio dos quais contituem-se as diretrizes, são criados considerando as evidências científicas julgadas como melhores. As diretrizes clínicas também são usadas no processo de gerenciamento e regulação dos sistemas de saúde (CONITEC, 2016).

Nesse contexto, considerando a importância do conteúdo presente nas diretrizes de um domínio, a(s) ontologia(s) que utiliza(m) documentos como este como fonte de conhecimento, devem refletir quaisquer alterações realizadas em seu conteúdo. Essa alteração é importante pois permite que os conceitos da ontologia permaneçam consistentes em relação ao conteúdo das suas diretrizes de domínio. Por exemplo, o documento de Avaliação e Gerenciamento da DRC (do inglês, *CKD Evaluation and Management*)<sup>5</sup> versão 1 fornecido pela *Kidney Disease: Improving Global Outcomes (KDIGO)*<sup>6</sup> teve seu conteúdo alterado e uma nova versão do documento foi lançada. Considerando que na versão recém lançada, um novo

<sup>5</sup><http://kdigo.org/guidelines/ckd-evaluation-and-management/>

<sup>6</sup><http://kdigo.org>

estágio da DRC foi incluído, as ontologias sobre a DRC que utilizam essa diretriz clínica como fonte de conhecimento deverão ser evoluídas tendo as mudanças em suas diretrizes de domínio como motivo de evolução de modo que a alteração referente à adição desse novo conceito seja refletida na ontologia. Remoções de conceitos e alterações realizadas em conceitos existentes nas diretrizes de um domínio também devem ser refletidas nas ontologias que as utilizam como fonte de conhecimento.

Mudanças ocasionadas por alterações realizadas na terminologia de referência de uma ontologia também devem ser refletidas em sua estrutura. A inclusão de novos conceitos, remoção de conceitos obsoletos e melhoramento de conceitos existentes são exemplos de alterações realizadas nos elementos utilizados como referência de uma ontologia e devem ser incorporados ao seu conteúdo. Tais elementos podem ser outras ontologias, terminologias cuja finalidade é servir como um padrão para nomenclatura de conceitos a ser utilizado por órgãos públicos ou privados de modo obrigatório ou não como: SNOMED CT, ICD-10 entre outros; diretrizes do domínio abordado na ontologia ou suas demais fontes de conhecimento.

A evolução de uma ontologia relativa à sua terminologia de referência também pode ser realizada com a finalidade de padronizá-la totalmente após a decisão do Engenheiro de Ontologias de mudar o padrão usado atualmente ou passar a utilizar uma terminologia cuja finalidade principal seja padronizar termos em vez de utilizar as fontes de conhecimento da ontologia como base para a sua representação terminológica. Um ontologia sobre a DRC que utiliza o padrão SNOMED CT como terminologia padrão deverá ser submetida à evolução por mudanças em sua terminologia de referência caso o termo “*Chronic Kidney Disease Stage I*” seja alterado para “*CKD Stage I*”.

Nesse contexto, tendo em vista o crescimento da utilização das ontologias pertencentes à área das ciências da vida bem como a importância do processo de evolução de ontologias, neste trabalho foi desenvolvido um *framework* para dar suporte ao processo de evolução de ontologias no domínio Biomédico com a finalidade de apoiar, de forma estruturada, o processo evolutivo dessas ontologias quando motivado pelos três motivos de evolução abordados anteriormente.

Conforme o dicionário Cambridge (2019), um *framework* é um conjunto particular de regras, ideias ou crenças usadas em geral para lidar com problemas e/ou auxiliar na tomada de

decisão. Dessa forma, o *framework* desenvolvido é composto por uma coleção de tarefas a serem realizadas com o objetivo de tornar claro o processo evolutivo das ontologias Biomédicas considerando as suas particularidades e tendo em vista as especificidades exigidas pelos três motivos de evolução identificados no presente trabalho.

O *framework* desenvolvido está dividido em 5 fases denominadas de Planejamento da Evolução, Implementação das Mudanças, Detecção das Mudanças, Tratamento de Inconsistências e Auditoria das Mudanças. Cada uma dessas fases é constituída por um conjunto de atividades a serem realizadas para alcançar seu objetivo. A divisão do *framework* em 5 fases foi estabelecida a partir de estudos e análises acerca do processo evolutivo tradicional de ontologias observado na maioria dos trabalhos sobre evolução de ontologias identificados na literatura e considerou o fato das ontologias Biomédicas serem em geral maiores que as ontologias de outros domínios. Esse fato faz as ontologias desse domínio necessitarem de um processo evolutivo claro e compreensivo de modo que se evite que erros decorrentes da evolução não sejam identificados e inconsistências que podem causar problemas lógicos e estruturais na ontologia, passem despercebidas.

Com base nos motivos de evolução identificados, definiu-se três perspectivas de evolução para as quais cada fase do *framework* foi adaptada. Assim, no *framework* construído aborda-se a evolução de ontologias Biomédicas ocasionada por mudança de foco, onde a ontologia passa a refletir conceitos relativos a uma área distinta da atual, porém pertencente ao mesmo domínio; mudanças ocorridas em suas diretrizes de domínio, onde a ontologia precisa passar por evolução para as refletir as mudanças realizadas em suas fontes de conhecimento e; mudanças referentes à sua terminologia de referência, onde a ontologia é submetida a evolução para implementar as alterações ocorridas na terminologia utilizada para padronizar seus termos ou para realizar a sua padronização terminologica total.

Para validar o *framework* diante das três perspectivas de evolução identificadas, foi realizado um estudo de caso onde este foi aplicado a três versões de uma ontologia sobre a DRC em processo de evolução. Para verificar a consistência das versões da ontologia utilizada no estudo de caso foram aplicadas consultas *SPARQL Protocol and RDF Query Language* (SPARQL)<sup>7</sup> criadas a partir das Questões de Competência (QCs) que definem os

<sup>7</sup><https://www.w3.org/TR/rdf-sparql-query/>



limites da ontologia utilizada no estudo. Tais QCs foram criadas com o apoio do especialista do domínio da ontologia que acompanhou o seu processo evolutivo.

A realização do estudo de caso para validar o *framework* construído oferece como contribuição principal, a visualização do mesmo de um ângulo voltado para a sua aplicação prática por meio do qual o seu comportamento pode ser avaliado. Essa visualização permite que possíveis limitações sejam identificadas e melhorias possam ser sugeridas. Além disso, o resultado satisfatório da realização dos experimentos permite concluir que o *framework* oferece meios suficientes para dar suporte à evolução adequada de ontologias do domínio Biomédico.

## **1.2 Objetivos**

### **1.2.1 Objetivo Geral**

O presente trabalho possui como objetivo geral, a modelagem e construção de um *framework* conceitual composto de um conjunto de processos baseados nas fases que devem ser seguidas durante um processo de evolução de ontologias tradicional, adaptando-as à realidade das ontologias Biomédicas quando submetidas à evolução motivada por mudança de foco, mudanças em suas diretrizes de domínio e mudanças relativas à sua terminologia de referência.

### **1.2.2 Objetivos Específicos**

De acordo com o objetivo geral disposto na subseção anterior, os objetivos específicos desta trabalho são:

- Apoiar o processo de planejamento da evolução da ontologia – O planejamento da evolução de uma ontologia geralmente começa com a definição dos motivos que a levaram a precisar evoluir. O apoio à definição dos motivos de evolução da ontologia diz respeito a ajudar os responsáveis pela evolução a definirem o que levou a ontologia a ser submetida a tal processo. Isso facilita a representação das mudanças que devem ser realizadas. Assim, esse objetivo específico tem como intenção principal ajudar os responsáveis pela evolução no planejamento das mudanças requeridas por

usuários ou especialistas de domínio, bem como sua representação, facilitando desse modo, o processo de edição da ontologia;

- Fornecer sugestões de métodos e materiais para apoiar a edição de ontologias - O processo de implementação de mudanças não deve ser discricionário. É importante que o mesmo seja supervisionado de forma a se ter controle de todas as alterações aplicadas para fins de auditoria e direcionamento imediato ao problema em caso de mal funcionamento ocasionado por erros de evolução. Por esse motivo, esse objetivo específico foi formulado com o fim de sugerir técnicas e ferramentas viáveis para serem usadas durante o processo de edição da ontologia em evolução.
- Apoiar a detecção de mudanças - Detectar a(s) mudança(s) ocorrida(s) na ontologia não é tarefa trivial. Tal atividade demanda maior esforço pois necessita da análise aprofundada da ontologia. A detecção de mudanças é uma tarefa imprescindível para o processo de evolução pois é através dela que podem ser identificadas as inconsistências da nova versão da ontologia, sejam elas lógicas (relacionadas aos axiomas da ontologia), ou estruturais (relacionadas à organização estrutural das entidades da ontologia). Desse modo, apoiar a detecção de mudanças em ontologias Biomédicas em evolução é a principal finalidade desse objetivo específico.
- Fornecer aporte ao usuário no processo de tratamento de inconsistências – Quando uma ontologia passa por mudanças é comum que ocorram inconsistências em sua estrutura e nos artefatos que dela dependem. A identificação e resolução dessas inconsistências é um processo importante pois permite evitar que erros causem prejuízos ao usuários da ontologia. Assim, por meio desse objetivo específico pretende-se fornecer suporte para tal identificação e resolução de inconsistências.
- Acompanhar o processo de validação da ontologia – A consistência da ontologia em evolução precisa ser verificada antes que a mesma seja disponibilizada para utilização. Dessa forma, este objetivo específico possui como finalidade sugerir meios de organizar e executar o processo de validação da ontologia.

### **1.3 Organização do documento**

Os próximos capítulos deste trabalho apresentam, respectivamente, sua fundamentação teórica, onde são explanados temas relativos a ontologias, tipos de ontologias, ontologias Biomédicas, Evolução de ontologias e suas atividades mais relevantes como definição dos motivos de evolução, implementação e detecção de mudanças. O capítulo 3 aborda trabalhos relativos à evolução de ontologias e relaciona-os ao tema em investigação neste trabalho de modo a demonstrar pontos com maior grau de interesse pela comunidade de pesquisadores acerca do tema em questão ou áreas afins bem como sua aplicabilidade.

No capítulo 4 estão dispostos os materiais e métodos utilizados para a construção do *framework* proposto, que é detalhadamente explicado no capítulo 5. Um estudo de caso para demonstrar o funcionamento do *framework* construído foi explanado no capítulo 6. O capítulo 7 apresenta um relatório sumarizando a contribuição desse trabalho para a área em investigação por meio de curtas análises dos seus pontos mais relevantes. Nessa mesma seção estão descritos os trabalhos futuros do presente trabalho.

## Capítulo 2

### 2 Fundamentação Teórica

#### 2.1 Representação do Conhecimento

Para Sowa, Representação do Conhecimento é a utilização da lógica e das ontologias na construção de padrões computáveis para algum domínio específico (SOWA, 2000). Segundo Bueno (2005), Representação do Conhecimento consiste na especificação dos termos pelos quais o mundo é visto; uma forma de encontrar a abstração considerada correta para que alguém discorra sobre ela e a torne aplicável para a resolução de problemas. A autora completa ainda que diversas teorias da Inteligência Artificial (IA) que representam domínios específicos como medicina, física ou conceitos básicos como tempo e ação são ontologias ou têm uma forte relação com a representação ontológica.

A Representação do Conhecimento em grande escala requer uma ontologia de uso geral para organizar e reunir a grande variedade de domínios específicos do conhecimento. Essa ontologia precisa dar suporte a uma ampla variedade de tipos de conhecimento e deve ser capaz, inicialmente, de manusear qualquer domínio (RUSSEL e NORVIG, 2013).

A organização de objetos em categorias é uma parte imprescindível da Representação do Conhecimento, pois ainda que a interação com o mundo ocorra por meio de objetos individuais, uma grande parte do raciocínio ocorre no nível de categorias. As categorias podem ser usadas também para realizar previsões sobre objetos tendo em vista que os mesmos encontram-se classificados (RUSSEL e NORVIG, 2013).

É possível se deduzir a presença de certos objetos por meio de percepções, deduzir a aceitabilidade de uma categoria a partir da percepção das propriedades desses objetos e depois utilizar as informações a respeito da categoria para fazer previsões sobre os objetos. As categorias servem para organizar determinada base de conhecimento hierarquicamente. As relações de subclasse permitem que as informações categorizadas sejam organizadas em uma taxonomia ou hierarquia taxonômica (RUSSEL e NORVIG, 2013).

Conforme Sowa (2000) o campo da Representação do Conhecimento é geralmente chamado de Representação do Conhecimento e Raciocínio. Segundo o autor, isso acontece porque os formalismos da Representação do Conhecimento são inúteis se não for possível se realizar algum raciocínio com eles.

## **2.2 Ontologias**

Desde o início dos anos 90, ontologias têm se tornado um tópico especial em pesquisa investigado por várias comunidades de pesquisadores em IA, dentre elas está Engenharia do Conhecimento, Processamento de Linguagem Natural e Representação do Conhecimento. Mais recentemente, a noção do termo ontologia é também difundido em campos como Integração de Informação Inteligente, Informação Recuperada na Internet e Gestão do Conhecimento (STUDER, BENJAMINS e FENSEL,1998). .

Conforme Pinto e Martins (2004), ontologias são elementos importantes em muitas áreas, incluindo Gerenciamento e Organização do Conhecimento, Comércio Eletrônico e Recuperação e Extração de Informações. ontologias são responsáveis por promover e facilitar a interoperabilidade entre sistemas de informação, processamento inteligente por agentes e compartilhamento de conhecimento entre sistemas. Elas fornecem um entendimento compartilhado e consensual entre uma comunidade e sistemas heterogêneos e generalizados.

A principal motivação das ontologias é o fato de que estas permitem o compartilhamento e reuso do conhecimento em computação (STUDER, BENJAMINS e FENSEL,1998).

### **2.2.1 Definições Filosóficas**

Conforme citado por Guarino (1998) a partir da literatura fundamentada nas bases filosóficas do termo ontologia, este vocábulo teve sua origem na filosofia, onde o mesmo é uma descrição sistemática da existência. Em IA, por sua vez, tal expressão é conceituada por Gruber (1993), como algo que existe e pode ser representado.

Segundo Studer, Benjamins e Fensel (1998), o termo originalmente provém da filosofia. Foi possível se chegar a tal conclusão por meio da utilidade dada por Aristóteles,

filósofo e cientista grego, à expressão ontologia, o qual utilizava-a para tentar classificar coisas do mundo.

Guarino (1998), considera ainda que, no sentido filosófico, é possível referir-se a ontologias como um sistema particular de contagem de categorias para uma visão determinada do mundo. Assim, este sistema é independente de uma linguagem específica. Como exemplo, o autor cita o conceito de ontologia para Aristóteles, que é sempre o mesmo, independente da linguagem usada para descrevê-la.

### **2.2.2 Definições na esfera da Inteligência Artificial**

Conforme Guarino (1998), o termo ontologia deixou de ser apenas uma esfera filosófica do passado e tem ganhado um papel específico no campo da Inteligência Artificial, Linguística Computacional e Teoria de Banco de Dados. Para o autor, o termo tem sido reconhecido em áreas de pesquisa tão diversas quanto Engenharia do Conhecimento, Representação do Conhecimento, Modelagem Qualitativa, Engenharia da Linguagem, *Design* de Banco de Dados, Modelagem da Informação, Integração da Informação, Análise Orientada a Objetos, Recuperação e Extração da Informação, Organização e Gestão do Conhecimento, *Design* de Sistemas Baseados em Agentes. Guarino (1998) salienta ainda que as áreas onde as ontologias estão sendo aplicadas atualmente são diferentes das anteriormente citadas, elas incluem: Integração Empresarial, Padronização do Conhecimento do Produto, Comércio Eletrônico, Sistemas de Informação Geográfica, Jurídica e Biológica.

De acordo com Studer, Benjamins e Fensel (1998), a IA lida com o raciocínio sobre modelos do mundo. Dessa forma, é comum pesquisadores em IA adotarem o termo ontologia para descrever o que pode ser computacionalmente representado do mundo em um programa. O autor completa definindo ontologia como uma especificação formal, explícita de uma conceitualização compartilhada, onde conceitualização é um modelo abstrato de algum fenômeno no mundo por ter identificado os conceitos relevantes desse fenômeno. Explícito significa que os tipos de conceitos usados, bem como suas restrições são definidos de maneira explícita. Formal, se refere ao fato de que uma ontologia pode ser entendida por máquina. Compartilhada, por sua vez, reflete a noção de que uma ontologia capta do conhecimento consensual, ou seja, ela não é fechada para algo individual, mas sim aceita por um grupo.

Studer, Benjamins e Fensel (1998) cita ainda que ontologias são ótimas alternativas para facilitar a comunicação interpessoal em organizações, pois segundo o autor, elas fornecem termos, bem como seus significados, relações e restrições, entre outros objetos, e durante o processo de comunicação, todos os participantes devem colaborar com essas definições.

Segundo Guarino (1998), em alguns casos o termo ontologia é apenas um nome “elegante” que denota o resultado de atividades habituais como análise conceitual e modelagem de domínio, executado por meio de metodologias padronizadas. Em diversos casos, no entanto, as ontologias apresentam suas próprias metodologias e especificidades arquiteturas. Na esfera metodológica, a principal peculiaridade das ontologias é a adoção de uma abordagem altamente interdisciplinar onde a filosofia e a linguística têm um papel fundamental na análise da estrutura de determinada realidade em um alto nível de generalidade, e na elaboração de um vocabulário claro e rigoroso. Já na esfera arquitetural, o aspecto mais pertinente é a centralidade do papel que uma ontologia possui em um sistema de informação, quando conduzida para a perspectiva de sistemas de informação dirigidos por ontologias (GUARINO, 1998).

Comumente a Representação do Conhecimento tem concentrado muito esforço na representação de taxonomias, isso explica o fato de técnicas e linguagens de Representação do Conhecimento estarem sendo usadas no processo de implementação de ontologias. A maioria das técnicas e linguagens de Representação do Conhecimento permitem representar classes, objetos, conceitos, atributos, relações e instâncias. *Is-a* é um tipo de relação que permite formar hierarquias entre as entidades de uma ontologia (STUDER, BENJAMINS e FENSEL, 1998). Em uma relação *is-a* uma classe A é uma subclasse de B se cada instância de B também é uma instância de A. Outra forma de entender relações taxonômicas é enxergar uma subclasse como tipo de uma classe. Por exemplo, uma subclasse de uma classe representa um conceito que é um tipo do conceito que a superclasse representa (NOY e MCGUINNESS, 2001).

Contudo, ontologias são mais que taxonomias tendo em vista que ambas se diferem em dois aspectos: ontologias possuem uma estrutura interna mais rica e refletem algum conhecimento consensual. ontologias incluem todos as restrições relevantes entre classes,

valores de atributos, instancias e relações, ou seja, incluem axiomas. Tais restrições tornam fundamental o uso de relações *part-of* em alguns domínios (STUDER, BENJAMINS e FENSEL, 1998). A relação *part-of* relaciona uma entidade e seus componentes, e tem a intenção de ser transitiva (AITKEN, WEBBER e BARD, 2004).

A transitividade é um requisito importante para uma relação entre uma parte e sua totalidade. Em relações transitivas, se *A part-of B* e *B part-of C*, então *A part-of C*, ou seja, se relacionarmos o indivíduo A ao B, que por sua vez está relacionado a um terceiro indivíduo C, significa dizer que o indivíduo A também estará relacionado ao C (W3C, 2004).

### 2.2.3 Conceitualização

Guarino (1998) adotou o termo conceitualização para referir-se ao termo ontologia no âmbito filosófico. Para o autor, duas ontologias podem ser divergentes no vocabulário usado (usando palavras inglesas ou italianas, por exemplo) enquanto compartilham a mesma conceitualização (GUARINO, 1998). Studer, Benjamins e Fensel (1998) definem conceitualização como um modelo abstrato de algum fenômeno.

Para Guarino (1998), a noção de conceitualização citada anteriormente exige uma formalização adequada, uma vez que pode causar dúvidas. Genesereth e Nilsson (1987) embasaram o conceito dado por Guarino (1998) para o termo conceitualização, definido por este como uma estrutura  $\langle D, R \rangle$ , onde D representa o domínio e R é um conjunto ou relações relevantes presentes neste domínio. Tal definição de conceitualização foi utilizada por Gruber (1995) para definir uma ontologia como uma especificação de uma conceitualização.

Após a definição de conceitualização anteriormente dada por Guarino (1998), o mesmo considera que agora é possível descrever mais claramente o papel de uma ontologia, a qual é apresentada pelo autor como um conjunto de axiomas lógicos construídos para descrever o significado pretendido em um vocabulário. Dada uma linguagem L com o compromisso ontológico K, uma ontologia para L é um conjunto de axiomas projetados de tal modo que o conjunto de modelos se aproxima ao máximo do modelo pretendido de L de acordo com K.

Gruber (1993), assim como Guarino (1998) já utilizava o conceito de conceitualização. Esse conceito foi especificado por ele, a partir do embasamento teórico



obtido em Genesereth e Nilsson (1987), como os objetos, conceitos e demais entidades adotadas para representar alguma área de interesse e os relacionamentos mantidos entre essas entidades. O autor completa que conceitualização é uma visão abstrata e simplificada do mundo que se deseja representar por algum motivo. Gruber (1993) ressalta ainda que toda base de conhecimento, sistema baseado em conhecimento ou agente de nível de conhecimento está envolvido em alguma conceitualização, de forma explícita ou implícita.

Para Gruber (1993), uma ontologia é uma especificação explícita de uma conceitualização. Conforme citado por Guarino (1998) a partir da literatura fundamentada nas bases filosóficas do termo ontologia, este vocábulo teve sua origem na filosofia, onde o mesmo é uma descrição sistematizada da existência. Em IA, por sua vez, tal expressão é conceituada por Gruber (1993), como algo que existe e pode ser representado.

Quando o conhecimento de um domínio é representado em um formalismo declarativo, o conjunto de objetos que podem ser representados são chamados de universo do discurso. Tal conjunto de objetos, bem como os relacionamentos descritíveis entre eles, são refletidos no vocabulário representacional com o qual um sistema baseado em conhecimento representa o conhecimento propriamente dito. Dessa forma, no contexto de IA, pode-se descrever a ontologia de um programa pela definição de um conjunto de termos de representação. Nessas ontologias, definições associadas a nomes de entidades no universo do discurso como: classes, relações, funções entre outros objetos, com textos legíveis por humanos descrevendo o que os nomes significam e os axiomas formais que restringem a interpretação e melhor uso desses termos. Formalmente, uma ontologia é uma padronização de uma teoria lógica (GRUBER, 1993)

#### **2.2.4 Critérios para projeto de ontologias**

Segundo Gruber (1993), ontologias formais devem ser projetadas. A partir do momento em que se escolhe como representar algo em uma ontologia, realiza-se, ao mesmo tempo, o projeto de decisões. Para acompanhar e evoluir o projeto de ontologias, Gruber (1993), defende que são necessários critérios objetivos fundados sobre o propósito do artefato resultante em vez de baseados em noções prioritárias de naturalidade ou verdade. Seguindo esse pensamento, o autor propôs um conjunto de critérios para projeto de ontologias cuja finalidade é compartilhar conhecimento e interoperabilidade entre programas baseados em

uma conceitualização compartilhada. O conjunto de critérios definidos por Gruber (1993) e seus respectivos significados podem ser observados adiante.

#### **2.2.4.1 Clareza**

Em uma ontologia deve-se transmitir com efeito o propósito ao qual pretende-se chegar com os termos nela definidos. Tais definições devem ser objetivas. Contrapondo-se à motivação para definir determinado conceito que pode surgir a partir de um contexto social ou de atributos computacionais, a definição dos termos de uma ontologia deve ser independente desses requisitos. A partir da utilização do formalismo é possível se chegar a esse fim. Quando uma definição pode ser declarada por meio de axiomas lógicos, ela deve ser formalizada. Uma definição completa em vez de uma definição parcial é sempre preferível. Todas as definições devem ser documentadas em linguagem natural (GRUBER, 1993).

#### **2.2.4.2 Coerência**

Uma ontologia deve ser coerente no sentido de ser capaz de aprovar inferências que sejam consistentes com as definições. No mínimo, os axiomas devem ser logicamente consistentes. A coerência também deve ser aplicada para conceitos definidos informalmente. Caso uma sentença inferida por meio de um axioma contradiz uma definição ou exemplo fornecido de modo informal, então a ontologia é incoerente (GRUBER, 1993).

#### **2.2.4.3 Extensibilidade**

Uma ontologia deve ser projetada para antecipar o uso de um vocabulário compartilhado. Ela deve ofertar uma fundamentação conceitual para uma variedade de tarefas antecipadas e a representação deve ser elaborada de modo a estender e especializar a ontologia monotonicamente. Em outras palavras, deve ser capaz de definir novos termos para usos especiais no vocabulário existente, de modo que não haja a necessidade de revisão das definições nela constantes (GRUBER, 1993).

#### **2.2.4.4 Mínima codificação parcial**

A conceitualização deve ser especificada em um nível de conhecimento sem depender de uma codificação simbólica particular. A codificação parcial ocorre quando uma representação é realizada puramente por conveniência de notação ou implementação. Gruber

(1993) explica que a codificação parcial deve ser reduzida, porque agentes de compartilhamento de conhecimento podem ser implementados em diferentes sistemas e estilos de representação.

#### **2.2.4.5 Mínimo compromisso ontológico**

O compromisso ontológico das ontologias deve ser suficiente para suportar as atividades de compartilhamento de conhecimento pretendidas. Uma ontologia deve ter o mínimo de exigências possível acerca da área de modelagem, permitindo que a ontologia esteja livre para ser estendida e instanciada conforme necessário (GRUBER, 1993).

### **2.3 Ontologias Biomédicas**

ontologias são artefatos importantes para bioinformática e para as ciências da vida (HORRIDGE et al., 2014). Geralmente, ontologias pertencentes a essas áreas fornecem um vocabulário harmonizado descrevendo e estruturando um domínio específico de interesse, por exemplo, funções moleculares de proteínas ou a anatomia de determinadas espécies. O vocabulário mencionado consiste de conceitos, que são tradicionalmente estruturados em árvores ou em grafos acíclicos onde os nós de conceitos estão interconectados por relacionamentos do tipo *is-a* e *part-of*, que são os tipos mais comuns de relacionamentos em ontologias Biomédicas. Objetos biológicos, como genes e proteínas, podem ser semanticamente e uniformemente descritos ou anotados em ontologias pela sua associação com seu respectivo conceito. Por exemplo, proteínas estão associadas a conceitos da GO para descrever sua função proteica e especificar processos com os quais estão envolvidas (HARTUNG, KIRSTEN e RAHM, 2008).

Cui, Tao e Zhang (2016) explicam que pesquisas sobre ontologias Biomédicas bem como suas aplicações estão se estendendo. Cerca de 10.000 publicações indexadas pela PubMed<sup>8</sup>, nos últimos 10 anos envolvem o uso de ontologias, dessas, por volta de 1.500 contêm a palavra-chave "ontologia" no título. Ainda conforme os autores, a BioPortal<sup>9</sup>, do Centro Nacional de ontologia Biomédica, fornece acesso a quase 400 ontologias, constituídas por 5,3 milhões de classes utilizadas em uma variedade de aplicações de informática biomédica.

<sup>8</sup><https://www.ncbi.nlm.nih.gov/pubmed/>

<sup>9</sup><https://bioportal.bioontology.org/>

Um papel importante das ontologias Biomédicas é ser uma fonte de vocabulário, ou seja, uma lista de nomes para entidades representadas nestas ontologias. Detalhadamente, colecionar nomes é o papel de uma terminologia, não de uma ontologia. Cabe às linguagens de manipulação de ontologias como *Web ontology Language* (OWL), tratar nomes como *labels* ou *annotations* (formas de representação terminológica de uma ontologia em OWL). O componente terminológico de ontologias Biomédicas é um recurso importante para sistemas de processamento de linguagem natural e tarefas de suporte ao gerenciamento de conhecimento tais como anotação de recursos, recuperação de informações, acesso à informação e mapeamento por meio dos recursos. Contudo, a estrutura dos nomes das entidades presentes em ontologias Biomédicas dá suporte a apenas parte do léxico do domínio, principalmente para ontologias representadas em idiomas diferentes do inglês, e fornece apenas uma base para o gerenciamento da variação terminológica (BODENREIDER, 2008).

ontologias Biomédicas são vistas como elementos importantes de obtenção da interoperabilidade semântica e troca de informações em biomedicina, juntamente com padrões de mensagens e modelos de informação clínica. ontologias são pontos principais em modelos de diretrizes clínicas e desempenham papel muito importante na integração de dados heterogeneos provenientes de fontes divergentes, que é uma questão de pesquisa com nível considerável de criticidade (BODENREIDER, 2008).

Atualmente as ontologias têm sido usadas para facilitar a integração de dados, habilitar a descoberta do conhecimento e gerenciar grandes volumes de dados do domínio biomédico. Na Tabela 1 podem ser observados os tamanhos, em termos de número de conceitos e relacionamentos, das três ontologias mais conhecidas no domínio biomédico: *Standardized Nomenclature of Medicine - Clinical Terms* (SNOMED CT), *Foundational Model of Anatomy* (FMA) e GO.

Conforme Cui, Tao e Zhang (2016) a SNOMED CT é a maior terminologia clínica do mundo. A ontologia foi desenvolvida pela *International Health Terminology Standard Development Organization* (IHTSDO) e fornece ampla cobertura de medicina clínica, incluindo achados, doenças e procedimentos para uso em registros médicos eletrônicos. Na SNOMED CT os conceitos estão relacionados hierarquicamente dentro de cada sub-hierarquia, por exemplo: “amostra de tecido do coração” *is-a* “amostra de tecido”, que

significa que “amostra de tecido do coração” é “amostra de tecido”, tendo em vista que a relação *is-a* tem a interpretação: B *is-a* A, que significa que todos os B são A's (no entanto, às vezes é considerado como *instance of* e, sob essa interpretação, *is-a* não é transitiva ou seja, caso B possua um filho C, este não terá nenhuma relação com A) (AITKEN, WEBBER e BARD, 2004).

Tabela 1. Estatísticas de três importantes ontologias Biomédicas.

<b>Sistema ontológico</b>	<b>Conceitos</b>	<b>Relacionamentos</b>
SNOMED CT	>340.000	>440.000
FMA	>83.000	>1.000,000
GO/ <i>Biological Process</i>	>25.000	>47.000
GO/ <i>Molecular Function</i>	>10.000	>11.000
GO/ <i>Cellular Component</i>	>3.000	>5.000

Fonte. Adaptado de Cui, Tao e Zhang (2016).

Com base na definição dada por Rosse e Mejino Jr (2003) para a FMA, Cui, Tao e Zhang (2016) defendem que essa ontologia é tanto uma teoria da anatomia humana quanto um instrumento computacional. Como uma teoria, oferece uma estrutura unificada para a natureza das diversas entidades que compõem a estrutura corporal dos organismos biológicos, bem como suas relações. Como um artefato computacional, é uma representação formal desta teoria, adequada para manipulação de máquinas. O modelo subjacente ao FMA é uma representação baseada em *frames*.

Segundo Lassila e MsGuinness (2001) um *frame* representa conceitos e objetos. Junto ao *frame* está uma coleção de atributos possivelmente contendo tipos (ou restrições de valor) e possivelmente inicializado com valores. Na FMA estes *frames* possuem conceitos que incluem anatomia canônica macroscópica, microscópica e sub-celular. Segundo Clarkson (2016) a FMA é uma das maiores ontologias Biomédicas, com mais de 104.000 classes e 14 tipos de relações.

A GO é um conjunto de três ontologias que descrevem atributos de produtos genéticos em três domínios da biologia molecular que não se sobrepõem: Componente Celular, que são as partes de uma célula ou seu ambiente extracelular; Função Molecular, que diz respeito às atividades elementares de um produto de gene a nível molecular, como a ligação ou a catálise; e Processo Biológico, operações ou conjuntos de eventos moleculares

com início e fim definidos, pertinentes ao funcionamento de unidades vivas integradas (células, tecidos, órgãos e organismos). Dentro de cada ontologia, os termos possuem definições de texto e identificadores exclusivos (CUI, TAO e ZHANG, 2016) .

Relações *is-a* e *part-of* são usadas para fazer ligações entre os termos das ontologias GO. Dessa forma, ontologias GO são grafos acíclicos dirigidos. Em tais grafos, uma entidade pode ter vários links para os pais via relações *is-a* e *part-of* (AITKEN, WEBBER e BARD, 2004).

Além dos sistemas ontológicos suportados por agências governamentais e organizações de saúde, tem-se o BioPortal<sup>10</sup>, maior repositório de ontologias criadas pela comunidade de pesquisa biomédica, pertencente ao *National Center for Biomedical ontology* (NCBO)<sup>11</sup>. Com o BioPortal também é possível realizar-se a obtenção de recursos para mapeamento, anotação, edição e visualização de ontologias (CUI, TAO E ZHANG, 2016).

## 2.4 Evolução de ontologias

Devido à sua natureza dinâmica, o conhecimento de qualquer domínio faz as ontologias mudarem ao longo do tempo. A mudança do conhecimento de uma ontologia pode estar relacionada a alterações do domínio, especificação, conceitualização ou à combinação desses motivos. Algumas dessas mudanças são realizadas para introduzir novos conceitos e/ou remover conceitos desatualizados bem como fazer alterações nas estruturas e significados desses conceitos. Uma alteração em uma ontologia pode ser proveniente de um especialista em conhecimentos de domínio, uma requisição de um usuário da ontologia ou uma mudança na área de aplicação (JAVED, ABGAZ E PAHL, 2009).

Conforme Klein (2004), as ontologias não são permanentes nem estáveis. ontologias mudam similarmente às visões sobre o mundo de modo a acompanhá-las (por exemplo, os principais problemas políticos). As ontologias também mudam porque o conhecimento sobre determinado domínio passou por processo de melhoramento (por exemplo, a descoberta do novo efeito de determinado medicamento sobre uma doença) ou porque o mundo muda fisicamente (por exemplo, a divisão territorial de determinado país). É importante dar ênfase às mudanças em ontologias porque elas têm efeito sobre como os dados deveriam ser

<sup>10</sup><http://bioportal.bioontology.org/>

<sup>11</sup><https://www.bioontology.org/about-ncbo>

interpretados e manuseados. Contudo, tais efeitos não podem ser identificados apenas observando as ontologias discricionariamente pois para isso é preciso saber as razões pelas quais foi necessário aplicar determinada mudança na ontologia bem como a finalidade pela qual a ontologia é usada.

Evolução de ontologias é a adaptação oportuna de uma ontologia aos seus requisitos de domínio, às tendências das instâncias da ontologia e aos padrões de uso das aplicações baseadas na ontologia em evolução, bem como a propagação consistente dessas mudanças em elementos dependentes. Uma ontologia poderá ser considerada consistente quando todas as restrições definidas em sua estrutura e conteúdo forem satisfeitas. A necessidade de definir o domínio e o alcance de cada relação de uma ontologia é um exemplo de restrição estrutural. Restrições de conteúdo, por outro lado, estão relacionadas aos axiomas na ontologia (STOJANOVIC et al., 2002). A evolução de ontologias possui como objetivo garantir a semântica correta de todas as mudanças que devem ser aplicadas (STOJANOVIC et al., 2003).

A evolução de ontologias lida com o crescimento da ontologia bem como com a obtenção e organização de novas informações que devem ser incorporadas ao seu conteúdo (KHATTAK, LATIF e LEE, 2013).

A necessidade de implementar mudanças em uma ontologia pode advir de diversos motivos que incluem: a descoberta de informações (que podem ser dados de instâncias, outra ontologia, uma nova observação, etc) mudança no foco ou no ponto de vista da conceitualização representada, incorporação de novas informações fornecidas por alguma fonte externa, mudanças no domínio da ontologia, descoberta de um padrão problemático durante o processo de modelagem, necessidade de comunicação entre fontes heterogêneas de informação, a fusão de informações de diferentes ontologias entre outros (FLOURIS et al., 2008).

Klein (2004) define mudanças em ontologias como uma ação aplicada a uma ontologia que fornece como resultado uma versão diferente da versão original dessa ontologia. Baseando-se nos diferentes níveis de interpretação de uma ontologia é útil fazer distinções entre os diferentes tipos de alterações que podem ser feitas em seu conteúdo, dessa forma, definiu-se três tipos de mudanças ontológicas: mudança conceitual, que é a mudança

aplicada a uma conceitualização; mudança de especificação, definida como uma mudança em uma especificação de uma conceitualização e; mudança de representação, ou seja, uma mudança na representação da especificação de uma conceitualização. Uma mudança na especificação de uma ontologia não é necessariamente uma mudança em sua conceitualização e mudanças na especificação de uma ontologia não são, por definição, mudanças ontológicas. Por exemplo, existem mudanças na especificação de um conceito que não expressa o sentido de que houve mudança no conceito em si.

Por ser um tipo de KOS complexo e com entidades intimamente conectadas, a evolução de uma ontologia pode ser realizada de diversas formas. Por exemplo, após a remoção de um conceito, os subconceitos relacionados a ele podem ser removidos ou mantidos. Caso opte-se por manter tais subconceitos, eles devem ser reconectados a um novo pai, que deverá ser o pai do conceito deletado ou o conceito raiz, com vistas a preservar a estrutura e manter a coerência da ontologia. Além disso, o usuário pode querer manter apenas alguns conceitos. Desse modo, o número de soluções possíveis para uma mudança é proporcional à sua complexidade (STOJANOVIC et al., 2003).

Segundo Stojanovic et al. (2003) cada mudança requerida no processo de evolução atende aos requisitos de apenas alguns usuários. Considerando que as necessidades de um usuário não podem ser antecipadas é impossível determinar exatamente como resolver determinada requisição de mudança. Assim, para ser eficiente, um sistema de evolução de ontologias precisa estar apto para tratar dois problemas: como o usuário pode especificar sua requisição de mudança e como tal requisição pode ser realizada. No primeiro problema é requisitado um método para expressar as necessidades do usuário de forma exata, clara e declarativa. Esse método contradiz algumas abordagens existentes onde o usuário escolhe a mudança a ser requerida a partir de um conjunto de mudanças predefinidas, o que não cobre todas as suas necessidades. O segundo problema diz respeito à limitação dos sistemas quanto às formas de resolver determinado requerimento de mudança. O sistema de apoio à evolução deve ser capaz de calcular e fornecer ao usuário todos os caminhos possíveis para satisfazer seu requerimento.

O processo de evolução de ontologias não é trivial, devido à variedade de fontes e consequências das mudanças, o mesmo não pode ser realizado manualmente. Desse modo, tal



processo deve ser realizado com o apoio do sistema de gerenciamento de ontologias. Um aspecto importante no processo de evolução é garantir a consistência da ontologia quando ocorrerem mudanças em seu conteúdo, considerando também a semântica dessas mudanças. A formalização da semântica da mudança requer uma definição do modelo de ontologia juntamente com suas operações de mudança como adições, deleções e operações de mover, as condições de consistência e as regras para impor essas condições (HAASE e STOJANOVIC, 2005).

Tendo em vista que uma ontologia evolui para um novo estado, suas dependências e serviços podem se tornar inválidos. Em consequência disso, soluções para o gerenciamento de mudanças em ontologias devem responder a questões como: “como manter todas as mudanças de um modo consistente e coerente?”. Durante o processo de enriquecimento ontológico, ou seja, melhoramento do conteúdo de determinada ontologia, as mudanças realizadas fazem-na evoluir de um estado consistente para outro sem preservar informações sobre o estado anterior. Esse fato dificulta a referência ao estado anterior a menos que as mudanças sejam preservadas. Além disso, informar sobre o histórico de mudanças se torna uma tarefa crítica quando um usuário sem autorização realiza mudanças, ou quando um engenheiro de ontologias quer reverter as mudanças. A estrutura de armazenamento para essas informações também é crucial para a recuperação efetiva de informações (KHATTAK, LATIF e LEE, 2013).

Uma mudança aplicada em uma parte da ontologia pode gerar incoerências em outras partes da mesma ontologia, nas instâncias baseadas na ontologia, bem como em ontologias e aplicações dependentes. As causas e consequências das mudanças aplicadas em uma ontologia torna a evolução um procedimento muito complexo, desse modo a evolução de ontologias deve ser considerada como um processo organizacional e técnico (STOJANOVIC et al., 2002). A evolução de ontologias é muito importante atualmente. A principal razão para isso é o crescimento da sua utilização e o aumento dos custos associados à sua adaptação para refletirem as mudanças requeridas. (STOJANOVIC, 2004).

O gerenciamento de mudanças em ontologias lida com o problema de decidir que modificações devem ser realizadas como resposta a requisições por mudanças. Esse mecanismo garante que as mudanças sejam aplicadas na ontologia e que a mesma permaneça

em estado consistente. Ele ainda é responsável por tratar quatro questões: evolução, versionamento, integração e fusão de ontologias. No versionamento uma ontologia em evolução é manuseada por meio da criação e gerenciamento de diferentes versões. Evolução e versionamento de ontologias devem ser agregados em uma abordagem holística para gestão das mudanças em ontologias bem como seus efeitos (KHATTAK, LATIF e LEE, 2013).

A integração é um processo de composição de ontologias utilizando informações encontradas em duas ou mais ontologias que cobrem o mesmo domínio da ontologia em processo de integração. Similarmente à integração, a fusão de ontologias é o processo de compor uma ontologia com informações encontradas em duas ou mais ontologias cobrindo domínios sobrepostos ou idênticos (KHATTAK, LATIF e LEE, 2013).

O processo de evolução de ontologias segue duas variantes: população e enriquecimento de ontologias. Na primeira variante, novas instâncias pertencentes a conceitos codificados anteriormente podem ser adicionadas ou instâncias existentes podem ser alteradas. O resultado desse procedimento é a modificação do A-Box da ontologia que passa a refletir novas realidades. No processo de enriquecimento de ontologias, novos conceitos de domínio, propriedades ou restrições são adicionadas ou entidades existentes são alteradas. Tais mudanças resultam em alterações na T-Box da ontologia (KHATTAK, LATIF e LEE, 2013).

Segundo Horrocks, Patel-schneider e Welty (2007), T-Box e A-Box são recursos de especificação de bases de conhecimento. T-Box contém operações usadas para construir uma terminologia. A forma básica de declarações em uma T-Box, é a definição de um conceito. Por exemplo, uma mulher pode ser definida como uma pessoa do sexo feminino por meio da seguinte declaração:

$$\textit{Mulher} \equiv \textit{Pessoa} \cap \textit{Sexo Feminino}$$

A-Box contém conhecimento extencional sobre o domínio de interesse com asserções sobre indivíduos relacionados a esses conhecimentos. Essas asserções são geralmente chamadas de asserções associativas. Um exemplo pode ser observado adiante. Nele afirma-se que maria é uma pessoa do sexo feminino (HORROCKS, PATEL-SCHNEIDER e WELTY, 2007).

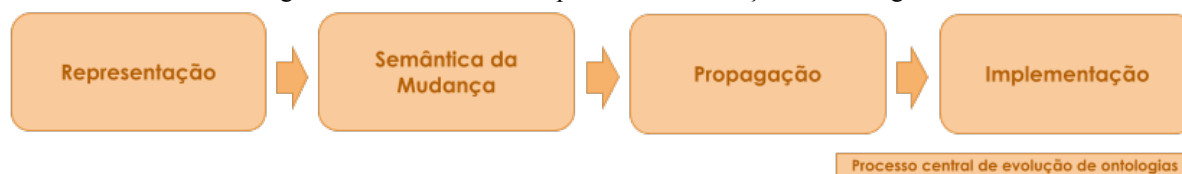
$Sexo\ Feminino \cap Pessoa(MARIA)$

Com base no conceito dado anteriormente para mulher, pode-se derivar que MARIA é uma instância do conceito Mulher. Esse tipo de afirmação pode ser classificada como asserção de conceito. Similarmente, no próximo exemplo, onde afirma-se que MARIA tem BENJAMIN como filho, tem-se uma asserção atributiva, onde um papel é atribuído ao indivíduo MARIA (HORROCKS, PATEL-SCHNEIDER e WELTY, 2007).

$temFilho(MARIA, BENJAMIN)$

Stojanovic (2004) cita aspectos cruciais para um processo de evolução de ontologias onde a sua consistência e dos seus elementos dependentes seja mantida. Na Figura 1, organizados como fases do processo de evolução, podem ser observados esses aspectos.

Figura 1. Fases cruciais do processo de evolução de ontologias.



Fonte. Adaptado de Stojanovic (2004).

Conforme Stojanovic (2002), frequentemente a finalidade das mudanças em ontologias pode ser expressada em um nível mais alto. Por exemplo, o usuário pode precisar gerar um conceito comum D a partir de dois conceitos C1 e C2. Isso pode ser alcançado por meio da aplicação sucessiva de uma lista de mudanças elementares, que são mudanças que modificam apenas uma entidade do modelo ontológico, por exemplo, “AdicionarConceito D”, “DeletarSubConceitoDa relação de C1 para seu pai atual”, “AdicionarSubConceitoDa relação de C1 para D” , “DeletarSubConceitoDa relação de C2 para seu pai atual” e “AdicionarSubConceitoDa relação de C2 para D”.

Contudo, essa representação pode ter como desvantagens a incompatibilidade entre a intenção da requisição e a forma com a qual a requisição é alcançada. Isso acontece quando se deseja criar um superconceito a partir de dois conceitos, porém, um precisa dissolver esta operação em cinco passos separados, tornando todo o processo propenso a erros. Outra desvantagem da representação da mudança em um nível mais alto, é a possibilidade de realização de mudanças desnecessárias caso essas mudanças forem aplicadas uma de cada

vez, por exemplo, remover um subconceito de  $C1$ , pode ocasionar mudanças nas instâncias de propriedades que devem ser revertidas quando for atribuído o subconceito da relação de  $C1$  para  $D$ .

De modo a evitar tais problemas, deve ser possível expressar mudanças a um nível inferior ao demonstrado anteriormente, de modo a tornar a mudança diretamente visível. Isso conduz às mudanças compostas que representam um grupo de mudanças elementares aplicadas ao mesmo tempo (STOJANOVIC, 2002). A título de exemplo, na Tabela 2 está representado um conjunto de mudanças compostas.

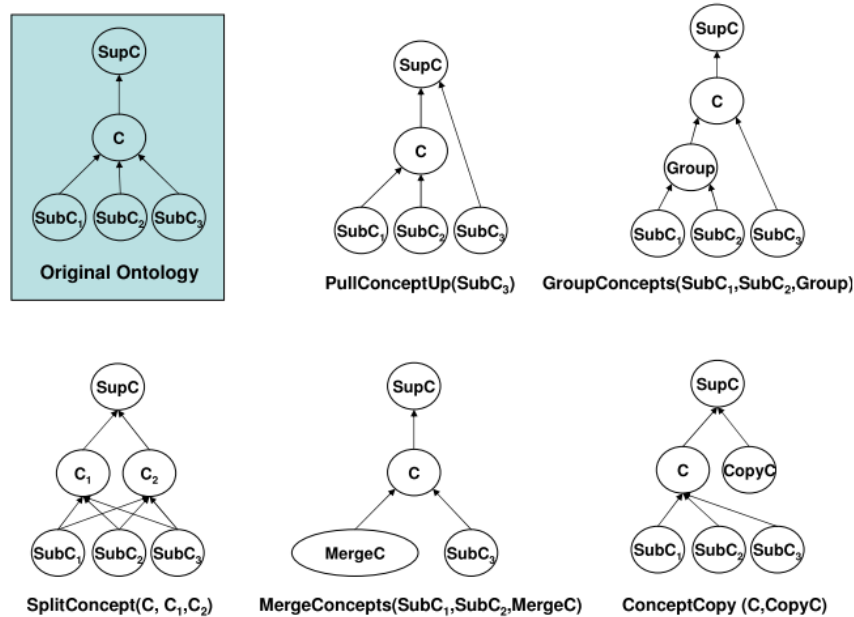
Tabela 2. Exemplos de mudanças compostas.

<b>Mudança Composta</b>	<b>Descrição</b>
Fusão de conceitos	Substituição de vários conceitos por apenas um e agregação de todas as suas propriedades e instâncias
Divisão de conceitos	Divisão de um conceito $c$ em dois conceitos $c1$ e $c2$ e distribuição de propriedades e instâncias entre eles
Agrupamento de conceitos	Criação de um superconceito comum $c$ para os conceitos $c1$ e $c2$ e transferência das propriedades comuns para o novo superconceito
Cópia de conceito	Duplicar um conceito $c$ com todas as suas propriedades e instâncias por meio da criação de um novo conceito adicionando a ele todos os pais de $c$
Generalização de conceito	Adicionar um novo conceito entre um determinado conceito $c$ e todos os seus conceitos pais
Trazer conceito para cima	Anexar um conceito $c$ para os pais de todos os seus conceitos pais
Puxar conceito para baixo	Anexar um conceito $c$ para os filhos de todos os seus conceitos pais exceto $c$
Extensão de herança	Adicionar um novo conceito $c$ e anexá-lo a um conceito $c1$ como seu pai e o conceito $c2$ como seu filho
Especialização de conceito	Adicionar um novo conceito entre um dado conceito $c$ bem como todos os seus filhos

Fonte. Adaptado de Stojanovic (2004)

A Figura 2 apresenta o funcionamento de algumas das mudanças compostas descritas na Tabela 2 quando aplicadas a conceitos pertencentes a uma hierarquia.

Figura 2. Exemplos de mudanças compostas relacionadas a uma hierarquia de conceitos.



Fonte. Stojanovic (2004)

Conforme Stojanovic (2004) a aplicação de mudanças elementares em uma ontologia pode induzir inconsistências em outras partes dessa ontologia. Essas inconsistências podem ser estruturais ou semânticas. As primeiras ocorrem quando as restrições do modelo da ontologia é invalidado por meio, por exemplo, do uso de entidades indefinidas. Inconsistências semânticas, por sua vez, ocorrem quando o sentido de determinada entidade é alterado devido às mudanças realizadas na ontologia.

Para resolver tais problemas é necessário realizar outras mudanças que podem levar à ocorrência de novas inconsistências tornando a tentativa de corrigir os problemas um processo repetitivo e oneroso. Quando uma ontologia possui muitos conceitos pode ser ainda mais difícil compreender totalmente as mudanças induzidas bem como as consequentes inconsistências. Para ajudar no melhor entendimento dos efeitos de cada alteração, existe a semântica da mudança, segunda fase do processo de evolução de ontologias criado por Stojanovic (2004), que permite a resolução de mudanças induzidas de um modo sistemático e contribui com máxima transparência fornecendo uma visão detalhada de cada mudança garantindo assim a consistência de toda a ontologia (STOJANOVIC, 2004).

ontologias frequentemente reusam e estendem outras ontologias. Desse modo, uma alteração aplicada a uma ontologia pode corromper as demais ontologias que dela dependem e consequentemente, todos os artefatos baseados nessas ontologias. Esse problema pode ser contornado por meio da aplicação recursiva do processo de evolução de ontologias aplicado à ontologia em processo de evolução a partir da qual surgiram os problemas. Contudo, isso também requer o uso de métodos de sincronização entre ontologias dependentes devido a sua independência. A propagação da mudança, terceira fase do processo de evolução de ontologias proposto por Stojanovic (2004), tem como objetivo permitir que todos os artefatos dependentes da ontologia em evolução estejam consistentes mesmo após a aplicação das mudanças.

A última fase do processo proposto por Stojanovic (2004) é chamada de implementação de mudanças e tem como papel informar o engenheiro de ontologias sobre todas as consequências de uma mudança que foi requerida, aplicar todas as mudanças requeridas e derivadas bem como manter uma linha do tempo das mudanças que foram realizadas.

Considerando que as tarefas realizadas em ontologias como gerenciamento, modificação, evolução e versionamento causam dúvidas relativas às suas definições, Stojanovic (2004) fez distinções entre as mesmas caracterizando cada uma como segue:

- Gerenciamento de ontologias é todo conjunto de métodos e técnicas necessárias para usar eficientemente múltiplas variantes de ontologias de origens possivelmente diferentes para tarefas distintas;
- A modificação de ontologias ocorre quando um sistema de gerenciamento permite mudanças para ontologia que está em uso sem considerar sua consistência;
- A evolução de ontologias é realizada quando o sistema de gerenciamento facilita a modificação da ontologia preservando uma consistência;
- O versionamento de ontologias acontece quando o sistema de gerenciamento permite manusear as mudanças de uma ontologia por meio da criação e gestão de diferentes versões da mesma.

O sucesso da aplicação de ontologias em ambientes não controlados, descentralizados e distribuídos requer suporte substancial para gerenciamento de mudanças em ontologias bem como para a evolução das mesmas (KLEIN e NOY, 2003). Dada uma ontologia  $O$  e suas duas versões:  $V_1$  e  $V_2$ , o suporte completo para o gerenciamento de mudanças no ambiente dessa ontologia inclui apoio à tarefa de transformação de dados, acesso aos dados, atualização da ontologia, raciocínio consistente e verificação e aprovação (KLEIN e NOY, 2003).

Para Stojanovic (2004), o processo de evolução de ontologias é importante porque o desenvolvimento de uma ontologia é dinâmico e a evolução de ontologias possui três desafios: complexidade, pois uma ontologia é um modelo rico e possui inter-relacionamentos que podem aumentar os efeitos das mudanças; dependências, pois uma ontologia pode reusar e estender outras ontologias; e distribuição física, devido ao fato do desenvolvimento de ontologias ser um processo descentralizado e colaborativo.

A complexidade da evolução da ontologia aumenta à medida que as ontologias crescem em tamanho, de modo que um processo estruturado de evolução da ontologia é necessário. Conforme sugerido por Haase e Stojanovic (2005), o processo começa com a captação de alterações. A próxima fase é a representação da mudança de modo formal e explícito. A semântica da fase de mudança é importante pois evita inconsistências ao computar mudanças adicionais que garantem a consistência da ontologia (HAASE e STOJANOVIC, 2005).

Na fase de propagação da mudança, todos os artefatos dependentes da ontologia em processo de evolução (instâncias de ontologia na Web, ontologias dependentes e aplicações que usam a ontologia alterada) são atualizados. Durante a fase de implementação de mudanças, são aplicadas as mudanças necessárias e as mudanças induzidas. Na fase de validação da mudança, o usuário avalia os resultados e reinicia o ciclo caso necessário (HAASE e STOJANOVIC, 2005).

Ao longo do seu ciclo de vida, uma ontologia pode ter diversas versões. Acompanhar as diferenças entre as versões de uma ontologia é uma tarefa necessária principalmente para processos de reuso (PINTO e MARTINS, 2004).

Alguns autores consideram o versionamento da ontologia uma variante mais forte da evolução ontológica. Considerando esse ponto de vista, a evolução da ontologia está

preocupada com a capacidade de mudar a ontologia sem perder dados ou a validade da ontologia, enquanto que o versionamento deve permitir o acesso a diferentes variantes da ontologia. Desse modo, enquanto a evolução da ontologia está preocupada com a validade da versão mais recente, o versionamento de ontologias lida com a validade, interoperabilidade e gerenciamento de todas as versões anteriores, incluindo a atual (FLOURIS et al., 2008).



## Capítulo 3

### 3 Trabalhos Relacionados

Os trabalhos relacionados à presente pesquisa foram selecionados com base em sua similaridade com a proposta deste trabalho cujo objetivo principal é oferecer um *framework* para apoio à evolução de ontologias Biomédicas. De modo a se analisar mais profundamente os trabalhos relacionados à área em investigação neste trabalho, foi realizada uma busca por temas relevantes que constituem o processo de evolução de ontologias de modo a se saber quais desses temas são mais abordados, o que permite determinar o nível de importância da aplicação dos mesmos em um processo evolutivo convencional. O uso reduzido de determinado tema permitiu que se identificasse quais áreas carecem de maior atenção por parte dos pesquisadores da área de evolução de ontologias.

Os temas utilizados no processo de análise foram definidos conforme a literatura pertinente sobre evolução de ontologias que julga como importantes para o processo evolutivo a representação e a semântica da mudança, a consistência da ontologia, a consideração quanto ao desenvolvimento ser ou não colaborativo, a verificação e validação da mudança, o versionamento e o mapeamento entre versões. Assim, os temas citados foram utilizados como critérios de análise dos trabalhos relacionados. Além desses temas utilizou-se ainda a verificação do suporte ou não à evolução de ontologias do domínio biomédico. Após o processo de análise, os trabalhos relacionados foram organizados na Tabela 3 levando em conta a abordagem ou não, em sua estrutura, dos temas selecionados como critérios de análise. A seguir, cada um dos trabalhos selecionados são sumariamente descritos.

Khattak, Latif e Lee (2013) (1) propõem um *framework* compreensivo e metodológico para apoiar a resolução de problemas relacionados ao gerenciamento de mudanças em ontologias em processo de evolução. Tal gerenciamento engloba problemas relacionados ao versionamento, procedência, consistência, recuperação, representação e visualização da mudança. O histórico de mudanças é o requisito central do *framework*. Esse trabalho tem como objetivo principal fornecer um mecanismo para rastrear temporariamente mudanças em

uma ontologia ao longo da sua vida útil. De maneira específica esse trabalho objetiva apoiar o gerenciamento, recuperação e visualização de mudanças para que se entenda o comportamento do processo evolutivo de uma ontologia. Para alcançar o objetivo do trabalho propôs-se um modo de manter e gerir mudanças. Para o primeiro foi criado um Histórico de Mudanças da ontologia (CHO, do inglês *Change History ontology*), um esquema para representar mudanças ocorridas em ontologias cuja finalidade é recuperar e rastrear mudanças com o reuso de construtores existentes e padrões de projeto de ontologias. Os elementos centrais do CHO são as classes *ontologyChange* e *ChangeSet*. A classe *ontologyChange* possui um subconjunto chamado *AtomicChange* que representa todas as classes, propriedades, instâncias e mudanças a nível de restrições atômicamente.

A classe *ChangeSet* faz a junção de todas as mudanças em um intervalo de tempo específico de forma coerente e gerencia essas mudanças organizando-as de maneira indexada ao tempo em que ocorreram. Para gerenciar eficientemente as mudanças criou-se um *Log* do Histórico de Mudanças (CHL, do inglês *Change History Log*). O CHL é similar a base de dados relacionais e utiliza uma técnica de registro para armazenar mudanças de ontologias. O registro de mudança ajuda na recuperação da ontologia para um estágio prévio em caso de mudanças não autorizadas, conflitos entre versões ou até mesmo inconsistências devido ao fechamento acidental do editor.

O *framework* proposto nesse trabalho é implementado como um *plug-in* para repositórios de ontologias, como *joseki*<sup>12</sup> e editores de ontologias como o *Protégé*, *software* de código aberto para apoio à edição de ontologias desenvolvido por pesquisadores da Universidade de *Stanford*, cujas principais características são: arquitetura expansível, fácil usabilidade e nível considerável de detalhamento (Mattos, Simões e Farias, 2010). Foi realizada uma comparação entre *plug-ins* existentes para avaliar sua acurácia, consistência e facilidade de compreensão. Os *plug-ins* utilizados na comparação foram: *ChangeCapturing*<sup>13</sup> utilizado no editor *NeOn Toolkit*<sup>14</sup>, *ChangesTab* e *Version Log Generator* do *Protégé*, *ChangeDetection*, aplicação *stand alone* e *ChangeTracer*, o *plug-in* desenvolvido nesse trabalho que deve ser usado na ferramenta *Protégé*. As comparações foram realizadas para avaliar a capacidade de captura das mudanças de cada *plug-in*. Para que se realizasse as

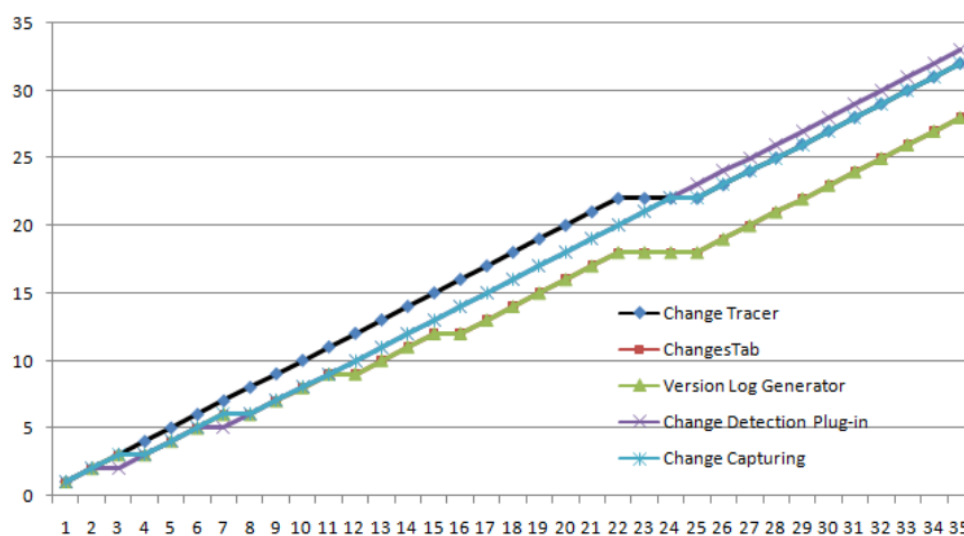
<sup>12</sup><https://sourceforge.net/p/joseki/wiki/Home/>

<sup>13</sup>[http://neon-toolkit.org/wiki/1.x/Change\\_Capturing.html](http://neon-toolkit.org/wiki/1.x/Change_Capturing.html)

<sup>14</sup><http://neon-toolkit.org/>

comparações, em uma ontologia foram aplicadas 35 mudanças a nível de hierarquia (10), classes (15) e propriedades (10). Depois foi aplicada uma equação para verificar o procedimento de recuperação de cada *plug-in*. Os resultados das comparações mostraram que o *framework* implementado no *plug-in ChangeTracer* teve alta acurácia e melhor cobertura. Após testes exastivos os algoritmos de recuperação mostraram consistência e acurácia. No gráfico da Figura 7 pode ser observado o desempenho de cada *plug-in*.

Figura 3. Resultado das comparações entre os *plug-ins* avaliados.



Fonte: Khattak, Latif e Lee (2013).

Foi apresentada como benefício do *framework* a possibilidade de reformular consultas para responder a requisições do usuário. Para reformular consultas o sistema solicita ao CHL um teste que verifica se o esquema de representação subjacente sofreu alterações. Caso sim a consulta é reformulada de modo a reconhecer as mudanças e serem executadas sobre a ontologia evoluída. Isso ajuda a evitar a quebra de consultas de um sistema que ainda não foi adaptado para a nova versão da ontologia. Outro benefício é o reconciliamento dos mapeamentos entre ontologias. As mudanças registradas pelo CHL podem ser utilizadas para reconciliar o mapeamento entre ontologias evoluída em vez de reiniciar o processo de mapeamento reduzindo assim o consumo de tempo. São ainda benefícios do *framework* proposto, o gerenciamento de mudanças e a recuperação de ontologias. O CHL registra todas as mudanças indexando-as ao tempo em que ocorreram, tal indexação ajuda no retorno da ontologia para um estado prévio consistente. O gerenciamento das mudanças pode ajudar um

novo usuário a entender as mudanças que foram realizadas. O CHL ajuda ainda a entender as mudanças de maneira semântica. Por meio do registro de mudanças também é possível a aplicação de algoritmos de Aprendizagem de Máquina (AM) para identificar padrões de mudança. O rastreamento temporal de mudanças de ontologias também é um benefício do *framework*. Por meio do CHL é possível implementar ferramentas de visualização de ontologias para exibir seus diferentes estados, o que pode ajudar no rastreo temporário de mudanças e melhor entendimento do comportamento da evolução da ontologia (KHATTAK, LATIF e LEE, 2013).

Haase *et al.* (2005) (2) apresentam um *framework* que combina métodos separados para manuseio de inconsistências em ontologias em evolução. Foram analisadas 4 abordagens para manusear inconsistências ocorridas em ontologias baseadas em Lógica de Descrição (DL do inglês, *Description Logic*). As abordagens foram comparadas por meio do uso de uma base formal comum que levou em consideração características como os requerimentos, condições para aplicação e diferentes cenários de uso de cada abordagem.

O *framework* proposto é baseado em noções formalmente definidas que incluem mudanças em ontologias e inconsistências em ontologias baseadas em DL. Os principais componentes do *framework* são: evolução consistente, que garante a consistência continuada pela aplicação de estratégias de evolução; reparação de inconsistências, onde as inconsistências da ontologia são reparadas; raciocínio com ontologias inconsistentes, onde ontologias inconsistentes devem ser capazes de retornar resultados significativos; raciocínio multiversões, no qual todas as versões prévias da ontologia devem ser consideradas durante o tratamento de inconsistências (HAASE *et al.*, 2005).

O principal benefício desse trabalho é a identificação de tipos comuns de problemas em ontologias em evolução. Após a comparação entre as abordagens foi possível concluir que as mesmas diferem tanto no uso de máquinas e na forma que lidam com ontologias inconsistentes, variando desde o diagnóstico e reparação de inconsistências, raciocínio na presença de inconsistências até o rastreo dessas inconsistências ao longo do processo de evolução da ontologia. Por meio das comparações realizadas foi possível detectar que diferentemente do que se pensou nas fases iniciais da análise, as abordagens não lidam apenas com ontologias inconsistentes mas também com aspectos como o tempo de desenvolvimento

ou de uso da ontologia e requisições de informações como a causa da inconsistência ou o histórico de mudanças da ontologia (HAASE *et al.*, 2005).

Reis *et al.* (2015) (3) definiram um framework completo e original baseado em heurísticas formais que dirigem a adaptação do mapeamento entre KOS nomeado DyKOSMap *framework*. A abordagem leva em conta a definição de mapeamentos, a evolução do KOS e as mudanças que podem ser aplicadas para mapeamentos. Esse estudo avalia experimentalmente a heurística e o *framework* proposto por meio de um estudo de caso real do domínio biomédico, usando o mapeamento oficial de vários KOSs biomédicos. Especificamente, neste trabalho foi proposto um conjunto de heurísticas formalmente definidas que dirigem o processo de adaptação do mapeamento entre ontologias. Esta abordagem visa prover a adequada evolução entre mudanças ocorridas em KOSs e padrões de adaptação do mapeamento.

Este trabalho apresenta como contribuições um novo *framework* para a adaptação do mapeamento entre KOSs. Foram propostos algoritmos de AM implementados no *framework*, o que introduz técnicas de manuseio de dois KOS em evolução; um conjunto de heurísticas implementadas no *framework* expressando condições adequadas para guiar decisões sobre a adaptação do mapeamento de acordo com a Ação de Adaptação do Mapeamento (MAA, do inglês, *Mappings Adaptation Action*); um conjunto de experimentos para mostrar a relevância de manter o mapeamento do *framework* proposto com heurísticas para a adaptação do mapeamento entre ontologias (REIS *et al.*, 2015).

Um estudo de caso é realizado para avaliar experimentalmente a proposta baseando-se no uso de várias versões de grandes KOSs biomédicos e seus conjuntos de mapeamentos associados. Os experimentos realizados possuem como objetivo analisar até que ponto a abordagem pode corresponder à evolução de mapeamentos do mundo real bem com examinar o comportamento geral produzido pelo *framework* e a adaptação obtida para cada conjunto de dados. Conforme o resultado das avaliações, o *framework* apresentou alta efetividade para todos os conjuntos de dados na adaptação do mapeamento. Os resultados obtidos podem ser utilizados para refinar o mapeamento e manuseio de técnicas de adaptação dos novos conceitos adicionados que podem resultar em novas correspondências. Neste trabalho foi mostrada a viabilidade da adaptação do mapeamento baseada na MAA e heurísticas

modeladas bem como o quão longe a abordagem pode ir para habilitar a adaptação automática (REIS *et al.*, 2015).

Javed, Abgaz e Pahl (2009) (4) propõem uma abordagem para lidar com evolução de ontologias, em particular com a representação da mudança por meio de um framework de operadores em camadas baseado em padrões. Como resultado de um estudo empírico foram identificados quatro diferentes níveis de operadores de mudança baseados na granularidade, domínio específico e abstração de mudanças. As primeiras duas camadas são baseadas em operadores de mudança estrutural genéricos. As outras duas são padrões específicos de mudança de domínio. Essas camadas de padrões de mudança capturam a real mudança em domínios selecionados. Nesse trabalho é discutida a identificação e integração de diferentes camadas com correte e restrições de consistência.

Uma característica central da estrutura de evolução proposta é a operacionalização de mudanças. Operações de mudanças podem ser atômicas, compostas ou complexas. Isso indica que a efetividade de uma mudança é significativamente dependente da granularidade, de como os operadores de mudança são combinados e da extensão de seus efeitos na ontologia. Os efeitos dos operadores de mudança podem afetar a consistência da ontologia. Assim, o tratamento coerente dos operadores de mudança e seus efeitos sobre a consistência em cada nível de granularidade torna-se imprescindível (JAVED, ABGAZ E PAHL, 2009).

Outra característica central da proposta é a representação das mudanças que são frequentes em um domínio, em um maior nível de granularidade. Por exemplo, padrões específicos de domínio que são geralmente negligenciadas em operadores de mudança composicional de menor nível abordados na literatura. Assim, a categorização de operadores em um domínio específico permite suportar um alto nível de abstração. Esta abstração permite mapear os níveis específicos de domínio para níveis abstratos e facilitar a ligação de ontologias com ontologias superiores (JAVED, ABGAZ e PAHL, 2009).

A avaliação empírica realizada foi dividida na seleção do domínio e na análise propriamente dita. Administração Universitária e Sistemas de Banco de Dados foram os domínios selecionados. Na análise empírica o problema foi abordado como um estudo empírico e conceitualização de dois domínios. A abstração das mudanças em um alto nível na hierarquia é feita pela conceitualização de resultados do estudo. Baseando-se na observação

dos operadores de mudança identificados em todas as versões da ontologia, foram estudados os padrões comuns e a partir desse estudo foi criado um *framework* de operadores de mudanças. Foram identificados quatro níveis de operadores de mudança e padrões baseados na granularidade das mudanças. No primeiro nível foram identificadas as mudanças elementares que são tarefas atômicas, no segundo nível foram agregadas as mudanças para representar tarefas compostas e complexas, no nível três, foram agregados padrões de mudança de domínios específicos. O nível quatro foi construído baseado-se na abstração dos padrões de mudanças do domínio específico. Os operadores do primeiro e segundo nível são predefinidos, já os do terceiro e quarto precisam ser customizados (JAVED, ABGAZ e PAHL, 2009).

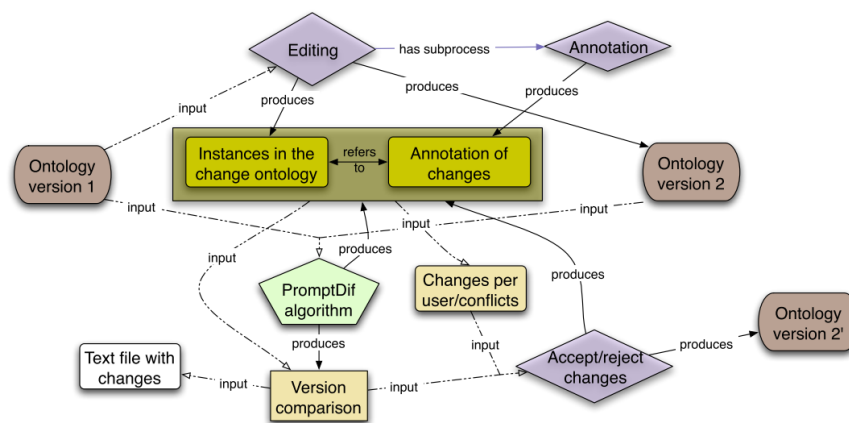
O estudo permitiu observar que mudanças em ontologias são dirigidas por tipos comuns, geralmente mudanças frequentes no domínio de aplicação. O número de padrões de mudanças básicas podem ser fornecidos de modo que o usuário possa adaptar e gerar seus próprios padrões para conhecer sua própria demanda de domínio, o que torna a evolução rápida e fácil. O *framework* possui como benefício a possibilidade de lidar com mudanças estruturais e semânticas em dois níveis sem perder sua independência, permitir a definição de um conjunto de mudanças de um domínio específico e compartilhar mudanças de um domínio específico entre outras ontologias que tenham conceitualizações e especificações similares. O estudo empírico indicou que a solução é validada e adequada para manusear efetivamente a evolução de ontologias (JAVED, ABGAZ e PAHL, 2009).

Conforme Noy *et al.* (2006) (5), o principal problema do desenvolvimento colaborativo de ontologias é a diversificação de tarefas de manutenção do seu conteúdo, pois essa diversidade requer a utilização de sistemas para suporte diversificados. Em seu trabalho Noy *et al.* (2006), cita a ferramenta *Protégé* como alternativa viável para a manutenção de ontologias, pois dá suporte à adição de *plug-ins* específicos para cada atividade que se deseja desempenhar durante esse processo. Entretanto, a utilização de múltiplos *plug-ins* é vista pelos autores como um problema, pois dificulta o processo de manutenção. Considerando essa problemática, e com o intuito de saná-la, os autores propuseram um *framework* flexível o suficiente para trabalhar em qualquer ambiente colaborativo, onde o *Protégé* e seus *plug-ins*

são utilizados para acompanhar ambientes onde não existem relatórios de fases das mudanças realizadas.

No centro desse *framework* está a *Change and Annotation ontology* (CHAO), uma ontologia que reúne duas classes principais (*Change* e *Annotation*) e cujas instâncias relatam mudanças específicas e metainformações sobre elas como: autor, data e hora, anotações, status de aceitação, entre outras. O gerenciador de mudanças do *Protégé* é usado para edição de ontologias e os *plug-ins* criam instâncias CHAO como um produto secundário resultante do processo de edição/manutenção da ontologia (NOY *et al.*, 2006). Os componentes resultantes da modelagem do *framework* proposto por Noy *et al.*, (2006), bem como suas relações, podem ser observados na Figura 8

Figura 4. Componentes do *framework* para evolução de ontologias em ambientes colaborativos



Fonte: Noy *et al.* (2006)

A definição do *framework* foi baseada em dois *plug-ins* da ferramenta *Protégé*: o *Change-management plug-in* que fornece acesso ao histórico de mudanças e habilita alterações, e o *Prompt plug-in* que faz a comparação entre versões e auxilia na tarefa de aceite ou rejeição de mudanças. Por meio desse trabalho foi possível que se identificasse as atividades mais relevantes a serem consideradas no processo de evolução de ontologias em ambientes onde o mesmo necessita ser realizado por vários colaboradores.

Por meio da descrição de diferentes representações de mudanças, Klein e Noy (2003) (6) propõem um *framework* para integrá-las e mostram como diferentes representações estão relacionadas pela descrição de algumas técnicas e heurísticas que suplementam informações



em uma representação que possui informações acerca de representações anteriores. O núcleo do *framework* proposto nesse trabalho é apresentado como uma ontologia de operações de mudanças.

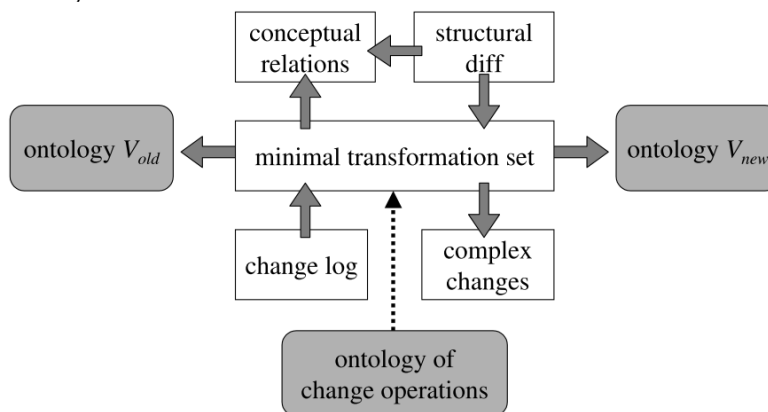
Em muitos casos, ontologias são desenvolvidas em ambientes não controlados, descentralizados e distribuídos. Desse modo, as aplicações provenientes desse tipo de ontologias requerem suporte substancial para o gerenciamento das mudanças que venham a ocorrer na ontologia bem como para o seu processo de evolução. Dada uma ontologia *O* e suas duas versões *V1* e *V2*, um suporte completo para o gerenciamento de mudanças em seu ambiente inclui apoio à transformação dos dados, ao acesso aos dados, à atualização da ontologia, ao seu raciocínio consistente e à sua verificação e aprovação. Na transformação dos dados, quando uma ontologia é atualizada para uma nova versão, os dados descritos na versão anterior precisam ser traduzidos para alinhar-se à nova versão. Por exemplo, em caso de união de dois conceitos *A* e *B*, suas respectivas instâncias devem ser combinadas. No acesso aos dados, ainda que a transformação não tenha ocorrido, deve ser possível acessar os dados da versão anterior por meio da nova versão, ou seja, o usuário deve poder recuperar dados que eram acessíveis por meio de consultas direcionadas à versão antiga, por meio de consultas agora direcionadas à nova versão. Além disso, instâncias de conceitos da versão anterior devem ser instâncias de conceitos equivalentes na nova versão. Na atualização da ontologia, as mudanças locais devem ser propagadas para os artefatos dependentes. No raciocínio consistente, quando uma ontologia passa por mudanças, deve-se analisar seu conteúdo com vistas a verificar se os axiomas da versão anterior continuam válidos na versão atual. Um exemplo disso é a verificação de uma mudança para saber se a mesma não afetou a relação entre dois conceitos. A tarefa de verificar e aprovar mudanças é necessária quando o processo evolutivo de uma ontologia centralizada é realizado colaborativamente, ou seja, por várias pessoas ao mesmo tempo ou quando os desenvolvedores desejam aplicar mudanças de maneira seletiva. De modo a simplificar a verificação da ontologia, deve haver uma interface de usuário por meio da qual seja possível que desenvolvedores aceitem ou rejeitem mudanças específicas, permitindo a execução de algumas mudanças e retrocedendo outras.

Klein e Noy (2003) apresentam passos para representar informações relativas a mudanças em uma nova versão de uma ontologia. O primeiro passo é o uso da versão anterior

da ontologia para encontrar informações de mudanças sem explicitá-las; o segundo está relacionado ao uso de um registro com as mudanças que foram realizadas na versão anterior que resultaram na versão atual da ontologia. O terceiro passo é o uso de uma ferramenta *diff* para o fornecimento de uma visão declarativa das mudanças entre as duas versões. Um conjunto de mudanças conceituais entre versões fornecendo uma especificação explícita de relações entre conceitos da versão anterior e da versão atual, devem ser utilizados para alcançar o quarto passo. O último passo diz respeito ao uso de um conjunto de transformações que descrevem um conjunto de operações de mudança suficiente para a transição da versão anterior para a atual fornecendo assim uma visão operacional da mudança (KLEIN e NOY, 2003).

O *framework* (Figura 9) proposto possui como objetivo relatar informações de mudanças disponíveis em diferentes formalismos e fornecer mecanismos para derivar novas informações a partir de informações preexistentes. O núcleo do *framework* é um conjunto mínimo de transformação que fornece um conjunto de operações necessárias e suficientes para transformar uma ontologia.

Figura 5. Representação esquemática do *framework* para descrição de mudanças



Fonte: Klein e Noy (2003)

Esse trabalho apresenta como contribuição o estudo de formalismos para representar mudanças em ontologias, um *framework* baseado em um componente para definição de mudanças de ontologias, uma ontologia de operações de mudanças básicas, que são mudanças que afetam apenas uma característica do modelo de conhecimento OWL e a definição de mudanças complexas de ontologias que fornecem base para uma transformação de dados mais

eficiente e recursos avançados de interação com o usuário. Mudanças complexas são definidas como operações compostas por múltiplas operações básicas ou que incorporam algum conhecimento adicional sobre a mudança. Regras e heurísticas para a definição de mudanças complexas entre versões de uma ontologia, também são apresentadas como contribuições desse trabalho (KLEIN e NOY, 2003).

Klein (2004) (7) apresenta um *framework* que consiste em uma especificação geral e concisa para mudanças em ontologias e um número de modelos de processos que descrevem como uma especificação de mudança pode ser gerada e usada para diferentes tarefas de gerenciamento de mudanças. O *framework* definido possui como elemento central uma meta-ontologia de operações de mudanças que especifica mudanças padrão de uma ontologia. A noção de mudanças complexas e de um conjunto de transformações de uma ontologia e um template para especificar relações entre diferentes versões da ontologia também são elementos do *framework* proposto por Klein (2004). Os métodos aplicados nesse trabalho possibilitaram o entendimento sobre como novas informações acerca de mudanças podem ser derivadas a partir de outras representações de mudanças e assim suplementar informações de modo a apoiar tarefas de evolução de ontologias não permitidas pelas informações originais da mudança (KLEIN, 2004).

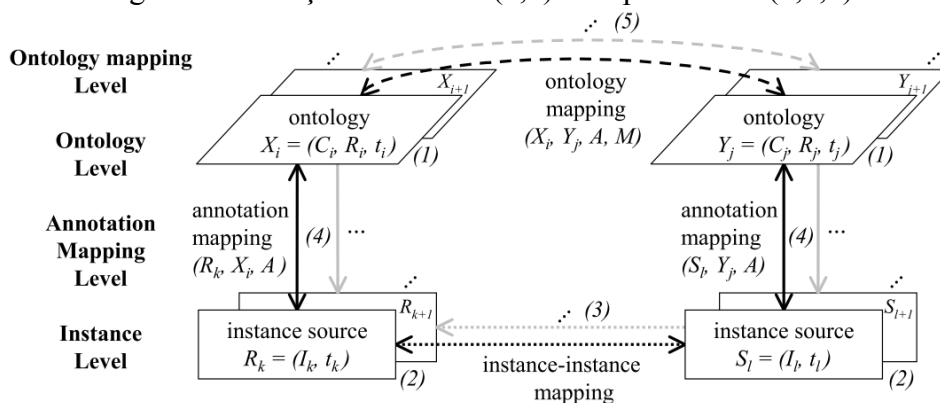
Hartung, Kirsten e Rahm (2008) (8) propõem um framework genérico para analisar a evolução de ontologias e os mapeamentos entre as mesmas especialmente anotações e similaridades. O framework foi usado para uma avaliação comparativa entre métricas de evolução de 16 ontologias da área das ciências da vida. Também foi analisada a evolução do mapeamento de anotações e de ontologias para a GO.

Esse trabalho apresenta como contribuição um *framework* que permite sistematicamente estudar a evolução e fontes de dados das instâncias de ontologias bem como a evolução do mapeamento a ela relacionado. O *framework* proposto suporta a computação de várias medidas gerais para descrever versões de ontologias e mapeamentos bem como sua evolução. Outra contribuição é a aplicação do *framework* proposto por meio de uma avaliação compreensiva de 386 versões de 16 ontologias sobre as ciências da vida incluindo subontologias da GO e da NCI *Thesaurus*. Em particular foram usadas as métricas do *framework* proposto para analisar os principais tipos de mudanças dentre outras características

da evolução. A análise do impacto da evolução de ontologias para diferentes mapeamentos gerados também é uma contribuição desse trabalho cujo objetivo é ajudar desenvolvedores e usuários a melhor entenderem as consequências das mudanças em ontologias e guiar o desenvolvimento de algoritmos na geração de mapeamentos que permaneçam robustos mesmo com as mudanças (HARTUNG, KIRSTEN e RAHM, 2008).

O *framework* proposto por Hartung, Kirsten e Rahm (2008) possui 2 tipos básicos de evolução como pode ser observado na Figura 10. De um lado é realizada a investigação de fontes únicas de ontologias (1) e fontes de instâncias (2), para ambos os tipos de fonte a evolução é refletida em uma série de versões. Por outro lado é considerada a evolução dos mapeamentos. Esses mapeamentos existem entre versões de diferentes fontes de instância (mapeamento instância-instância (3)), entre versões de fontes de instâncias e ontologias (mapeamento de anotações (4)) e entre versões de diferentes ontologias (mapeamento de ontologias (5)).

Figura 6. Evolução das fontes (1,2) e mapeamentos (3,4,5).



Fonte: Hartung, Kirsten e Rahm (2008)

Conforme o modelo de ontologia definido para o *framework* proposto, uma ontologia  $ON_v = (C, R, t)$ , é definida por um número de versões  $v$ , conceitos  $C = \{c_1, c_2, \dots, c_m\}$ , relacionamentos  $R = \{r_1, r_2, \dots, r_n\}$  e a hora em que foi criada  $t$ . Também foi definido um modelo de instância conforme o qual uma fonte de instância  $IS_v = (I, t)$  de determinada versão  $v$  consiste de um conjunto de instâncias  $I = \{i_1, i_2, \dots, i_n\}$ . O modelo de mapeamento de anotações  $AM = (IS_u, ON_v, A)$  descreve um mapeamento entre uma fonte de instância de uma versão  $u$  e uma ontologia  $ON$  de versão  $v$ . O mapeamento propriamente dito denotado por  $A$  é um conjunto de associações binárias entre instâncias  $I$  de  $IS_v$  e conceitos  $C$  de  $ON_v$ . No

modelo de mapeamento de ontologias  $OM = (X_u, Y_v, A, M)$ , duas ontologias  $X_u, Y_v$  possuem um conjunto de correspondências  $A$  baseadas em um algoritmo de combinação  $M$ . Uma correspondência única  $n_k = (x_k, y_k, sim_k) \in A$  compreende dois conceitos de ontologias (conceito  $x_k$  de  $X_u$  e conceito  $y_k$  de  $Y_v$ ) e um valor de similaridade  $sim_k$ . O valor de similaridade indica a força da similaridade entre dois conceitos de uma ontologia e geralmente é um valor numérico do intervalo  $[0,1]$  (HARTUNG, KIRSTEN e RAHM, 2008).

Também foi definido um modelo comum de evolução genérico e aplicável a todos os modelos definidos, em particular os para ontologias, instâncias, anotações e mapeamentos de ontologias. As bases do modelo de evolução são conjuntos de objetos de  $O_{vi}$  de uma versão  $vi$  de uma fonte que evolui. Tais objetos compreendem conceitos, relacionamentos, associações de anotações e correspondências entre ontologias. A partir das análises realizadas nas 16 ontologias foi possível observar que a maioria delas são fortemente alteradas e crescem significativamente e que a maioria dos seus conceitos são interligados por relacionamentos do tipo *Is-a* ( $> 85\%$ ) (HARTUNG, KIRSTEN e RAHM, 2008).

Stojanovic et al. (2002) (9) apresentam uma abordagem orientada a processos para ajudar na evolução de ontologias com base em seus requerimentos. Tal abordagem é focada na manutenção da consistência após a aplicação de mudanças complexas durante a evolução de uma ontologia. A abordagem proposta analisa as causas e consequências das mudanças e garante a consistência da ontologia e dos artefatos que dela dependem. É introduzida a ideia de estratégia de evolução permitindo o usuário customizar o processo de acordo com as suas necessidades. O usuário pode sugerir mudanças adicionais que podem produzir uma ontologia mais adequada às suas necessidades. A abordagem é aplicada utilizando o *Kaulshure ontology on Semantic Web framework* (KAON), um *framework* que utiliza recursos disponíveis e fornece ferramentas para engenharia, descoberta, gerenciamento e apresentação de ontologias e metadados. Além da interação do usuário com o processo de evolução a abordagem possui como benefício o melhoramento contínuo por meio da descoberta semi-automática. A validação da abordagem e a estratégia de evolução desenvolvida ajuda o usuário no melhor entendimento dos efeitos de cada mudança fornecendo um discernimento detalhado sobre elas.

No trabalho realizado por Flouris *et al.* (2006) (10) é proposto um *framework* geral para inconsistências, negações e mudanças em ontologias, onde é fornecida uma base única de inconsistências e processamento de mudanças para ontologias baseadas em DL. Foram identificadas distinções entre diferentes tipos de inconsistências e negações em ontologias. Os autores discutiram ainda como o *framework* pode fornecer a criação de tarefas de gerenciamento e raciocínio de ontologias.

Oliver *et al.* (1999) (11) propõem um modelo conceitual, um conjunto de operações de mudanças e um modelo de documentação de mudanças que pode ser apropriado para terminologias controladas no cuidado em saúde. O modelo é adequado para a resolução de problemas como o gerenciamento de terminologias médicas e sua evolução por meio da revisão de sistemas existentes para entender como eles representam conceitos e mudanças manuais de modo a acessar suas similaridades e diferenças e inferir acerca de requerimentos e metas comuns.

Zablith (2008) (12) propõe um *framework* chamado Evolve cujo foco é reduzir ou eliminar entradas de usuários durante o processo de evolução de ontologias através da exploração de fontes de conhecimento. O *framework* possui como componente a descoberta da informação, onde são detectadas informações externas a serem adicionadas à ontologia; a validação de dados, onde as informações coletadas são validadas por meio da aplicação de regras heurísticas e; mudanças ontológicas, processo de descoberta de relações onde são determinadas as ligações entre os conceitos do conhecimento existente. Para o componente “mudanças ontológicas” foi proposta uma técnica de combinação gradual que se inicia com métodos simples e segue com métodos mais complexos.

O primeiro método é uma *string* de combinação para identificar equivalências. No segundo método, caso não existam equivalências o termo a ser adicionado é considerado como novo. No terceiro método são buscadas relações subjacentes usando hierarquia de sentidos e caso não existam usa-se a *Web Semântica* para descoberta de relações mais ricas. No quarto método é explorada toda a *web* através de *Applications Programming Interfaces* (APIs) de mecanismos de busca, juntamente com o uso de padrões lexicais. Ainda assim, caso não se encontre relações o termo é descartado ou pode ser possível recorrer-se à checagem manual. Outro componente do *framework* é a validação da evolução, onde é realizada a

validação do processo evolutivo da ontologia por meio da checagem de consistência e duplicação bem como do processo de raciocínio temporal. O gerenciamento da evolução é o último componente do *framework* e é onde as mudanças realizadas são registradas para garantir funcionalidades como rastrear e retroceder mudanças.

Haase e Stojanovic (2005) (13) propõem um modelo com foco na semântica da mudança para ontologias OWL, considerando consistências estruturais, lógicas e definidas pelo usuário. Nesse trabalho são definidos métodos para detectar e resolver inconsistências em uma ontologia OWL depois da aplicação de mudanças e introduzidas estratégias de resolução para garantir que a consistência seja mantida durante o processo evolutivo da ontologia. Pelo fato de algumas mudanças terem estados diferentes de consistência, definiu-se estratégias de resolução permitindo o usuário controlar a evolução. Para exemplificar, foram apresentadas várias condições de consistência, por exemplo, a consistência de uma ontologia, onde foi definido que uma ontologia  $O$  é consistente com respeito a um conjunto de consistências  $K$ , se para todo  $k \in K$   $O$  satisfaz a condição de consistência  $k$  ( $O$ ).

A principal contribuição do trabalho realizado por Haase e Stojanovic (2005) foi a criação de uma abordagem para formalizar a semântica da mudança da OWL incorporada a um processo genérico de evolução de ontologias, tendo em vista que a formalização permitiu definir-se condições de consistência arbitrárias classificadas como estruturais, lógicas e definidas pelo usuário. A formalização permitiu ainda definir estratégias de resolução que atribuem funções de usuário que garantem que essas condições de consistência sejam satisfeitas durante a evolução da ontologia.

Tratar os impactos das mudanças antes da sua implementação permanente é a proposta do trabalho realizado por Abgaz, Javed e Pahl (2012) (14). Para isso foram identificados possíveis impactos semânticos e estruturais e proposto um método de análise de mudanças button-up (de baixo para cima) que contém duas fases. A primeira foca na análise do impacto de operações de mudanças atômicas, também chamadas de mudanças elementares e definidas por Stojanovic *et al.* (2002) como mudanças que modificam apenas uma entidade do modelo ontológico. A segunda fase se concentra nas mudanças compostas, também chamadas de mudanças complexas por implementarem duas ou mais mudanças atômicas. O método desenvolvido nesse trabalho fornece informações cruciais de impacto e pode ser usado para

relacionar estratégias de evolução e conduzir análises “e-se” antes da evolução. É fornecida ainda uma análise detalhada de como e por que as mudanças afetam as entidades. Ajudar o usuário a ver os impactos das operações de mudança bem como suas causas, permitir comparações entre estratégias de implementação em termos de impactos e permitir aos usuários a realização de análises “e-se” antes de alterar a ontologia permanentemente são os benefícios desse trabalho.

Lambrix *et al.* (2016) (15) propõem um conjunto de funcionalidades para sistemas de evolução de ontologias com foco na visualização de ontologias e suas respectivas versões bem como informações requeridas pelas tarefas do processo evolutivo. Nesse trabalho também é feita uma revisão do estado da arte de sistemas de evolução de ontologias com respeito à visualização e requerimentos assim como é discutido como o usuário pode e deve ser introduzido no uso de ferramentas para suporte ao processo de evolução. Um dos desafios encontrados pelos autores nas técnicas de visualização é a representação da riqueza de ontologias em um caminho escalável e compreensivo tendo em vista que ferramentas que realizam essa tarefa precisam representar as diferenças entre duas versões de um modo compreensivo em diferentes níveis de granularidade.

Uma abordagem para detecção de mudanças no contexto de um *framework* para OWL DL é proposta por Plessers, Troyer e Casteleyn (2007) (16) . A motivação desse trabalho reside em ajudar usuários a entender a evolução de uma ontologia e fornecer uma visão geral das mudanças nela aplicadas. O *framework* desenvolvido permite engenheiros de ontologias fazerem requisições e aplicações de mudanças de um modo que garanta a consistência da ontologia e dos artefatos que dela dependem. O mesmo inclui um mecanismo para detecção de mudanças que permite gerar automaticamente uma visão detalhada de mudanças que têm ocorrido baseado-se em um conjunto de definições. O *framework* permite ainda que os usuários possam ter seu próprio conjunto de definições que resultam em diferentes visões de mudanças, com diferentes pontos de vista sobre como a ontologia foi alterada. O *framework* possui dois tipos de usuários: o Engenheiro de ontologias, que é parcialmente responsável pela ontologia, e todos que precisam ser informados sobre as mudanças da ontologia da qual dependem.



Foi apresentada uma Linguagem de Definição de Mudanças (CDL do inglês, *Changes Definition Language*) e introduzido um *log* de versões como alternativa para lidar com o entendimento da evolução. A CDL permite usuários definirem formalmente o significado das mudanças. O *log* de versões mantém um registro com todas as versões de todos os conceitos definidos em uma ontologia. Junto com o *log* de versões o CDL fornece a definição de uma abordagem para a detecção de mudanças cujo propósito é detectar se as mudanças representadas no CDL realmente ocorreram por meio da realização de consultas temporais no *log* de versões. A abordagem proposta nesse trabalho permite que diferentes usuários definam conjuntos divergentes de mudanças representando seus interesses e especificando tais mudanças em um nível de abstração mais apropriado. Isso permite que cada usuário obtenha seu próprio conjunto de definição de mudanças bem como seu próprio *log* de evolução (PLESSERS, TROYER e CASTELEYN, 2007).

O objetivo do trabalho realizado por Plessers e Troyer (2006) (17) é verificar se uma ontologia permanece em estado consistente mesmo depois da aplicação de mudanças. Nesse trabalho é proposto um algoritmo para selecionar axiomas responsáveis pela inconsistência de uma ontologia construída em OWL DL, bem como um conjunto de regras que podem ser usadas por Engenheiros de ontologias para resolver as inconsistências detectadas. Foram realizadas duas tarefas baseadas na distinção entre TBox, que compreende o conhecimento terminológico da ontologia, e ABox, o conhecimento assercional da ontologia. Na primeira tarefa foi adicionado ou modificado um axioma para TBox, na segunda um axioma foi adicionado ou modificado para ABox. A principal contribuição desse trabalho é a definição de uma abordagem que determina os axiomas que causam inconsistências lógicas, bem como um conjunto de regras que podem ser aplicadas para resolver as inconsistências que forem detectadas.

Baseando-se em uma análise empírica, Abgaz, Javed e Pahl (2011) (18) propõem um framework conceitual para análise de impacto baseada em dependências. Também foram identificados possíveis impactos e suas causas, a dependência entre as entidades, sua severidade e fatores que afetam o processo de análise de impacto em Sistemas Baseados em Conteúdo Dirigido a ontologias (ODCBSs do inglês, ontology-Driven Content-Based Systems). Essa proposta apresenta uma solução para a natureza dinâmica do conteúdo das

ontologias que apoiam os ODCDSs bem como sua interdependência de conteúdo que agravam o desafio da evolução dinâmica. Outro problema para o qual esse trabalho apresenta uma solução é a dificuldade de determinar mudanças em ODCBSs relativamente grandes que em geral consome muito tempo e não garante uma solução completa. O *framework* proposto é usado para fornecer diretrizes formais e terminologias pra apoiar mudanças operacionais e analíticas. Durante a análise da proposta foram identificadas dependências e impactos em um *framework* conceitual categorizado baseando-se em diferentes critérios que servem como entrada para o proceso de análise de impactos. Na pesquisa realizada por Abgaz, Javed e Pahl (2011) a análise empírica das dependências em ODCBSs, dos tipos de impacto e dos fatores que afetam sua análise permitiu garantir uma visão prévia, captura e apresentação automática, redução e determinação acurada dos impactos das mudanças sobre sistemas dependentes.

Ochs *et al.*, (2015) (19) introduz uma rede de abstração diff. Redes de abstração diff são redes compactas que sumarizam e visualizam mudanças estruturais globais devido a operações de edição de ontologias que resultam em uma nova versão. A rede criada pode ser usada para apoiar curadores na identificação de mudanças não entendíveis e não requeridas.

Como pode ser observado na Tabela 3, predominam os trabalhos que focam na semântica da mudança. A consistência da ontologia foi um tema abordado em 9 trabalhos analisados. A representação e a verificação e validação ficam logo atrás tendo sido abordados em 7 trabalhos. O desenvolvimento colaborativo é investigado em 5 dos trabalhos analisados. Os temas menos abordados são o versionamento e o mapeamento entre versões. O maior número de abordagens relacionadas à semântica da mudança denota a importância dada à representação compreensiva e clara das mudanças antes da sua aplicação durante o processo de evolução de ontologias. Outro ponto de considerável importância durante a evolução ontológica é o uso dos recursos necessários para a manutenção da consistência da ontologia em evolução. Assim, diversos trabalhos estudados nesse levantamento, abordam esse tema apontando formas de manter a consistência das ontologias após passarem pelo processo evolutivo.

O desenvolvimento colaborativo é um tema em evidência em trabalhos sobre evolução de ontologias tendo em vista o crescimento de ontologias desenvolvidas com a contribuição de diversas pessoas. ontologias Biomédicas como SNOMED CT, uma ontologia desenvolvida

por meio da contribuição de diversos profissionais da saúde de diversos países do mundo, são exemplos de ontologias desenvolvidas colaborativamente. A verificação e validação da ontologia está diretamente relacionada à manutenção da sua consistência, pois é durante esses processos que são identificados problemas de consistência causados pela evolução.

A principal contribuição desse capítulo para este trabalho é a identificação dos tópicos mais abordados por pesquisadores do campo de pesquisa voltado para a evolução de ontologias. Tal identificação possibilita a observância dos tópicos que exigem maior atenção na formulação de estratégias de evolução. Os tópicos menos abordados aqui identificados (versionamento e mapeamento) denotam um estado de ascensão das pesquisas a eles relacionadas.

O trabalho de número 20 representa o *framework* proposto nesta dissertação, o qual apresenta um método conceitual para suporte à Evolução de ontologias biomédicas. O mesmo possui como contribuição um conjunto de processos inter-relacionados com tarefas formalmente descritas cujo objetivo se concentra em abordar o processo evolutivo de ontologias Biomédicas diante de três motivos de evolução mais comuns que são: a mudança de foco da ontologia, mudanças nas diretrizes do domínio da ontologia e mudanças na terminologia de referência utilizada para a representação terminológica da mesma.

Tendo como referência os três motivos de evolução citados, o *framework* fornece atividades a serem realizadas no processo evolutivo dentre as quais estão a representação semântica das mudanças, tarefas de implementação das mudanças representadas, verificação e validação das mudanças por meio de atividades de análise e auditoria, e a resolução de inconsistências decorrentes das mudanças com o intuito de evitar que a evolução cause prejuízos à consistência lógica e/ou estrutural da ontologia em evolução.

Comparando-se o trabalho aqui proposto com os trabalhos relacionados analisados, este mostra-se como o mais completo em relação aos elementos utilizados como critério de análise. Os trabalhos analisados realizam estudos isolados acerca dos elementos que compõem o processo evolutivo. Em contrapartida, o *framework* construído e descrito nesta dissertação aborda detalhadamente cada uma das etapas cruciais do processo evolutivo de ontologias utilizando-se de ferramentas e exemplos que possibilitem que o mesmo seja aplicado adequadamente à evolução de ontologias do domínio biomédico. Além do

detalhamento de cada uma das fases da evolução, o *framework* pode ainda ser adaptado aos três possíveis motivos de evolução citados anteriormente que são a mudança do foco da ontologia, mudanças em suas diretrizes de domínio e mudanças relativas à sua terminologia de referência, que pode motivar evoluções para padronização terminológica adaptativa ou para padronização terminológica total.

Em síntese, o *framework* construído une todas as etapas do processo de evolução de ontologias, que são abordadas de modo fragmentado nos trabalhos relacionados, detalhando-as e adaptando-as para apoiarem a evolução das ontologias biomédicas motivada por mudança de foco, mudança nas suas diretrizes de domínio e por mudanças relacionadas à sua terminologia de referência.

Tabela 3. Resultado das buscas por temas do processo de evolução de ontologias em 19 trabalhos relacionados.

Identificação do artigo	Temas							
	Representação da Mudança	Semântica da Mudança	Consistência da ontologia	Desenvolvimento Colaborativo	Verificação e Validação da Mudança	Versionamento	Mapeamento	Suporte para a Evolução de ontologias Biomédicas
1	✓	✓		✓	✓			
2			✓			✓		
3				✓	✓		✓	✓
4	✓	✓						
5				✓	✓			
6	✓	✓		✓	✓			
7	✓	✓		✓				
8						✓	✓	✓
9	✓	✓	✓		✓			
10			✓					
11	✓							✓
12	✓	✓	✓		✓			
13		✓	✓					
14		✓	✓					
15		✓			✓			
16		✓	✓		✓			
17		✓	✓					
18		✓	✓					
19		✓			✓			
20	✓	✓	✓		✓			✓

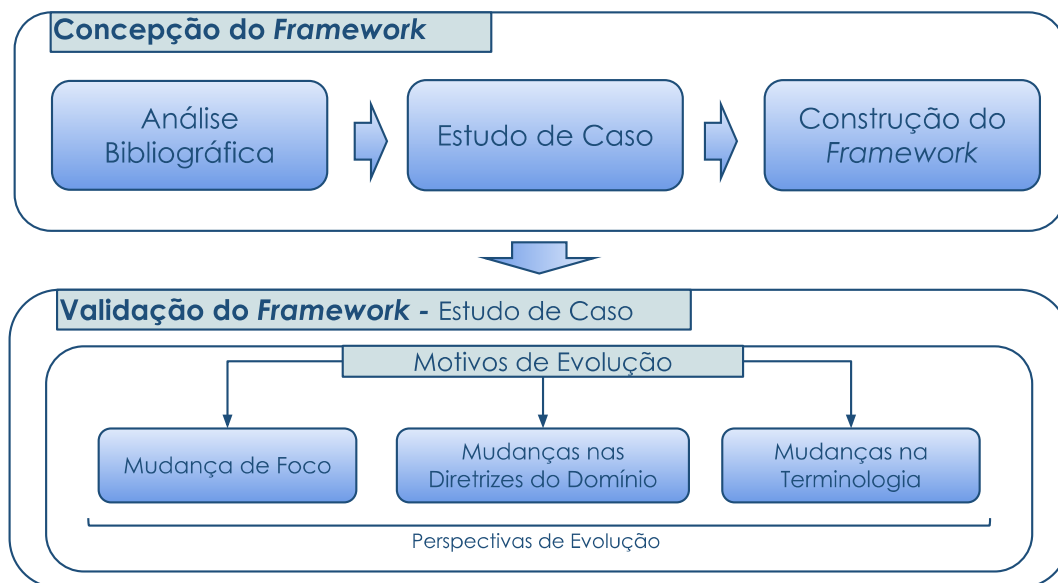
Fonte: Autoria própria

# Capítulo 4

## 4 Materiais e Métodos

O trabalho descrito nesta dissertação foi desenvolvido em duas etapas principais (Figura 7) nas quais foram aplicados diferentes materiais e métodos para alcançar o objetivo de cada uma. A primeira etapa é a concepção do *framework*, onde estão explanados os recursos utilizados para a criação do *framework* propriamente dito. Dentre eles está um estudo de caso, onde uma ontologia sobre a Doença Renal Crônica (DRC) foi submetida a processo de evolução com o intuito de obter-se subsídios para a criação do *framework*. Esse estudo de caso teve como principal finalidade identificar elementos essenciais a serem considerados durante a evolução de ontologias biomédicas. O estudo de caso está descrito em Sousa et al. (2017).

Figura 7. Estrutura da metodologia utilizada na construção do *framework* para suporte à Evolução de ontologias Biomédicas.



Fonte: Autoria própria

A validação do *framework* é a segunda etapa da metodologia. Nela foi realizado um novo estudo de caso por meio do qual a aplicabilidade do *framework* construído foi validada. O estudo de caso foi realizado com base nas fases que compõem a estrutura do *framework* concebido por meio deste trabalho. Nesse estudo de caso foram consideradas três perspectivas baseadas nos três motivos de evolução identificados no estudo de caso realizado para levantar

requisitos para a construção do *framework*. Tais motivos são a mudança de foco da ontologia, as mudanças em suas diretrizes de domínio e mudanças em sua terminologia de referência. As duas etapas da metodologia são detalhadamente descritas nas subseções seguintes.

## **4.1 Conceção do *framework***

### **4.1.1 Análise Bibliográfica**

O *framework* definido neste trabalho foi criado tendo como base a literatura pertinente sobre evolução de ontologias. Inicialmente foram realizadas buscas por trabalhos relacionados a *frameworks*, abordagens ou metodologias que tenham como finalidade apoiar a evolução de ontologias. Para realizar as buscas foram utilizadas plataformas livres de disponibilização de artigos científicos onde foram usadas as palavras-chave “*Framework*” ou “Metodologia” ou “Abordagem” em conjunto com “Evolução de ontologias”.

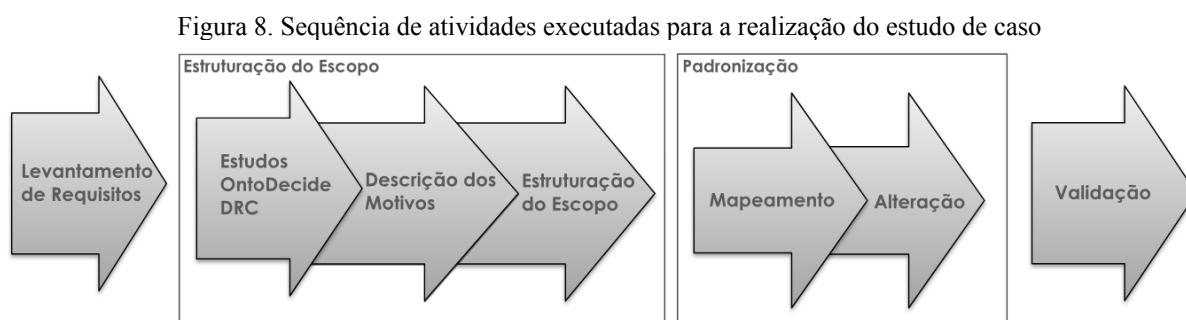
Após as buscas, os trabalhos encontrados foram analisados tendo como finalidade observar como cada estrutura de apoio foi projetada considerando seu escopo bem como os benefícios oferecidos pelo método utilizado de modo a verificar se o mesmo poderia oferecer alguma utilidade para a construção deste trabalho. Após a análise dos trabalhos encontrados foram realizados estudos aprofundados sobre o processo convencional de Evolução de ontologias, tendo tais trabalhos como direcionadores das pesquisas e estudos realizados. Esse processo de estudos foi importante pois possibilitou o entendimento das etapas consideradas essenciais no processo evolutivo de ontologias, como representação, implementação e validação das mudanças, o que permitiu a modelagem e construção de uma solução voltada pra o melhoramento de soluções existentes. O entendimento do processo tradicional e geral de Evolução de ontologias possibilitou também o reúso de soluções anteriormente propostas com vistas a remodelá-las oferecendo melhorias e maior detalhamento bem como a especificação para o suporte à evolução de ontologias pertencentes ao domínio biomédico.

### **4.1.2 Estudo de caso para apoio à construção do *framework***

O estudo de caso aqui descrito foi realizado com o intuito de fornecer base para a construção do *framework* proposto nesta dissertação. No mesmo foi realizada a evolução da ontoDecideDRC, ontologia criada para dar suporte ao diagnóstico, estadiamento e tratamento de pacientes com DRC(TAVARES, 2016). O estudo teve como objetivo principal identificar atividades que se mostrassem necessárias para a evolução eficiente de ontologias voltadas

para o domínio da saúde. A decisão de expandir o alcance do *framework* para o domínio biomédico foi tomada posteriormente à realização desse estudo de caso e pode ser justificada por evitar a restrição de domínios representados em ontologias em evolução que envolvam a área médica, tendo em vista que o domínio biomédico está relacionado também com a Biologia.

A Figura 8 ilustra a sequência de tarefas executadas para a realização do estudo de caso onde a ontologia ontoDecideDRC passou por processo de evolução para atender à necessidade dos seus usuários de que a mesma tivesse seu foco especializado para a representação de conceitos relativos aos cuidados realizados durante a atenção primária do paciente. A atenção primária do paciente ou atenção básica é constituída pelos cuidados realizados antes do paciente ser encaminhado para a atenção especializada. Segundo o Sistema Único de Saúde (SUS)<sup>15</sup>, a atenção básica é considerada uma porta de entrada para a gestão ordenada dos cuidados dos usuários do sistema (MINISTÉRIO DA SAÚDE, 2016).



Fonte: Autoria própria

Ao mudar-se o foco da ontoDecideDRC, esta, que anteriormente representava conceitos relacionados ao diagnóstico, estadiamento e tratamento do paciente com DRC e portanto incluía os conceitos relativos à atenção primária e à atenção especializada, passou a representar apenas os conceitos relacionados aos cuidados realizados antes do encaminhamento do paciente para a atenção especializada. O objetivo principal da mudança de foco foi reduzir falhas no diagnóstico da DRC, bem como a ausência de informações importantes e ou a presença de informações incorretas no formulário de encaminhamento do paciente para o Nefrologista. A nova versão da ontologia teve seu nome alterado para ONTODRC.

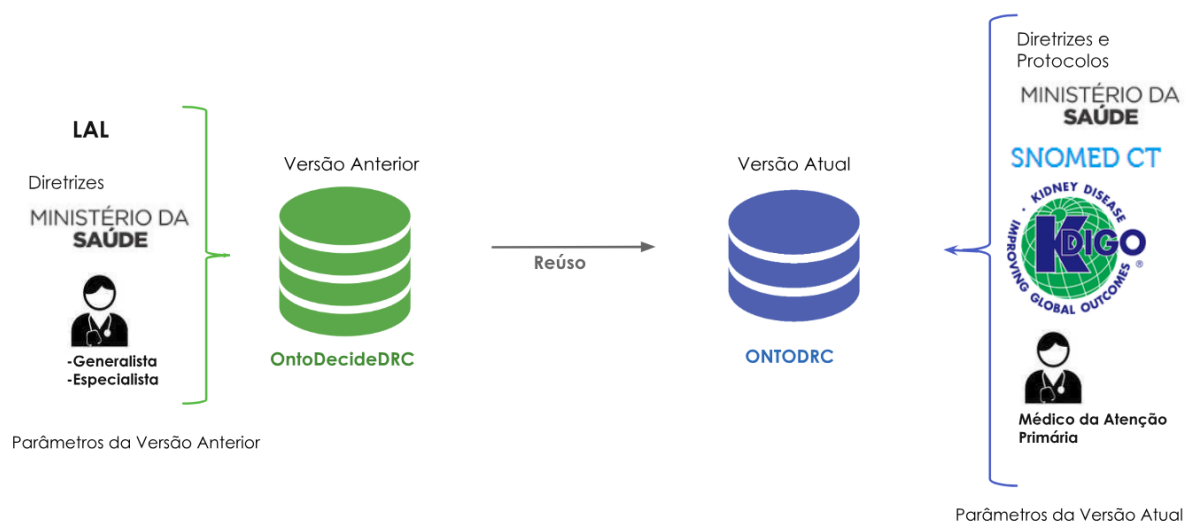
<sup>15</sup><http://portalsaude.saude.gov.br/index.php/cidadao/entenda-o-sus>



A Figura 9 mostra os parâmetros da ontologia antes e depois do processo de evolução. Antes de ser submetida à evolução a ontologia OntoDecideDRC utilizava um Lexico Amplicado da Linguagem (LAL) que, segundo Breitman e Leite (2004), é uma linguagem de representação da Engenharia de Requisitos cujo objetivo é realizar o mapeamento do vocabulário de um Universo de Informações (UdI). Em Engenharia de Requisitos, o UdI compreende o cenário no qual o *software* cujo vocabulário está sendo representado por meio do LAL, deverá ser desenvolvido e posto em operação. Com vistas a padronizar a fonte terminológica da ontologia e por ser o maior padrão de termos clínicos do mundo, a SNOMED CT foi escolhido como terminologia de referência da ontologia em evolução.

Por representar conceitos relacionados à atenção primária e especializada do paciente com DRC, a principal fonte de conhecimento da ontoDecideDRC era o documento de Diretrizes Clínicas para o cuidado do paciente com Doença Renal Crônica (DRC)<sup>16</sup> no SUS. Porém, como o novo foco da ontologia passou a ser o cuidado do paciente até o seu encaminhamento para a atenção especializada, além do uso do documento de Diretrizes Clínicas para o cuidado do paciente com DRC no SUS passou-se a utilizar o documento de Protocolos de Encaminhamento da Atenção Básica para a Atenção Especializada - Endocrinologia e Nefrologia do Ministério da Saúde<sup>17</sup>.

Figura 9. Parâmetros da ontologia antes e depois do processo evolutivo



Fonte: Autoria própria

16 [http://bvsmms.saude.gov.br/bvs/publicacoes/diretrizes\\_clinicas\\_cuidado\\_paciente\\_renal.pdf](http://bvsmms.saude.gov.br/bvs/publicacoes/diretrizes_clinicas_cuidado_paciente_renal.pdf)

17 [http://dab.saude.gov.br/portaldab/biblioteca.php?](http://dab.saude.gov.br/portaldab/biblioteca.php?conteudo=publicacoes/protocolos_atencao_basica_atencao_especializada)

[conteudo=publicacoes/protocolos\\_atencao\\_basica\\_atencao\\_especializada](http://dab.saude.gov.br/portaldab/biblioteca.php?conteudo=publicacoes/protocolos_atencao_basica_atencao_especializada)

Para adaptar a classe da ontologia que faz a classificação da DRC à diretriz internacional, criada e mantida pela KDIGO, foi incluída como fonte de conhecimento da ontologia em evolução, o documento de Avaliação e Gerenciamento da DRC fornecido pela KDIGO. Quanto aos profissionais que realizam o acompanhamento do paciente no domínio da DRC, antes da evolução a ontologia incluía conceitos voltados para o médico Nefrologista e para o Médico da Atenção Primária (MAP), porém, após a evolução o foco da mesma passou a ser apenas o MAP.

#### **4.1.2.1 Levantamento de Requisitos**

Nesta fase foi realizado o estudo das fontes de conhecimento da ontologia considerando o seu novo foco. Tais fontes são: as Diretrizes Clínicas para o cuidado do paciente com DRC no SUS, os Protocolos de Encaminhamento da Atenção Básica para a Atenção Especializada - Endocrinologia e Nefrologia do Ministério da Saúde e o documento de Avaliação e Gerenciamento da DRC fornecido e mantido pela KDIGO.

#### **4.1.2.2 Definição do Escopo**

A tarefa de definição do escopo da ontologia foi dividida em 3 subtarefas que são: Estudos sobre a ontoDecideDRC, Definição dos Motivos da Evolução, Estruturação do Escopo. Na primeira subtarefa foi realizado um estudo aprofundado do foco da ontoDecideDRC. Logo após, suas classes, relacionamentos e instâncias foram analisadas com a finalidade de entender sua estrutura hierárquica. Na segunda subtarefa, os motivos que justificam a evolução da ontologia foram definidos. O motivo da evolução reside na mudança de foco da ontoDecideDRC, que passou a ser o apoio ao processo de cuidado do paciente antes de ser encaminhado para o Nefrologista. O novo foco da ontologia foi sugerido por um médico especialista em Nefrologia, que se baseou em sua experiência de presenciar a grande ocorrência do diagnóstico impreciso e informações incorretas e inconsistentes, no formulário de encaminhamento do serviço de saúde.

Na subtarefa Estruturação do Escopo, de acordo com o novo foco da ontologia, a estrutura da nova versão foi definida. Nessa fase, os componentes desnecessários e os que deveriam ser adicionados foram definidos com a ajuda do médico especialista. Foram realizadas as operações de mudanças básicas na ontologia, que segundo Klein e Noy (2003), modificam apenas uma característica do modelo de conhecimento OWL, por exemplo, operações de adição ou deleção sobre cada uma dessas características. Adicionalmente também foram realizadas mudanças complexas, que são compostas por múltiplas operações básicas, ou aquelas que contêm algum conhecimento adicional sobre a mudança, como por exemplo, uma operação de mudança sobre relações de superclasse. Um exemplo de operação simples realizada neste estudo de caso foi a adição da classe “MotivosDeEncaminhamentoMaisComuns” que, conforme o nome sugere, se refere às doenças que comumente causam o encaminhamento do paciente para o Nefrologista. A alteração da classe “Classificacao” para “Estadiamento” é um exemplo de operação complexa tendo em vista que a alteração do nome dessa classe levou à sua alteração em todos os locais onde a mesma está sendo utilizada, por exemplo, nos axiomas das suas subclasses, representados em OWL pelo axioma de classe *SubClassOf*, responsável por determinar a relação entre tais subclasses e a classe alterada.

#### 4.1.2.3 Padronização

Após a estruturação do escopo foi realizada a padronização da nomenclatura da ontologia submetida ao processo de evolução. O padrão escolhido para essa tarefa foi o SNOMED CT, que substituiu o padrão usado até então, que era baseado nas diretrizes para o cuidado ao paciente com DRC no SUS. SNOMED CT é uma terminologia clínica que pode ser usada para representar clinicamente informações relevantes de forma consistente, confiável e compreensiva como uma parte integrante da produção de informações eletrônicas de saúde (SNOMED CT, 2017).

A tarefa de padronização foi dividida em duas etapas: Mapeamento, no qual termos correspondentes aos da ontologia foram pesquisados no *browser* SNOMED CT e Alteração, etapa na qual os termos da terminologia foram incluído na ontologia como *annotations* criadas para este fim. Foi criada a *annotation* FSN do tipo *string* para receber o nome preferível do termo e a *annotation* SCTID do tipo *integer* para receber o identificador

definido pela SNOMED CT para cada termo importado. Esse processo foi realizado para todos os conceitos da ontologia presentes na terminologia SNOMED CT. Devido à inexistência, no repositório de termos SNOMED CT, de conceitos especificamente utilizados no cuidado de pacientes com DRC no Brasil, a padronização não foi realizada para todos os termos da ontologia. É importante esclarecer que os termos inexistentes são advindos de conceitos utilizados nas Diretrizes Clínicas para o cuidado do paciente com DRC no SUS e portanto, são importantes para a manutenção da consistência da ontologia em relação às suas fontes de conhecimento. Exemplos de termos da ontologia inexistentes na SNOMED CT são: “RiscoModeradamenteAumentado”, “RiscoAusente”, “CategoriasDosNiveisDeAlbuminuria”, “CondutaEstagio1”, “CondutaEstagio2”, entre outros. Os conceitos para os quais não foram encontrados termos correspondentes no repositório de termos do padrão SNOMED CT, estão identificados na ontologia por meio de uma *annotation* do tipo *boolean*. Para os conceitos que não foram padronizados devido ao problema relatado, o valor da *annotation* é *true*.

#### 4.1.2.4 Validação

A validação foi a última tarefa do estudo de caso. Nela o Nefrologista avaliou, com a ajuda da equipe responsável pela evolução, a estrutura, os conceitos e os axiomas presentes na ontologia resultante de modo a verificar sua consistência e, conseqüentemente, a sua confiabilidade ao ser aplicada no contexto para o qual foi criada. As inconsistências resultantes das alterações aplicadas bem como a literatura sobre o tema foram métodos de análise e estudos para a identificação de formas de resolução de inconsistências em ontologias após evolução. Assim, neste estudo de caso, as inconsistências da ontologia não foram corrigidas em sua totalidade.

#### 4.1.2.5 Resultados

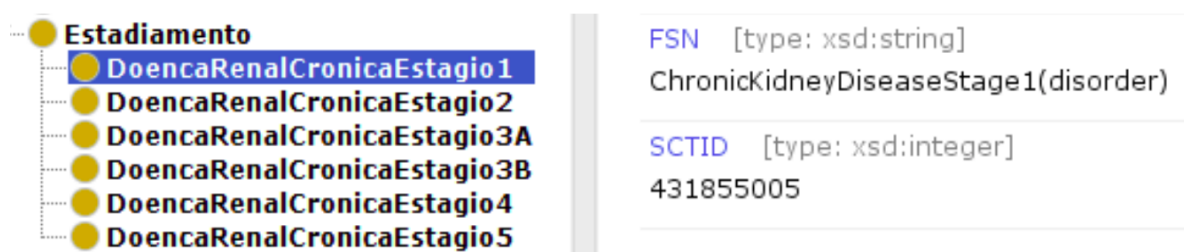
A seguir são apresentadas as principais classes da ontologia ontoDecideDRC que sofreram mudanças básicas e/ou complexas:

- A classe “Classificacao” foi renomeada para “Estadiamento”. A Figura 10 apresenta a estrutura hierárquica da classe Estadiamento. Na mesma também é possível observar as *annotations* criadas para a classe “DoencaRenalCronicaEstagio1” durante a padronização dos termos da Ontologia.

- A classe “MotivosDeEncaminhamentoMaisComuns”, adicionada à ontoDecideDRC durante o seu processo evolutivo apresenta como subclasses, doenças que também podem levar ao encaminhamento do paciente para o Nefrologista e que se apresentam como fatores de risco da DRC. A hierarquia da mesma pode ser observada na Figura 11.

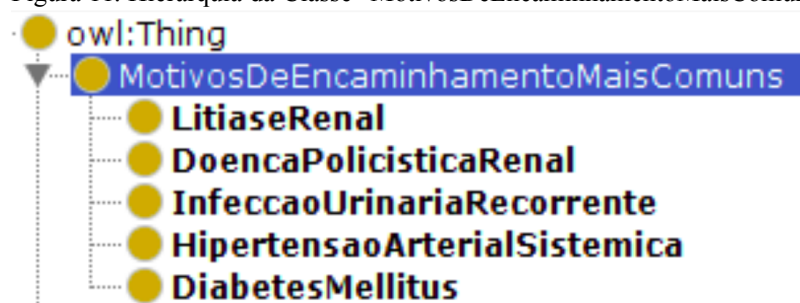
Nenhuma propriedade da ontologia sofreu alteração. A validação da ontologia resultante também foi realizada com o apoio do Nefrologista que considerou confiáveis os termos nela usados, tendo em vista que foram baseados em diretrizes mundialmente reconhecidas.

Figura 10. Hierarquia da Classe Estadiamento



Fonte: Protégé

Figura 11. Hierarquia da Classe "MotivosDeEncaminhamentoMaisComuns"



Fonte: Protégé

Na Figura 12 podem ser observadas as métricas das duas versões da ontologia. Devido à mudança de foco e conseqüentemente a redução do escopo da nova versão, a quantidade de classes e axiomas da nova versão são menores.

Figura 12. Métricas das duas versões da ontologia

Ontology metrics:	OntoDecideDR	ONTODRC
Metrics		
Axiom	1157	397
Logical axiom count	391	114
Declaration axioms count	257	110
Class count	187	49
Object property count	26	14
Data property count	4	7
Individual count	15	13
Annotation Property count	29	31
DL expressivity	SHIF(D)	SHIF(D)

Fonte: Adaptado da ferramenta Protégé.

#### 4.1.2.6 Conclusão

Após a evolução, a nova ontologia (ONTODRC) passou a refletir conceitos mais específicos em relação à sua versão anterior. Em vez de abordar todo o processo de tratamento da DRC, a ontologia foca nas fases iniciais da doença, com o intuito de minimizar falhas como diagnóstico impreciso e informações incorretas e inconsistentes no formulário de encaminhamento, e como consequência reduzir o retrabalho no tratamento da doença.

O estudo de caso apresentou como contribuição uma visão mais ampla da importância da evolução do conhecimento contido em ontologias, particularmente quando esse conhecimento é relativo ao suporte à decisão em saúde. A utilização da terminologia SNOMED CT permitiu uma visão mais abrangente da terminologia, bem como das suas vantagens na padronização de termos clínicos. A não existência dos termos específicos do cuidado clínico brasileiro direcionado a paciente com DRC, contribuiu para o início de novas investigações cujo objetivo foi encontrar modos de padronizar termos inexistentes na terminologia. A presença do médico especialista na realização do estudo de caso permitiu a definição precisa do motivo da evolução e foi imprescindível para a análise da ontologia resultante.

Portanto, pode-se concluir que os métodos utilizados para a realização do estudo de caso relatado neste artigo ofereceram subsídios eficientes para dar suporte ao processo de evolução de ontologias utilizadas para suporte à decisão no domínio da saúde. Como esperado, o estudo também proporcionou uma base teórica e prática para a formulação das tarefas que compõem o *framework* construído e detalhado nesta dissertação.

### 4.1.3 Construção do *framework* para suporte à evolução de ontologias Biomédicas

Após as pesquisas e análises dos trabalhos sobre o tema, foi realizada a construção do *framework* proposto. Para esse fim, inicialmente foram identificadas as fases de um processo convencional de evolução de ontologias. Após a identificação as mesmas foram analisadas com a finalidade de identificar em quais fases desse processo seria necessária a implementação de novas atividades para dar suporte ao processo evolutivo quando este ocorresse especificamente durante a evolução de ontologias Biomédicas. Essa etapa foi importante para a concepção do *framework* porque foi nela onde se identificou as fases do processo evolutivo onde deve haver maior suporte à evolução quando esta for voltada para ontologias Biomédicas. Para a implementação efetiva dessa tarefa foram consideradas as peculiaridades das ontologias Biomédicas de modo que o *framework* construído pudesse oferecer suporte considerando tais especificidades.

Conforme exposto na seção 2.3, onde as ontologias Biomédicas são detalhadamente abordadas, a principal diferença desse tipo de ontologias quando comparado com os demais é o fato de serem maiores e assim mais complexas. Desse modo, a maior atenção exigida pelas ontologias Biomédicas em determinadas fases do processo evolutivo está estritamente relacionada à sua maior complexidade ocasionada pelo maior volume de conceitos representado. Assim, basicamente todas as fases da evolução de ontologias requerem algum cuidado adicional quando se tratar da evolução de ontologias Biomédicas. Baseando-se na literatura pertinente sobre o tema, foram identificadas como fases de um processo convencional de Evolução de ontologias, a Representação Semântica das Mudanças, onde estão, formalmente representadas as mudanças requeridas por usuários; a Propagação dessas mudanças nos elementos dependentes da ontologia em evolução, fase que tem como finalidade, dentre outras, a manutenção da consistência da ontologia. A implementação das mudanças na ontologia é a última fase de um processo evolutivo com atividades chave para o alcance da finalidade da evolução corrente. Nela as alterações são efetivamente aplicadas e utilizadas pelos serviços que implementam conceitos da ontologia em evolução.

Tendo em vista que o *framework* proposto pretende tornar mais claro e compreensivo o processo evolutivo das ontologias Biomédicas, novas fases foram definidas e as existentes foram reimplementadas com melhorias refletidas na criação de atividades específicas pertencentes a determinada fase de modo a dar maior ênfase à evolução de ontologias

Biomédicas. O maior volume de conceitos representados nas ontologias Biomédicas também causa um número maior de inconsistências, desse modo, com vistas a fornecer maior enfoque no tratamento de inconsistências, no *framework* proposto, foi criada uma nova fase para realizar especificamente tarefas cuja finalidade é constituída pela manutenção da consistência da ontologia em evolução.

Existem três motivos mais comuns que ocasionam a evolução de ontologias. Tais motivos são: a necessidade de mudança de foco da ontologia, mudanças na diretrizes do domínio representado na ontologia que conseqüentemente levam à necessidade de atualização do conhecimento constante na mesma, e mudanças na terminologia de referência da ontologia que torna necessária a atualização terminológica da ontologia. Diante desses motivos e do impacto dos mesmos nas atividades a serem realizadas durante o processo evolutivo, o *framework*, cujo processo metodológico é aqui descrito, foi concebido com base nos mesmos com o intuito de fornecer apoio à evolução de ontologias Biomédicas que forem submetidas à evolução devido a qualquer um dos três motivos citados.

Desse modo, o *framework* foi construído considerando três perspectivas de evolução baseadas nos três motivos mais comuns. As atividades a serem aplicadas durante a evolução da ontologia foram pensadas para dar suporte o processo evolutivo conforme o motivo pelo qual a mesma está evoluindo. Por exemplo, caso a ontologia esteja em processo de evolução por mudanças em sua terminologia de referência, dois caminhos poderão ser seguidos de acordo com o contexto no qual a mesma esteja inserida. Se a ontologia já possui uma terminologia de referência, o processo evolutivo deverá ser iniciado com a captura das mudanças ocorridas na terminologia para que as mesmas sejam refletidas na ontologia. Por outro lado, caso a ontologia não possua uma referência para padronização terminológica, estará sendo realizada a padronização inicial e total da ontologia e tal processo deverá ser iniciado com a busca dos termos da ontologia na terminologia escolhida como referência, de modo a se identificar os termos existentes ou propor novos termos caso seja necessário e a terminologia escolhida fornecer subsídios para isso.

Para o *framework* construído foram definidas 5 etapas. A primeira, nomeada como Planejamento da Evolução, constitui-se de atividades voltadas para a captura e representação sintática e semântica das mudanças a serem aplicadas. Essa etapa fornece como resultado um relatório em linguagem natural com todas as mudanças que devem ser aplicadas na ontologia



e suas devidas descrições. A segunda etapa foi chamada de Implementação das Mudanças, nela as mudanças representadas podem ser implementadas por meio da edição da ontologia em evolução. Tal etapa deve ser apoiada por um *software* de edição de ontologias. A terceira etapa é a primeira que possui a finalidade de estabelecer o controle do processo evolutivo. Chamada de Detecção das Mudanças, nesta fase, as mudanças aplicadas são visualizadas de modo que erros cometidos durante a aplicação das mudanças sejam identificados e corrigidos antes de serem iniciadas as etapas mais avançadas do processo evolutivo como Tratamento de Inconsistências e Auditoria de Mudanças. Além da identificação de erros ocasionados pela implementação das mudanças planejadas, a Detecção das Mudanças possui como finalidade a minimização da ocorrência de inconsistências na estrutura taxonômica e lógica da ontologia.

Quarta fase do processo evolutivo, o processo de concepção da fase de tratamento de inconsistências se deu inicialmente pela criação de três atividades denominadas, identificação, classificação e resolução de inconsistências. Para serem classificadas corretamente, todas as mudanças efetivamente aplicadas precisam ser detectadas, assim, a fase de Detecção de Mudanças é essencial para a realização das tarefas da fase de Tratamento de Inconsistências. Tais tarefas foram definidas e refinadas verificando-se desse modo a necessidade da criação de possíveis subtarefas que foram responsáveis por transformar as atividades dessa fase em atividades constituídas por subtarefas condizentes com a finalidade de cada uma. Por exemplo, na identificação de inconsistências, uma tarefa chave para a identificação dos problemas nas entidades da ontologia em evolução é a identificação das mudanças realizadas, sabendo-se quais mudanças foram feitas é possível se identificar quais entidades foram afetadas por elas. Desse modo a identificação das mudanças aplicadas passou a ser uma tarefa de extrema relevância para a tarefa de identificação de inconsistências. Seguindo esse mesmo método, foram definidas as subtarefas das atividades da fase de tratamento de inconsistências e das demais fases que constituem o *framework*. Por exemplo, a Classificação de Mudanças, outra subtarefa constituinte da fase de Tratamento de Inconsistências é dependente da identificação das inconsistências e tem como finalidade a determinação da ordem de resolução de inconsistências.

As inconsistências podem ser classificadas em estruturais ou lógicas. As inconsistências estruturais são aquelas que afetam a estrutura hierárquica da ontologia, enquanto as lógicas afetam as regras que determinam o sentido do conhecimento representado

na ontologia. Em geral, os problemas lógicos são decorrentes dos problemas estruturais decorrentes das mudanças aplicadas na ontologia. Por esse motivo, recomenda-se a resolução das inconsistências estruturais antes da resolução das inconsistências lógicas de modo a se reduzir o desperdício de tempo com a resolução de problemas que podem ser automaticamente resolvidos após a correção dos problemas estruturais. Por exemplo, uma ontologia sobre a DRC em evolução por mudança de foco sofreu a deleção do conceito “Nefrologista”. Naturalmente, as regras que determinavam os responsáveis por conduzir o paciente de acordo com o estágio da doença no qual o mesmo se encontra, ficarão inconsistentes e, conseqüentemente as inferências da ontologia não poderão ser realizadas da maneira correta. Com a mudança aplicada, as regras da ontologia que fazem referência à classe removida sofreram inconsistências estruturais pois dizem respeito a um conceito inexistente na estrutura taxonômica da ontologia. A resolução desse problema pode ser feita de duas maneiras: alteração da regra para fazer referência a outro conceito, por exemplo, o conceito “Nefrologista” pode ser substituído por “Médico da Atenção Primária”; ou a deleção do axioma. Ao aplicar a resolução desse problema estrutural, conseqüentemente, os erros decorrentes da referenciação do conceito inexistente serão removidos e a ontologia poderá inferir corretamente sobre o conhecimento em questão.

A quinta fase do *framework* foi chamada de Auditoria de Mudanças e é a segunda fase que foi criada com a finalidade de manter o controle do processo evolutivo. A Auditoria de Mudanças é uma etapa na qual são realizadas tarefas de inspeção das mudanças realizadas. O relatório de mudanças planejadas é utilizado como guia da atividade de auditoria de modo que se possa realizar comparações entre o que foi planejado e o que realmente foi implementado.

O fornecimento necessário de um relatório no fim de cada fase do *framework* contendo informações relativas às atividades executadas em cada uma delas, foi pensado como uma alternativa de oferecer informações relevantes para a realização das demais fases tendo em vista que as mesmas estão interligadas e dependem uma da outra para se alcançar o objetivo final da *framework*. Por exemplo, na fase de Tratamento de Inconsistências, para classificá-las como estruturais ou lógicas é importante que se saiba quais inconsistências ocorreram. Para que isso seja possível, um relatório contendo as inconsistências identificadas deve ser fornecido como entrada para a tarefa de classificação. Similarmente, para que as

inconsistências da ontologia sejam resolvidas na atividade de resolução de inconsistências, as mesmas devem ser dadas como entrada para referida fase.

Os relatórios também poderão ser úteis em evoluções posteriores auxiliando na correção de problemas por meio do uso de estratégias aplicadas em problemas similares ocorridos em evoluções anteriores, ou mesmo na definição de novas estratégias baseadas nas aplicadas anteriormente. Esse tipo de ação pode reduzir o tempo gasto com a busca por soluções já existentes reduzindo assim o tempo total usado para a realização do processo evolutivo.

## **4.2 Validação do *Framework***

Um estudo de caso foi realizado com vistas a simular a aplicabilidade do *framework* e assim, validá-lo. Desse modo, a ontoDecideDRC (TAVARES, 2016), ontologia que foi utilizada no estudo de caso realizado com a finalidade de obter subsídios para a construção do *framework* proposto nesta dissertação, passou por todas as fases que compõem o *framework* proposto, diante das três perspectivas de evolução identificadas. Tais perspectivas foram definidas a partir dos motivos de evolução detectados e descritos na seção 4.1.3. Desse modo, três versões da ontologia ontoDecideDRC foram submetidas a evolução. Cada uma das versões foi evoluída por um motivo distinto. A primeira, por mudança de foco, a segunda por mudanças nas diretrizes do domínio representado na ontologia e a terceira por mudanças na terminologia de referência da ontologia. Ao fim da evolução acompanhada pelo *framework*, testes foram aplicados de modo a verificar a consistência da ontologia ao responder questões relacionadas ao domínio representado.

Para a realização desses testes foram utilizadas consultas SPARQL, linguagem de consulta para *Resource Description Framework* (RDF) tendo em vista que a OWL foi a linguagem utilizada para o desenvolvimento da ontologia. As consultas SPARQL foram criadas com base em Questões de Competência, que são questões relativas ao domínio da ontologia e que devem ser respondidas pela mesma caso esta tenha sido corretamente desenvolvida (BORTOLATO, 2014). As QCs são particularmente úteis pois ajudam tanto no estabelecimento de um limite em relação ao escopo da ontologia como no seu processo de validação (ISOTANI e BITTENCOURT, 2015).

A linguagem de consulta SPARQL foi escolhida como a linguagem formal para realizar a validação da ontologia por possuir alta expressividade, o que possibilita a

representação de um número maior de QCs (BORTOLATO, 2014) . No estudo de caso é feito o uso de ferramentas de *software* para facilitar o entendimento do *framework* proposto, entretanto, tais ferramentas são sugestões que podem ou não ser utilizadas pelos ontologistas durante o processo de evolução.

## Capítulo 5

### 5 *Framework* para suporte à evolução de ontologias Biomédicas

#### 5.1 Definições Preliminares e Anotações

**Definição 1.** A Engenharia de ontologias possui como finalidade primordial, dar suporte ao processo de criação de ontologias que sejam adequadas para atender as necessidades relativas à representação e uso correto da informação (ISOTANI e BITTENCOURT, 2015).

**Definição 2.** O Engenheiro de ontologias é o profissional responsável pelo processo de criação de ontologias.

**Definição 3.** OWL é um padrão de linguagens para construção de ontologias, desenvolvido pela *World Wide Web Consortium (W3C)*<sup>18</sup>. De acordo com a sublinguagem utilizada, a OWL pode ser classificada em três tipos que se diferenciam pelo nível de expressividade: *OWL-Lite*, *OWL-DL*, *OWL-Full* (HORRIDGE *et al.*, 2011).

**Definição 4.** Em OWL, indivíduos ou instâncias representam objetos no domínio da ontologia ou no domínio do discurso. Indivíduos também podem ser referenciados como instâncias de classes (HORRIDGE *et al.*, 2011).

**Definição 5.** Em OWL, propriedades determinam as relações entre dois indivíduos. Por exemplo, a propriedade *temIrmao* pode ligar o indivíduo *Ana* ao Indivíduo *Marilia* (HORRIDGE *et al.*, 2011).

**Definição 6.** Classes são representações concretas de conceitos. Em OWL, classes são conjuntos de indivíduos. Por vezes, o termo conceito é usado no lugar do termo classe. Classes podem ser organizadas em hierarquias onde as mesmas poderão ser classificadas em superclasses ou subclasses. Tais hierarquias são também chamadas de taxonomias (HORRIDGE *et al.*, 2011).

**Definição 7.** Uma ontologia  $O \equiv \{E, R\}$ , onde:

<sup>18</sup><https://www.w3.org/>

- *E* representa o conjunto de entidades de uma ontologia *O*
- *R* representa os relacionamentos de uma ontologia *O*

**Definição 8.** Uma entidade  $e \equiv \{C, P, I, A\}$ , onde:

- *C* é o conjunto de conceitos de uma ontologia *O*
- *P* é o conjunto de propriedades de uma ontologia *O*
- *I* é o conjunto de instâncias de uma ontologia *O*
- *A* é o conjunto de axiomas de uma ontologia *O*

**Definição 9.** Uma mudança *m* consiste em uma alteração aplicada a uma ou mais entidades de uma ontologia *O* como ação necessária para o alcance da finalidade do processo evolutivo dessa ontologia.

**Definição 10.** Mudanças podem ser classificadas como simples ou complexas.

**Definição 11.** Mudanças simples são caracterizadas por alterações que necessitam de apenas uma ação para atingir seu objetivo. Por exemplo, operações de adição e deleção (KLEIN e NOY, 2003).

**Definição 12.** Mudanças complexas são caracterizadas por alterações que necessitam de mais de uma ação para atingir seu objetivo. Por exemplo, a operação de mover conceitos, que é composta pela remoção e posterior adição de um conceito em outro local da ontologia (KLEIN e NOY, 2003).

**Definição 13.** Uma mudança *m* pode ser rejeitada quando não satisfizer a finalidade para a qual foi especificada, foi incorretamente aplicada ou não fazia parte do plano de evolução.

**Definição 14.** Uma mudança *m* pode ser invalidada se, mesmo após as inconsistências da ontologia terem sido tratadas, identificou-se que a mesma teve sua consistência alterada e passou a não fornecer respostas para questões que justificam sua existência.

**Definição 15.** Uma mudança *m* semanticamente representada é aquela cuja finalidade para a qual foi requerida está explicitamente clara e pode ser entendida pelos responsáveis pela evolução sem prejuízo de ser mal interpretada.

**Definição 16.** A versão anterior de uma ontologia  $Va$  consiste na ontologia em evolução antes de serem aplicadas quaisquer mudanças em sua estrutura.

**Definição 17.** A versão atual ou nova versão de uma ontologia  $Vn$  consiste na ontologia em evolução depois de aplicadas as alterações requeridas presentes no plano de evolução.

**Definição 18.** Uma mudança  $m$  é detectada quando comparando-se a versão anterior  $Va$  de uma ontologia com a sua nova versão  $Vn$  são identificadas mudanças, sejam elas requeridas ou induzidas, em uma ou mais entidades da ontologia em evolução.

**Definição 19.** Uma condição de consistência  $\beta$  é uma verdade absoluta ou axioma determinado com base em um domínio específico a partir do qual uma entidade  $e \in E \in O$  é considerada consistente.

**Definição 20.** Uma ontologia  $O$  é considerada inconsistente se  $\forall e \in E \in O, e \text{ não satisfaz } \beta$ , onde  $e$  é uma entidade do conjunto de entidades  $E \in O$ .

**Definição 21.** A lógica de uma ontologia  $O$  é determinada por um conjunto de condições de consistência  $T$  onde existem várias entidades  $e \in E$ .

**Definição 22.** Uma inconsistência  $i$  é detectada quando alguma entidade  $e \in E \in O$  afeta a estrutura ou impede o raciocínio correto de uma ontologia  $O$  na qual foram aplicadas mudanças.

**Definição 23.** Um conjunto de entidades é inconsistente se havendo um método que detecta inconsistências que recebe uma ontologia  $O$  como parâmetro o retorno desse método é  $\alpha$ , onde  $\alpha = \{E \mid \forall e \in E \nexists e \in \beta\}$ , lê-se:  $\alpha$  é igual a um conjunto de entidades  $E$  tal que para toda entidade pertencente a esse conjunto não existe entidade que satisfaz a qualquer condição de consistência.

**Definição 24.** Uma inconsistência  $i$  de uma entidade  $e \in E \in O$  pode ser classificada em estrutural  $s$  ou lógica  $l$ . Conforme definido por Haase *et al.* (2005) nas inconsistências estruturais os construtores da linguagem utilizada no desenvolvimento da ontologia são afetados pelas mudanças. Nas inconsistências lógicas, as regras que definem a consistência da ontologia, geralmente definidas por meio de axiomas, sofrem alterações devido às mudanças e passam a não satisfazer os requerimentos definidos pra o seu domínio.

**Definição 25.** Erros de inconsistências são erros que afetam as condições de consistência  $T$  de uma ontologia  $O$  e causam ambiguidades em sua estrutura.

**Definição 26.** Erros de inconsistências podem ser classificados como erros circulatorios, erros de partição ou erros de inconsistência semântica (FAHAD e QADIR, 2008).

**Definição 27.** Erros circulatorios ocorrem quando uma classe de uma ontologia  $O$  em evolução é definida como subclasse de si mesma em qualquer nível da hierarquia dessa ontologia. Por exemplo, em uma ontologia sobre a DRC em evolução por mudança de foco, um erro circulatorio ocorrerá caso a classe “Nefrologista” seja definida como subclasse da classe “Medico” e a classe “Medico” seja definida como subclasse da classe “Nefrologista”. Tendo em vista que em OWL podem ser criadas hierarquias de propriedades, erros circulatorios também podem ocorrer nesse tipo de hierarquia (FAHAD e QADIR, 2008).

**Definição 28.** Erros de partição dependem do tipo de decomposição de superclasses em subclasses, que pode ser disjunta ou exaustiva (FAHAD e QADIR, 2008).

**Definição 29.** A decomposição disjunta ocorre quando as características das subclasses são descritas de forma independente (FAHAD e QADIR, 2008).

**Definição 30.** A decomposição exaustiva ou completa ocorre quando ontologistas seguem corretamente as restrições de completude entre subclasses e superclasses (FAHAD e QADIR, 2008).

**Definição 31.** Erros de partição ocasionados pela decomposição disjunta ocorrem quando ontologistas criam instâncias que pertencem a diversas subclasses disjuntas ou quando definem uma classe como uma subclasse de classes disjuntas. Por exemplo, em uma ontologia sobre a DRC, ocorrerá um erro de partição por decomposição disjunta caso, para a classe “Medico”, o ontologista defina as subclasses “Nefrologista” e “MedicoDaAtencaoPrimaria” como classes disjuntas e atribua a elas, um indivíduo comum (FAHAD e QADIR, 2008).

**Definição 32.** Erros de partição ocasionados pela decomposição exaustiva ocorrem quando ontologistas fazem uma decomposição ou partição exaustiva de uma classe em muitas subclasses mas nem todas as instâncias da classe base pertencem às suas subclasses. Ou seja, quando uma ou mais instâncias da classe base ou raiz não pertencem a todas as suas subclasses. Por exemplo, em uma ontologia sobre a DRC, ocorrerá um erro de partição por



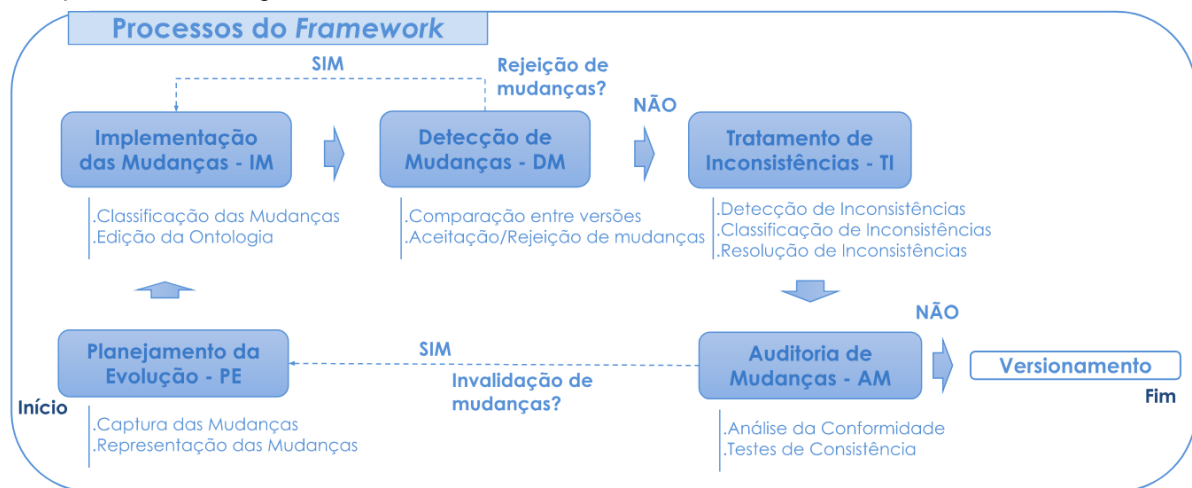
decomposição exaustiva caso, para a classe “Exames” que possui como subclasses “TaxaDeFiltracaoGlomerular” e “HemogramaCompleto”, o ontologista defina uma instância chamada “ExamesDeRotina” (FAHAD e QADIR, 2008).

**Definição 33.** Erros de inconsistência semântica ocorrem quando ontologistas criam uma hierarquia incorreta de classes por meio da classificação de determinado conceito como subclasse de outro ao qual esse conceito não pertence. Ou seja, um conceito será definido como subclasse de um conceito que não está relacionado a ele. Por exemplo, em uma ontologia sobre a DRC, ocorrerá um erro de inconsistência semântica caso se defina a entidade “Crianca” como uma instância da classe “Paciente” (FAHAD e QADIR, 2008).

## 5.2 Processos do *Framework*

Na Figura 13 é mostrada a forma como está organizado o *framework* proposto. Conforme a Figura, as fases do *framework* estão organizados como etapas do processo evolutivo de uma ontologia, consideradas pela literatura como essenciais.

Figura 13. Processos do *framework* para suporte à evolução de ontologias Biomédicas. Os retângulos com bordas arredondadas representam os subprocessos que compõem o framework. As setas com linhas pontilhadas denotam a necessidade de retorno a determinado subprocesso. As setas preenchidas representam a ordem normal de execução de cada subprocesso.



Fonte: Autoria própria

A primeira etapa, chamada de Planejamento da Evolução ou PE, trata da captura e da representação semântica das mudanças que devem ser realizadas na ontologia, é nessa etapa que as mudanças que se pretende realizar devem ser entendidas para que se possa visualizar seus impactos e formular as ações que devem ser tomadas diante dos mesmos.

No PE as mudanças que devem ser realizadas são identificadas e representadas de forma clara para o Engenheiro de ontologias e para os demais envolvidos no processo de evolução. O processo de identificação é chamado neste trabalho de Captura da Mudança, enquanto a representação semântica das mesmas após sua identificação é chamada de Representação da Mudança. A realização dessas atividades é dependente do motivo pelo qual a ontologia está evoluindo pois os caminhos a serem seguidos durante a evolução devem ser adaptados a ele.

A partir de um estudo de caso (4.1.2) realizado para levantar requisitos úteis para a construção do *framework* do qual trata esta dissertação, foram identificados três motivos possíveis de evolução:

- Mudança do foco da ontologia: Caso este seja o motivo que levou a ontologia a evoluir, as mudanças realizadas são ocasionadas pela representação de novos conhecimentos de domínio, tendo em vista que o foco da ontologia em relação ao conteúdo representado passou a ser outro.
- Mudanças nas diretrizes da ontologia: Uma ontologia estará evoluindo por esse motivo caso sejam necessárias mudanças em sua estrutura taxonômica ou lógica motivadas por alterações em suas fontes de conhecimento.
- Mudanças na terminologia de referência da ontologia: Caso tenham sido realizadas mudanças nos termos relativos ao domínio representado na ontologia, a mesma deverá refletir essas mudanças.

O PE fornece como saída um relatório escrito em linguagem natural contendo todas as mudanças a serem aplicadas na ontologia em processo evolutivo. O conteúdo desse relatório também deverá ser adequado ao motivo pelo qual a ontologia está sendo submetida a evolução.

Na Implementação das Mudanças (IM), etapa do processo evolutivo e subprocesso do *framework* (seção 5.2.2), as mudanças planejadas poderão ser realizadas de modo mais compreensivo após as mudanças requeridas terem sido planejadas. A finalidade do fornecimento de um relatório estruturado como saída da etapa de planejamento, é direcionar o engenheiro de ontologias durante o processo de edição em relação a questões como: quais

mudanças devem ser realizadas primeiro, quais as ações necessárias para que a mudança seja implementada e quais os impactos de cada mudança aplicada.

Na IM as mudanças planejadas serão aplicadas na ontologia por meio do uso de um editor de ontologias. A aplicação das mudanças também deverá ser voltada para a perspectiva de evolução definida pelo motivo pelo qual a ontologia está evoluindo. Por exemplo, o engenheiro de ontologias responsável pela evolução de uma ontologia que está passando por esse processo devido à mudança do seu foco, deverá decidir entre aplicar as mudanças adaptativas em conceitos que deverão ser mantidos ou adicionar informações relativas ao novo foco. Por outro lado, caso o motivo da evolução seja as mudanças ocorridas nas diretrizes representadas pela ontologia, as alterações a serem realizadas deverão considerar as mudanças taxonômicas ou lógicas que devem ser implementadas com a finalidade de refletir as mudanças nas diretrizes ou se novas diretrizes do domínio representado foram adicionadas.

Após aplicadas, as mudanças devem ser encaminhadas para análise, atividade realizada no subprocesso Detecção de Mudanças (DM) (seção 5.2.3). Caso existam mudanças rejeitadas por não atenderem ao plano de evolução (relatório de mudanças requeridas), será necessário retornar ao subprocesso anterior (seta com linha pontilhada iniciada no subprocesso Detecção de Mudanças), caso contrário, inicia-se o tratamento das inconsistências da ontologia aplicando as atividades do subprocesso Tratamento de Inconsistências (TI).

Para que sejam determinadas formas de resolver esses problemas, as causas dos mesmos devem ser identificadas. Assim, antes de se iniciar o tratamento das inconsistências é importante que se saiba quais mudanças foram efetivamente realizadas, por isso, o subprocesso anterior, onde as mudanças são detectadas, fornece como saída um relatório de mudanças aplicadas. Sabendo-se o que foi realizado na ontologia e conseqüentemente a localização de cada alteração, a identificação de possíveis problemas que tenham decorrido dessas mudanças pode ser uma tarefa menos onerosa. Após identificadas, as inconsistências podem ser resolvidas de acordo com a causa das mesmas. Por exemplo, um conceito deveria ter sido classificado como disjuncto de outro conceito, porém, isso não aconteceu. Tal fato gerou uma inconsistência de taxonomia na ontologia. A correção desse problema está diretamente relacionada à causa do problema, pois uma relação de disjunção deve ser

estabelecida entre os dois conceitos de modo que se estabeleça a consistência nessa parte da ontologia.

Após ter suas inconsistências tratadas a ontologia é encaminhada para o subprocesso Auditoria de Mudanças (AM). Nele as mudanças aplicadas na ontologia serão examinadas considerando sua correteza e completude em relação ao plano de evolução. Nesta dissertação, o plano de evolução também será chamado de relatório de mudanças a serem aplicadas ou relatório de mudanças representadas. A verificação da consistência da ontologia com base em suas Questões de Competência também deverá ser realizada nesta etapa. Caso a consistência da ontologia tenha sido mantida a mesma poderá ser encaminhada para o versionamento.

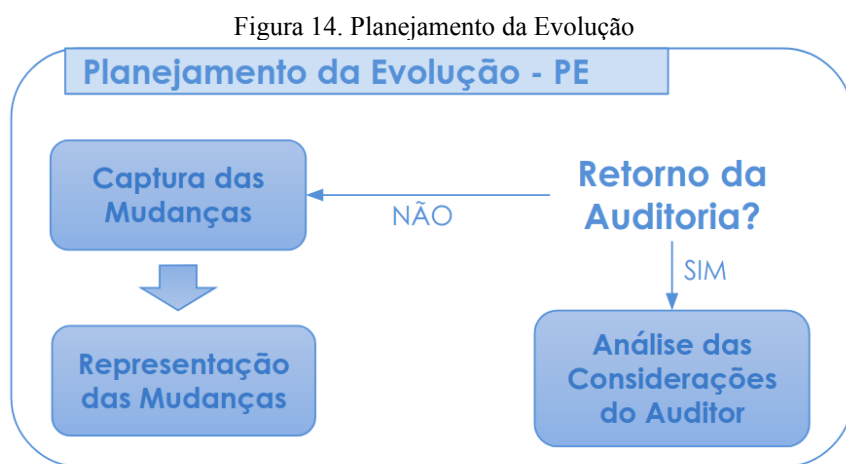
Cada subprocesso do *framework* possui um relatório de saída, escrito em linguagem natural, que fornece subsídios para a realização das atividades dos demais subprocessos e poderão ser úteis em caso de correção de erros de evolução bem como em evoluções posteriores. Após a auditoria, atividade realizada no último subprocesso do *framework* (AM), é fornecido um relatório de considerações do auditor produzido com base na avaliação comparativa entre a ontologia alterada e o relatório de mudanças requeridas. Em caso de invalidação de mudanças por problemas de inconformidade com o plano de evolução, a ontologia deverá ser re-encaminhada para a primeira etapa do processo evolutivo (PE) e todo o ciclo necessário para a sua evolução deverá ser realizado novamente. Caso a ontologia esteja em conformidade com os requisitos constantes no relatório de mudanças requeridas, a mesma é encaminhada para o versionamento, tarefa onde os responsáveis pela sua evolução devem escolher o melhor modo de versioná-la observando aqueles que favorecem o tipo de desenvolvimento (isolado ou colaborativo) da mesma. Nos casos em que a ontologia for desenvolvida colaborativamente é importante que a ferramenta usada para o seu versionamento permita o acesso simultâneo de colaboradores bem como a imediata disponibilidade das alterações recém aplicadas por determinado colaborador para serem visualizadas e acessadas pelos demais. Nesta dissertação, o processo de versionamento não será abordado detalhadamente.

Nas próximas subseções cada subprocesso do *framework* é detalhadamente explicado. Nas figuras que representam tais subprocessos, os retângulos com bordas arredondadas representam as ações que podem ser executadas no mesmo. O monitor denota a importância

do uso de *softwares* para apoiar a tarefa junto à qual está disposto. As setas demonstram a ordem de ocorrência da análise das entradas, execução das tarefas e produção das saídas.

### 5.2.1 Planejamento da Evolução - PE

Como definido na Figura 13 o processo de evolução deve ser iniciado com o planejamento (Figura 14). Neste subprocesso duas ações podem ser executadas dependendo do estágio da evolução em que a ontologia se encontra. Caso a ontologia esteja passando pelo planejamento pela primeira vez durante o processo evolutivo atual, ou seja, caso não esteja retornando ao planejamento por inconformidade com o plano de evolução detectada durante o subprocesso Auditoria de Mudanças (AM), deverá ser realizada a primeira tarefa da etapa de Planejamento da Evolução: a Captura das Mudanças.



Fonte: Autoria própria

#### 5.2.1.1 Captura das Mudanças

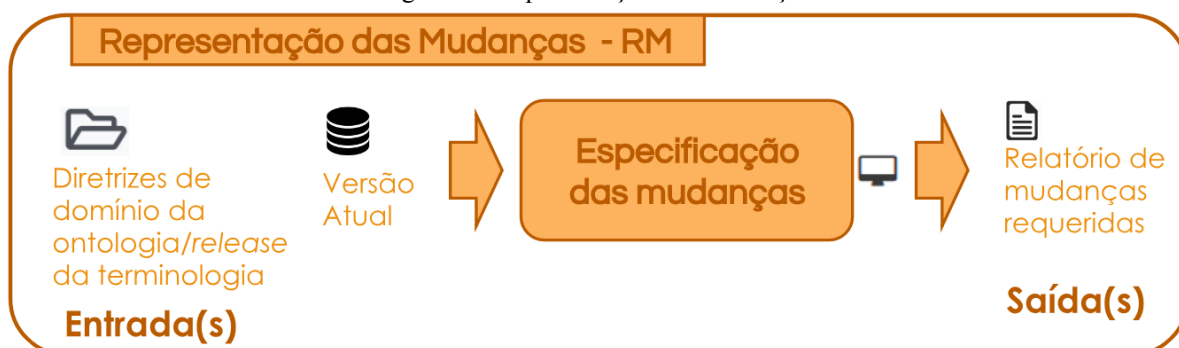
Na Captura das Mudanças, como o próprio nome sugere, é realizada a coleta das mudanças a serem realizadas. Nessa atividade deve-se identificar quais mudanças devem ser feitas. As mudanças a serem feitas devem levar em conta o motivo da evolução, desse modo, as alterações que devem ser aplicadas fundam-se em decisões relativas ao motivo da evolução, por exemplo, quando a ontologia está sendo evoluída pela necessidade de mudança em seu foco, as mudanças a serem aplicadas giram em torno da manutenção ou não de determinados conceitos que podem ou não ser adaptados ao novo domínio ou; da inclusão de novos conceitos relativos a esse domínio.

Considerando essa relação de dependência entre as decisões que devem ser tomadas durante a evolução da ontologia e o motivo da evolução, a identificação do motivo pelo qual a ontologia está em processo evolutivo é a primeira tarefa a ser realizada durante a captura das mudanças com o intuito de direcionar o processo evolutivo que se inicia. Identificado o motivo pelo qual a ontologia está sendo evoluída, as mudanças podem ser sintática e semanticamente representadas diante desse motivo.

### 5.2.1.2 Representação das Mudanças

A Representação das Mudanças (Figura 15) é a atividade do processo evolutivo onde as mudanças que devem ser aplicadas precisam ser entendidas pelos ontologistas. Denominada como Especificação das mudanças, a atividade realizada na tarefa de representação recebe como entrada a ontologia em sua versão atual (antes do início do processo de edição) e suas fontes de conhecimento. Especificar as mudanças, neste contexto, se relaciona com a identificação dos tipos de alterações que devem ser realizadas na ontologia.

Figura 15. Representação das Mudanças



Fonte: Autoria própria

Além da representação textual em linguagem natural das mudanças a serem feitas, outros fatores devem ser observados e especificados claramente de modo entendível pelos responsáveis pelo processo evolutivo. Tais fatores incluem os impactos das mudanças na estrutura taxonômica e lógica da ontologia em evolução, ou seja, quais entidades da ontologia serão afetadas pelas mudanças aplicadas e; a ordem que deve ser seguida para atingir o objetivo esperado pela mudança, por exemplo, se um conceito deve ser movido para outro local na ontologia, primeiro ele deverá ser removido de onde ele se encontra atualmente e

depois adicionado ao novo local, assim evita-se inconsistências relativas à duplicação de conceitos. Nesse exemplo, uma sequência de ações deverá ser executada para que seja possível se atingir o objetivo dessa alteração que é a alteração do local do conceito no qual a mudança está sendo aplicada.

Identificar, dentre as mudanças requeridas, quais são mudanças estruturais e quais são mudanças lógicas, também é um fator que deve ser levado em consideração durante a representação da mudança, pois de acordo com essa identificação será possível decidir quais mudanças devem ser aplicadas primeiro, tendo em vista que recomenda-se que mudanças estruturais sejam aplicadas antes, porém, se não existirem mudanças estruturais a serem aplicadas, as mudanças lógicas podem ser realizadas primeiro.

Diversos fatores devem ser levados em conta durante o processo de representação das mudanças, porém, esses fatores podem ser melhor especificados quando se define claramente por qual motivo a ontologia está sendo submetida a evolução. Nesta dissertação, a evolução será abordada diante de três perspectivas advindas de três possíveis motivos de evolução identificados a partir de um estudo de caso realizado no contexto deste trabalho de pesquisa para a obtenção de requisitos que dessem suporte à construção do *framework*. Tais motivos são a mudança de foco da ontologia, mudanças nas diretrizes nela representadas e mudanças na terminologia de referência da ontologia. Nas próximas subseções, aborda-se a tarefa de Especificação das Mudanças diante de três possíveis caminhos a serem seguidos no processo evolutivo decorrentes dos três motivos de evolução citados.

### **I. Especificação das Mudanças de uma ontologia em evolução por mudança de foco**

A primeira tarefa a ser realizada para especificar as mudanças nesta perspectiva de evolução é a de identificação do novo foco da ontologia. Ao se determinar o novo foco da ontologia pode-se então identificar as fontes de conhecimento do mesmo. Considerando uma ontologia do domínio da saúde, suas fontes de conhecimento podem ser diretrizes clínicas ou protocolos a serem observados no tratamento de pacientes portadores de determinada patologia, bases de dados gerais em saúde, acervos em saúde abertos ao público ou privados de uso público, por exemplo, acervos pertencentes a instituições hospitalares; bibliotecas em saúde validadas e internacionalmente utilizadas como NCIIt (do inglês, *National Cancer Institute Thesaurus*), terminologias clínicas como SNOMED CT e ICD-11 (do inglês,

*International Classification of Disease – 11th Revision*)<sup>19</sup>, entre outros. Tanto para o domínio da saúde como para os demais domínios, outras ontologias também podem ser utilizadas como fontes de conhecimento.

Diante do motivo de evolução em discussão, o Engenheiro de ontologias deverá decidir entre manter conceitos e adaptá-los à realidade do novo foco da ontologia, adicionar conceitos relativos a esse novo foco ou remover aqueles que não serão mais necessários. É importante esclarecer que, neste caso o que se denomina como conceito, compreende o conteúdo acerca de determinado domínio existente em uma ontologia, que em OWL podem ser classificados como Classes, Instâncias ou Propriedades. Assim, as mudanças a serem aplicadas em uma ontologia em evolução por mudança de foco, podem ser representadas como um conjunto de conceitos que pode ser formalmente especificado como segue:

$$C_{mf} \equiv \{c_{md}, c_a, c_r\}$$

Onde,

- $C_{mf}$  (mudança de foco) – Representa os conceitos a serem considerados durante a especificação das mudanças de uma ontologia em evolução por mudança de foco. Tais conceitos podem ser classificados em:
  - $c_{md}$  (modificados) – Representa os conceitos que serão mantidos, porém sofrerão algum tipo de mudança de modo a serem adaptados ao novo foco. Por exemplo, em uma ontologia sobre a DRC, o conceito “Medico” possui como subclasses “MedicoDaAtencaoPrimária” e “Nefrologista”. Tendo em vista que após evolução tal ontologia passou a abarcar conceitos relativos apenas à atenção primária, o conceito “Medico” poderia então ser substituído por “MedicoDaAtencaoPrimaria”, que seria um conceito existente a ser mantido mesmo diante do novo foco da ontologia, haveria dessa forma, a adaptação de um conceito existente ao novo foco da ontologia.
  - $c_a$  (adicionados) – Representa os conceitos que devem ser adicionados à ontologia em decorrência do novo foco da mesma. Tomando-se como exemplo a mesma

19 ICD-11 é a décima primeira edição da Classificação Internacional de Doenças e Problemas de Saúde Relacionados, criada e mantida pela WHO ( do inglês, *World Health Organization*) (WHO, 2018)



ontologia usada no exemplo do tópico anterior, a adição de conceitos relativos à conduta de pacientes antes de serem encaminhados para a atenção especializada são tipos de conceitos que deverão ser incluídos em uma ontologia em evolução por mudança de foco.

- $c_r$  (removidos) – Representa os conceitos que serão removidos da ontologia por não serem mais necessários em decorrência do novo foco da mesma. Uma ontologia sobre a DRC que aborda conceitos relativos a todos os estágios da doença, não precisará continuar representando todos esses estágios considerando que a mesma teve seu foco especificado para as práticas de cuidado realizadas apenas durante a atenção primária do paciente. Desse modo, os conceitos relativos aos estágios mais avançados da doença: 3B, 4 e 5, não precisarão mais ser representados nessa ontologia, podendo, portanto, ser removidos. Assim, apenas os estágios iniciais da DRC, que são: 1, 2 e 3A continuarão sendo representados na ontologia.

Após identificadas e especificadas em linguagem natural, as mudanças deverão ser organizadas em um relatório que terá como finalidade direcionar a implementação das mudanças quando estiver sendo realizada a edição da ontologia para refletir as mudanças requeridas. Na Tabela 4 pode ser observado um exemplo de relatório contendo as mudanças especificadas para implementação em uma ontologia sobre a DRC em evolução por mudança de foco. Conforme a tabela, as mudanças podem ser organizadas de acordo com as especificações das mudanças descritas acima.

Tabela 4. Exemplo de organização do relatório de mudanças a serem aplicadas em uma ontologia em evolução por mudança de foco

<b>Ontologia em Evolução:</b> ontoDecideDRC		
<b>Motivo da Evolução:</b> Mudança de foco da ontologia		
<b>Descrição do conceito</b>	<b>Mudança a ser aplicada</b>	<b>Descrição da Mudança</b>
Medico	Alteração	Substituir por “MedicoDaAtencao Primaria”
Conduta	Adição	Adicionar na hierarquia como conceito raiz
CondutaEstagio1	Adição	Adicionar na hierarquia como conceito filho do conceito “Conduta”
DoençaRenalCronicaEstagio4	Remoção	Remover conceito e todas as suas

Fonte: Autoria própria.

## II. Especificação das Mudanças de uma ontologia em evolução por mudanças nas diretrizes do domínio representado

Nesta perspectiva de evolução, a primeira tarefa a ser realizada para especificar as mudanças é a sua classificação. Tais mudanças podem ser classificadas como mudanças decorrentes de alterações nas diretrizes existentes e que conseqüentemente devem ser refletidas no conteúdo da ontologia em evolução; conceitos a serem adicionados na ontologia em virtude da inclusão de novos conceitos nas diretrizes por ela representadas e conceitos obsoletos que devem ser removidos da ontologia ou substituídos por novos conceitos. Desse modo, a evolução de uma ontologia por mudanças nas diretrizes relativas ao domínio nela tratado, ocasiona a necessidade de realização de um conjunto de mudanças que podem ser formalmente representado do seguinte modo:

$$C_{md} \equiv \{c_{al}, c_n, c_o\}$$

Onde,

- $C_{md}$  (mudança nas diretrizes) – Representa os conceitos das fontes de conhecimento do domínio representado na ontologia que devem ser considerados em seu processo evolutivo. Tais conceitos podem ser classificados em:
  - $c_{al}$  (alterados) – Representa os conceitos que serão alterados na ontologia em virtude de terem sofrido mudanças nas diretrizes do domínio representado. Utilizando como exemplo a ontologia sobre a DRC, considere que os valores limites da TFG utilizados para determinar em que estágio da doença o paciente se encontra, sofreu mudança. Tal mudança, naturalmente foi aplicada na diretriz utilizada para direcionar o cuidado de paciente portador dessa patologia. Desse modo, a ontologia que representa esse domínio também deverá ser alterada para refletir essa mudança e evitar divergências entre o conhecimento representado e sua fonte.
  - $C_n$  (novos) – Representa os novos conceitos incluídos nas fontes de conhecimento do domínio representado e que, em decorrência disso, devem ser adicionados à

ontologia. Por exemplo, em uma ontologia sobre a DRC, foi criado um novo estágio para classificação da doença. Esse novo estágio, bem como as formas de conduta do paciente que nele se encontram, foram incorporados à diretriz clínica para o cuidado de pacientes com DRC, como consequência disso, a ontologia que representa o domínio em questão também deverá ser atualizada de modo a refletir as adições de novos conceitos à diretriz.

- $C_o$  (obsoletos)– Representa os conceitos que serão removidos da ontologia por não serem mais utilizados na diretriz do domínio representado. Isso ocorre quando uma atualização na diretriz do domínio torna obsoleto determinado conceito. Para exemplificar, considere o fato de uma fórmula para cálculo da TFG do paciente com DRC ter entrado em desuso e por isso ter sido removida da diretriz que aborda os cuidados de paciente com a doença. Uma ontologia que representa esse domínio também deverá remover essa fórmula do seu conteúdo tendo em vista que a mesma não será mais utilizada e a manutenção desse conceito poderia levar a erros durante a utilização da ontologia pelos seus usuários finais.

Na Tabela 5 pode ser observado um exemplo de relatório contendo as mudanças especificadas para implementação em uma ontologia sobre a DRC em evolução por mudanças em suas diretrizes de domínio. Similarmente à estruturação do relatório com as mudanças a serem aplicadas em uma ontologia em evolução por mudança de foco, tais mudanças podem ser organizadas de acordo com as especificações das mudanças descritas nos tópicos acima os quais apresentam justificativas para o fato dos conceitos da ontologia necessitarem de mudanças.

Tabela 5. Exemplo de organização do relatório de mudanças a serem aplicadas em uma ontologia em evolução por mudanças em suas diretrizes de domínio

<b>Ontologia em Evolução:</b> ontoDecideDRC		
<b>Motivo da Evolução:</b> Mudanças nas diretrizes do domínio da ontologia		
<b>Descrição do conceito</b>	<b>Mudança a ser aplicada</b>	<b>Descrição da Mudança</b>
TFG1	Alteração	Alterar valor da propriedade de dado “TFG1” para $\geq 95$
DoencaRenalCronicaEstagio5co mTransplante	Adição	Adicionar na hierarquia como conceito filho do conceito “Estadiamento”

---

MDRDSimplificada	Remoção	Remover conceito e todas as suas referências da ontologia
------------------	---------	---

---

Fonte: Autoria Própria.

### **III. Especificação das Mudanças de uma ontologia em evolução por mudanças em sua terminologia de referência**

Esse motivo de evolução é ocasionado por mudanças na terminologia utilizada como referência para representação dos conceitos de uma ontologia ou pela necessidade de padronização total dos seus termos advinda da decisão do Engenheiro de ontologias de aderir ou utilizar um novo padrão que faça a substituição do padrão utilizado atualmente para tradução dos termos. Desse modo, a evolução da ontologia cujo motivo envolve a sua terminologia de referência possui duas perspectivas: evolução pra refletir alterações na terminologia utilizada atualmente e evolução para padronização total de uma ontologia que não possui terminologia de referência ou necessita realizar a substituição do padrão utilizado atualmente por meio da adoção de uma nova terminologia para representação conceitual.

SNOMED CT é uma terminologia clínica validada e internacionalmente utilizada para representar clinicamente informações relevantes de forma consistente, confiável e compreensiva (SNOMED CT, 2018). Tendo em vista que este trabalho é voltado para o domínio biomédico, todos os exemplos realizados nesta subseção farão referência à SNOMED CT como forma de tornar mais claro o processo de identificação de mudanças em terminologias de referência pertencentes a esse domínio. Nas subseções seguintes serão descritas as atividades a serem consideradas para a especificação das mudanças em um processo evolutivo ocasionado por alterações ocorridas na terminologia de referência da ontologia.

#### **i. Especificação das Mudanças de uma ontologia em evolução para refletir alterações ocorridas em sua terminologia de referência**

Neste caso a ontologia está evoluindo para refletir alterações na terminologia que já está sendo utilizada. Assim, a evolução por esse motivo se torna necessária porque um novo *release* da terminologia foi lançado e alterações referentes ao domínio representado na ontologia foram identificadas. Por exemplo, considere que a ontologia sobre a DRC citada nos exemplos anteriores utiliza SNOMED CT como terminologia padrão para representação dos

termos deste domínio e um novo *release* dessa terminologia foi lançado recentemente. Neste *release* o termo “*ChronicKidneyDisease*” teve sua aceitabilidade alterada. Caso a ontologia utilize a aceitabilidade como critério de seleção do termo que será incluído em seu conteúdo e a aceitabilidade do termo “*ChronicKidneyDisease*” tenha sido alterada de “*Preferred*” para “*Acceptable*”, a descrição do termo utilizada na ontologia deverá ser alterada de modo que o novo nome considerado preferível passe a ser utilizado.

Diante dessa perspectiva de evolução o processo recomendado para a realização da evolução da ontologia é o seguinte:

**a) Análise do novo *release* da terminologia**

Ao ser notificado sobre o lançamento da nova versão da terminologia de referência da ontologia, o engenheiro de ontologias deverá identificar quais mudanças referentes ao domínio representado foram realizadas. Em SNOMED CT os *releases* são disponibilizados na *SNOMED International Release Management*<sup>20</sup>. Neste mesmo endereço as versões também podem ser baixadas em Formato Portátil de Documento (PDF, do inglês, *Portable Document Format*). O último *release* da *SNOMED CT International* (versão de janeiro de 2019) foi lançada em 07 de dezembro de 2018 com acesso disponível apenas para membros.

Como descrito na subseção 2.5, em SNOMED CT os termos estão disponibilizados em *browsers* internacionais ou locais, que funcionam como interfaces gráficas de acesso aos termos. Os *releases* são disponibilizados de acordo com o *browser*. Por exemplo, se uma nova versão da *SNOMED CT International* for lançada a descrição do *release* será a seguinte: *July 2018 SNOMED CT International Edition*. A disponibilização dos *releases* facilita o processo de identificação das mudanças, pois neles há a descrição dos conceitos adicionados e daqueles que sofreram alteração de acordo com a área. Por exemplo, se foram adicionados conceitos sobre anatomia na terminologia, nas notas sobre o *release* estará a quantidade de conceitos adicionados bem como a descrição dessas inclusões, conforme pode ser observado na Tabela 6.

Essa representação auxilia o Engenheiro de ontologias no processo de identificação de alterações realizadas, conceitos adicionados e conceitos que foram inativados especificamente

<sup>20</sup><https://confluence.ihtsdotools.org/display/RMT/SNOMED+International+Release+Management+Home>

no domínio representado na ontologia em evolução. A Tabela 7 apresenta um resumo dos conceitos adicionados no último *release* da edição internacional da SNOMED CT. A Tabela 8 contém os números dos conceitos que foram melhorados e lançados nessa edição.

Tabela 6. Mudanças sobre anatomia implementadas no *release* de janeiro de 2019 da SNOMED CT edição Internacional

---

**Anatomy - New anatomy concepts: 442**

---

**Content quality improvements:**

---

- *Descriptions of 44 concepts updated to conform to the editorial guidance for the naming convention for laterality.*
  - *The attribute 123005000 |Part of (attribute)| has been inactivated and a new attribute |Proper part of| has been added in the January 2019 release. All part of relationships are inactivated and the different part of relationships will be implemented after the revision of hierarchical relationships for anatomy content.*
- 

Fonte: SNOMED International Release Management.

Conforme pode ser visto na Tabela 6, no último *release* da SNOMED CT *international*, foram adicionados 442 conceitos sobre anatomia. Na última linha da tabela estão as duas primeiras descrições de melhorias realizadas nos conceitos da terminologia e lançadas nesse *release*. Conforme no primeiro tópico dessa linha, 44 descrições de conceitos sobre anatomia foram alterados para estar em conformidade com a orientação editorial da convenção de nomenclatura para lateralidade. A segunda descrição de melhoramento diz respeito à inativação do atributo “*Part of*” e ativação do atributo que o substituirá: “*Proper part of*”.

Tabela 7. Resumo dos conceitos adicionados no *release* de janeiro de 2019 da SNOMED CT edição internacional.

<b>Estatísticas SNOMED CT</b>	<b>Número de conceitos adicionados</b>
<i>SNOMED CT Concept (SNOMED RT+CTV3)</i>	11.903
<i>Pharmaceutical / biologic product (product)</i>	7.986
<i>Clinical finding (finding)</i>	2.114
<i>Body structure (body structure)</i>	442
<i>Procedure (procedure)</i>	393
<i>Organism (organism)</i>	326
<i>Substance (substance)</i>	223
<i>Qualifier value (qualifier value)</i>	135
<i>Record artifact (record artifact)</i>	61
<i>Physical object (physical object)</i>	59

<i>Observable entity (observable entity)</i>	56
<i>Situation with explicit context (situation)</i>	50
<i>Event (event)</i>	27
<i>SNOMED CT Model Component (metadata)</i>	16
<i>Staging and scales (staging scale)</i>	11
<i>Specimen (specimen)</i>	2
<i>Social context (social concept)</i>	1
<i>Special concept (special concept)</i>	1

Fonte: SNOMED *International Release Management*.

Tabela 8. Resumo dos conceitos modificados no *release* de janeiro de 2019 da SNOMED CT edição internacional.

<b>Estatísticas de melhoria da SNOMED CT para conceitos existentes</b>	<b>Número</b>
<i>Total number of changes</i>	72.376
<i>Change to stated concept definition</i>	20.294
<i>Change to inferred concept definition</i>	57.325
<i>Change in concept definition status from Primitive to Sufficiently Defined</i>	1.269
<i>Description change</i>	20.288
<i>Inactivated concepts</i>	3.035
<i>Inactivated descriptions</i>	12.663
<i>Reactivated concepts</i>	21

Fonte: SNOMED *International Release Management*.

## **b) Especificação das mudanças realizadas na terminologia**

Após a identificação das mudanças no *release* da terminologia, os conceitos que passaram por melhoramento, foram incluídos ou inativados devem ser especificados em linguagem natural em um relatório onde estarão todas as mudanças a serem realizadas na ontologia com base no que foi alterado em sua terminologia de referência. Assim, as mudanças a serem implementadas em uma ontologia em processo de evolução devido a alterações em sua terminologia de referência podem ser formalmente representadas do seguinte modo:

$$C_{mt} \equiv \{c_{me}, c_{inc}, c_{ina}\}$$

Onde,

- $C_{mt}$  (mudanças na terminologia) – Representa os conceitos que sofreram mudanças na terminologia de referência da ontologia. Tais conceitos podem ser classificados em:
  - $C_{me}$  (melhorados) – Representa os conceitos que passaram por mudanças com o intuito de serem melhorados. Por exemplo, no *release* de janeiro de 2019 da edição internacional da terminologia SNOMED CT, foram realizadas 2.000 mudanças em relacionamentos hierárquicos sobre anatomia e modelos de conceitos referentes a doenças e procedimentos. Caso uma ontologia sobre anatomia utilize SNOMED CT como terminologia para representação conceitual, o engenheiro de ontologias deverá observar tais mudanças de modo a identificar quais delas são referentes aos termos representados na ontologia para que assim, possa ser realizado o trabalho de alteração com o intuito de refletir as mudanças da terminologia na ontologia referente ao domínio em estudo.
  - $C_{inc}$  (incluídos) – Representa os conceitos que foram incluídos na terminologia de referência da ontologia e deverão ser adicionados na mesma de modo a garantir a continuação da concordância entre a terminologia e a ontologia que a utiliza para representação de conceitos. Por exemplo, no último *release* da SNOMED CT, utilizado no exemplo do subtópico anterior e conforme a Tabela 7 foram adicionados 2.114 novos conceitos classificados como “*Clinical finding*”. *Clinical finding* ou Achado Clínico em português, é caracterizado pela SNOMED CT como uma observação normal ou anormal, julgamentos ou avaliações de pacientes realizadas pelo médico. Desse modo, qualquer ontologia que utiliza SNOMED CT como terminologia de referência deverá realizar uma análise dos conceitos que foram adicionados e classificados como Achados Clínicos pois o mesmo pode estar relacionado com qualquer domínio.
  - $C_{ina}$  (inativados) – Representa os conceitos que foram inativados na terminologia de referência da ontologia. Utilizando o exemplo dos tópicos anteriores sobre o último *release* da SNOMED CT edição internacional e conforme pode ser observado na Tabela 8, 3.035 conceitos foram inativados. Considerando que a



ontologia sobre a DRC, citada anteriormente, utiliza SNOMED CT como terminologia de referência e que dentre esses conceitos inativados estão conceitos relativos à DRC, especificamente relativos ao conteúdo representado nessa ontologia, tais conceitos deverão ser substituídos de modo a manter-se a conformidade dos conceitos dessa ontologia com os termos da sua terminologia de referência. Assim, os termos que passaram a ser utilizados no lugar dos que foram inativados na terminologia deverão ser adicionados na ontologia como substitutos daqueles que entraram em desuso.

Após especificadas, as mudanças ocorridas na terminologia poderão ser visualizadas mais claramente na ontologia que a utiliza como referência. Por isso, recomenda-se que seja produzido um relatório de mudanças a serem aplicadas na ontologia a partir das mudanças identificadas na terminologia. Nesse relatório deverá conter, além do nome atual, o nome anterior do termo de modo a facilitar o processo de detecção do conceito que deverá ser alterado na ontologia e o motivo da mudança, que poderá ser descrito considerando as mudanças conceituais descritas nos tópicos acima, ou seja, as mudanças poderão ser motivadas pela inclusão de novos conceitos, alteração ou inativação de conceitos existentes.

É importante que o SNOMED CT *identifier* ou SCTID do termo também seja considerado na elaboração do relatório, pois os SCTIDs dos termos também sofrem mudanças quando o termo é alterado na terminologia. Um exemplo de organização do relatório de mudanças a serem aplicadas na ontologia em evolução para refletir alterações ocorridas em sua terminologia de referência pode ser observado na Tabela 9.

Tabela 9. Exemplo de organização do relatório de mudanças a serem aplicadas em uma ontologia em evolução por alterações em sua terminologia de referência

<b>Ontologia em Evolução:</b> ontoSkin				
<b>Motivo da Evolução:</b> Alterações na terminologia de referência (SNOMED CT)				
<b>SCTIDs</b>		<b>Descrições</b>		<b>Motivo da Mudança</b>
<b>Anterior</b>	<b>Atual</b>	<b>Anterior</b>	<b>Atual</b>	
112625008	1806006	<i>Cutaneous eruption (morphologic abnormality)</i>	<i>Eruption (morphologic abnormality)</i>	Inativação de conceito
371598009	371598009	<i>Heberden's node (finding)</i>	<i>Heberden's node (disorder)</i>	Melhoria de conceito

Fonte: Autoria própria. As descrições e IDs pertencem à SNOMED CT e foram capturados no endereço eletrônico da SNOMED *International Release Management*

## **ii. Especificação das Mudanças de uma ontologia em evolução para padronização terminológica total**

A padronização terminológica total da ontologia pode ser motivada pela necessidade de aderir a uma terminologia para representação conceitual ou substituir a terminologia utilizada atualmente pela ontologia. O primeiro passo a ser considerado para especificação das mudanças de uma ontologia em evolução para padronização total dos seus conceitos é a escolha da terminologia de referência. Antes de definir a terminologia que será utilizada como referência para representação dos conceitos da ontologia é preciso saber se a terminologia abarca o todos os conceitos do conteúdo representado na ontologia. Isso é importante pois evita-se que se identifique a não existência de conceitos da ontologia na terminologia quando o processo de padronização já estiver avançado. Caso a terminologia utilizada seja SNOMED CT termos não existentes poderão ser adicionados à terminologia por meio da criação de expressões pós-coordenadas (seção 2.5).

Caso a terminologia escolhida seja internacional e nela não exista representação de termos em português, os conceitos a serem padronizados deverão ser traduzidos para o inglês para que o processo de busca dos termos seja facilitado. Tendo em vista que o *framework* definido nesta dissertação é voltado para o apoio à evolução de ontologias do domínio biomédico, em todos os exemplos serão considerados termos pertencentes a esse domínio, bem como termos constantes na terminologia SNOMED CT.

Após a escolha da terminologia a ser utilizada, deverá ser realizado o mapeamento dos termos da ontologia. Nesse processo, os conceitos presentes na ontologia deverão ser buscados na terminologia. Geralmente, as terminologias do domínio biomédico possuem um repositório de acesso gratuito onde os termos podem ser buscados utilizado-se palavras-chave, o que facilita o processo de identificação dos mesmos. São exemplos de padrões de classificação de termos do domínio biomédico que possuem repositórios para disponibilização desses termos, a ICD-10<sup>21</sup>, ICD-11<sup>22</sup>, GO<sup>23</sup> e SNOMED CT<sup>24</sup>. Na Figura 16 pode ser

21<https://icd.who.int/browse10/2010/en>

22<https://icd.who.int/browse11/l-m/en>

23[http://amigo.geneontology.org/amigo/dd\\_browse](http://amigo.geneontology.org/amigo/dd_browse)

24<https://browser.ihtsdotools.org/>

observado o resultado dado pela busca do termo “*chronic kidney disease*” no repositório da terminologia SNOMED CT edição internacional.

Além de terminologia, SNOMED CT é uma ontologia formalmente definida e por isso todos os seus termos estão organizados em categorias que refletem sua estrutura no mundo. Desse modo, quando um conceito é buscado em SNOMED CT o resultado é o conceito juntamente com seus conceitos pais e filhos, conforme pode ser observado na Figura 16, onde o termo selecionado, “*chronic kidney disease*”, pode ser visto na coluna da direita entre os seus conceitos pais (*Parents*) e filhos (*Children*). Isso facilita o processo de mapeamento por meio da identificação rápida dos conceitos e ajuda na estruturação correta do conteúdo ontológico.

Figura 16. Retorno do *browser* SNOMED CT para a busca do termo "*chronic kidney disease*"

The screenshot displays the SNOMED CT International Browser interface. At the top, it shows the release version 'International Edition 20180731' and the perspective 'Full'. The search bar contains the term 'chronic kidney disease', and the results show 191 matches found in 0.522 seconds. The search results are organized into a table with columns for the concept name and its description. The interface also includes navigation tabs for 'Concept Details', 'Expression Constraint Queries', and 'Summary'. The 'Concept Details' section is expanded, showing 'Parents' (Chronic disease of genitourinary system (disorder) and Renal impairment (disorder)) and 'Children' (15 concepts, including Chronic kidney disease due to hypertension (disorder), Chronic kidney disease due to systemic infection (disorder), etc.).

Fonte: *Browser* SNOMED CT

Para as ontologias que utilizam a terminologia de outras ontologias como referência, existe um *plug-in*, o BioPortal *Import plug-in*<sup>25</sup>, disponível para a ferramenta Protégé, que permite que os usuários importem classes de ontologias que estejam disponíveis no BioPortal<sup>26</sup>, repositório para armazenamento e gerenciamento de ontologias.

<sup>25</sup>[https://protegewiki.stanford.edu/wiki/BioPortal\\_Import\\_plug-in](https://protegewiki.stanford.edu/wiki/BioPortal_Import_plug-in)

<sup>26</sup><https://bioportal.bioontology.org/>

A *Snow OWL Meaningful Query (MQ)* <sup>27</sup> é uma plataforma semântica *online* para processamento de dados em saúde. A *Snow OWL MQ* é uma ferramenta gratuita e possui, dentre outras funcionalidades, o acesso a conceitos de mais de 70 terminologias incluindo SNOMED CT, *Logical Observation Identifiers Names and Codes (LOINC)*, ICD-10 e ICD-9. O *Snow OWL* também pode ser usado como um *plug-in*<sup>28</sup> disponível para a versão 4.1 do editor de ontologias Protegé por meio do qual podem ser acessados os termos da SNOMED CT. O Engenheiro de ontologias responsável pelo processo evolutivo poderá decidir se o mapeamento de termos será manual, ou seja, se o mapeamento será feito por meio de buscas em repositórios de termos ou de ontologias; ou se alguma das ferramentas disponíveis ou mesmo um conjunto de ferramentas oferece o suporte adequado para essa tarefa.

Ao escolher o termo, além da sua similaridade com o conceito da ontologia a classificação do mesmo também deve ser levada em conta. Por exemplo, em SNOMED CT os termos representados podem ser classificados em 19 hierarquias de domínio específico, dentre os quais estão *Disorder/Clinical Finding*, *Procedure*, *Substance*, *Organism* etc. O domínio *Disorder/Clinical Finding* apesar de ser representado como um só, é usado separadamente para classificar termos como *Disorder* (Doença) ou *Clinical Finding* (Achado Clínico). *Disorder* é definido como um estado clínico sempre anormal enquanto *Clinical Finding* classifica um termo como uma observação normal ou anormal. Assim, com o intuito de evitar o uso de termos com a classificação incorreta, ao buscar um termo é importante estar atento à classificação do mesmo.

Conforme na seção 2.5, em SNOMED CT os termos também podem ser classificados pela sua aceitabilidade. Assim, caso essa seja a terminologia escolhida, recomenda-se que termos classificados como preferíveis sejam os escolhidos para a representação dos conceitos da ontologia. Portanto, após a escolha do termo observando o seu domínio, para as ontologias que utilizam SNOMED CT como terminologia de referência, recomenda-se que o termo escolhido para ser importado para a ontologia seja o FSN ou o sinônimo com aceitabilidade “*Preferred*”. Além da descrição do termo é importante que o seu identificador único também seja capturado da terminologia. Isso evita que termos com descrições iguais sejam confundidos durante o seu processamento. Como discutido anteriormente nesta dissertação,

<sup>27</sup><https://mq.b2i.sg/snow-owl/#!/terminology>

<sup>28</sup>[https://protegewiki.stanford.edu/wiki/Snow\\_Owl](https://protegewiki.stanford.edu/wiki/Snow_Owl)

em SNOMED CT os termos são unicamente identificados por meio do SCTID.

Na Figura 17, pode ser observado o retorno do repositório SNOMED CT para o termo “*cancer*”. Conforme na figura, diversos resultados foram exibidos, desse modo, caberá ao usuário responsável pelo mapeamento dos termos escolher aquele que possui a classificação similar à do conceito na ontologia. Nesse caso, se o conceito buscado se referir à doença, o termo escolhido deverá ser o “*Malignance neoplastic disease (disorder)*”. No *browser* SNOMED CT, as hierarquias de domínio específico podem ser usadas para filtrar as buscas, assim, caso já se saiba a classificação do termo buscado, pode-se usar esse filtro para facilitar a identificação do mesmo no *browser*.

Figura 17. Retorno do *browser* SNOMED CT para a busca do termo "cancer"

The screenshot displays the SNOMED CT International Browser interface. At the top, it shows the release version 'en-edition v20180731' and the perspective 'Full'. The search bar contains the term 'cancer', and the results list shows 1382 matches found in 0.368 seconds. The results are filtered by language (English) and semantic tag (disorder). The top result is 'Malignant neoplasm, primary (morphologic abnormality)' with SCTID 363348000. The right-hand pane shows the 'Concept Details' for 'Malignant neoplastic disease (disorder)', including its parents, children, and associated morphology.

Options	Type at least 3 characters ✓ Example: shou fra
Search Mode: Partial matching search mode	Search: cancer
Status: Active concepts only	1382 matches found in 0.368 seconds.
Group by concept	Cancer Malignant neoplasm, primary (morphologic abnormality)
Filter results by Language	Cancer Malignant neoplastic disease (disorder)
Filter results by Semantic Tag	CA - Cancer Malignant neoplastic disease (disorder)
disorder 550	Lung cancer Primary malignant neoplasm of lung (disorder)
finding 272	dMMR cancer Microsatellite instability-high solid malignant tumor (disorder)
situation 200	H/O: cancer History of malignant neoplasm (situation)
record artifact 152	Skin cancer Malignant neoplasm of skin (disorder)
procedure 66	Cecal cancer Malignant tumor of cecum (disorder)

Fonte: *Browser* SNOMED CT

Durante o processo de mapeamento dos conceitos da ontologia utilizando a terminologia escolhida como o novo padrão para representação conceitual da mesma, um relatório contendo os termos mapeados deverá ser produzido. O relatório servirá de guia para o processo de implementação das mudanças na ontologia. Na Tabela 10 pode ser observado um exemplo de relatório contendo os termos que passarão a representar os conceitos de uma ontologia sobre a DRC em processo evolutivo e que utiliza como terminologia de referência o padrão SNOMED CT.

Tabela 10. Exemplo de organização do relatório de mudanças a serem aplicadas em uma ontologia em evolução para padronização terminológica total

<b>Ontologia em Evolução:</b> ontoDecideDRC		
<b>Motivo da Evolução:</b> Padronização terminológica total com SNOMED CT		
<b>Descrições</b>		<b>SCTID</b>
<b>Anterior</b>	<b>Atual</b>	
DoencaRenalCronica	<i>ChronicKidneyDisease(disorder)</i>	709044004
DoencaRenalCronicaEstagio1	<i>ChronicKidneyDiseaseStage1(disorder)</i>	431855005
Nefrologista	<i>Nephrologist(occupation)</i>	11911009

Fonte: Autoria própria. As descrições e IDs pertencem à SNOMED CT e foram capturados no *browser* SNOMED *International edition*

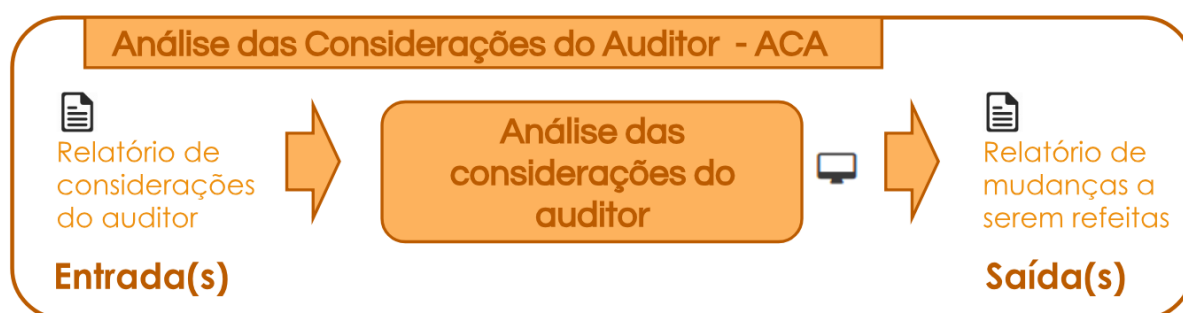
Após o mapeamento dos termos bem como sua organização no relatório de alterações a serem aplicadas, o processo de especificação das mudanças da ontologia em evolução para padronização terminológica total poderá ser concluído. O relatório de saída das atividades de cada subprocesso tem como objetivo facilitar o entendimento das mesmas pelo engenheiro de ontologias e demais responsáveis pela evolução da ontologia e assim subsidiar a execução das etapas posteriores, desse modo, as mudanças devem ser representadas em linguagem natural.

O relatório de mudanças a serem aplicadas, fornecido como saída da atividade de Representação das Mudanças pode ser comparado a um documento de requisitos para o desenvolvimento de um *software*, tendo em vista que conforme Sommerville (2013), um documento de requisitos deve descrever em detalhes os requisitos do sistema sobre o qual o mesmo foi escrito e no relatório de mudanças a serem feitas devem estar descritas todas as alterações requeridas pelos usuários da ontologia em evolução além das mudanças necessárias em decorrência do motivo da evolução da ontologia. A etapa posterior ao PE, que se chama Implementação de Mudanças, é totalmente dependente do relatório de mudanças pois apenas as mudanças que foram representadas deverão ser efetivamente implementadas na ontologia.

Caso a ontologia tenha passado por processo de auditoria onde foi detectada a ausência de mudanças que deveriam ter sido aplicadas ou a realização de mudanças indesejadas é necessário que a mesma retorne à fase inicial do *framework*. Devido a isso, a segunda ação que pode ser executada no subprocesso PE é a Análise das Considerações do Auditor (ACA) (Figura 18). Essa atividade possui como entrada um relatório onde estão descritas as considerações do auditor responsável pela verificação da similaridade entre o

plano de evolução e o resultado final da evolução. Tal resultado deverá ser a ontologia com as mudanças requeridas aplicadas bem como as inconsistências resolvidas. Por meio do relatório com as considerações do auditor é possível que se identifique os erros cometidos durante a evolução. Após a identificação dos erros, um novo plano de evolução é traçado, agora com o intuito de corrigir as falhas ocorridas no processo anterior.

Figura 18. Análise das Considerações do Auditor



Fonte: Autoria própria

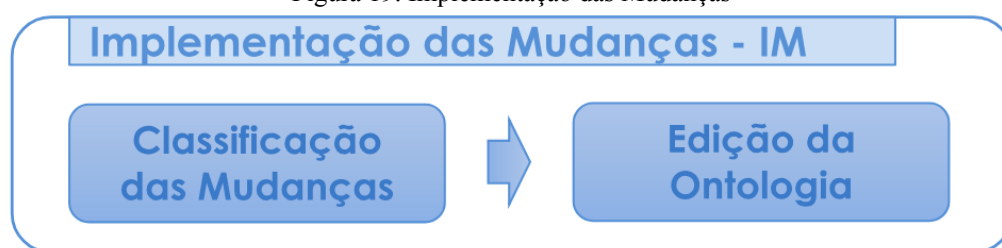
Os relatórios produzidos como saídas após a execução das atividades executadas no processo normal de evolução podem ser úteis para a aplicação das correções. Assim, não será consumido tanto tempo para corrigir as falhas, pois no relatório de mudanças aplicadas, por exemplo, devem conter as estratégias (como determinada mudança foi aplicada) usadas que podem servir como base para aplicar novas mudanças. O mesmo ocorre com o relatório fornecido pela fase de tratamento de inconsistências que contém as estratégias aplicadas para resolver as inconsistências que geralmente ocorrem após a aplicação de mudanças. É importante que o tempo esperado para a finalização da evolução da ontologia e disponibilização da mesma considere a possibilidade de ocorrência de falhas e consequente necessidade de aplicação de correções de modo a se evitar consequências inesperadas que prejudiquem a disponibilidade da ontologia. Após a execução do subprocesso PE a evolução segue normalmente para a realização do subprocesso Implementação de Mudanças explicado com detalhes na próxima subseção.

### 5.2.2 Implementação das mudanças - IM

A Implementação das Mudanças (Figura 19) é a segunda fase do *framework*. A execução do subprocesso PE é pré-requisito imprescindível para a implementação das

mudanças requeridas pelos usuários da ontologia considerando que tais mudanças foram identificadas durante a execução desse subprocesso. Nesta fase, as mudanças que foram descritas no relatório de mudanças a serem realizadas, fornecido como saída do subprocesso PE são inicialmente classificadas em uma tarefa denominada Classificação das Mudanças e depois, a ontologia deverá ser editada para refletir as mudanças requeridas e especificadas durante a fase de Planejamento da Evolução em uma tarefa denominada Edição da ontologia. Nas próximas subseções essas suas tarefas são detalhadamente descritas.

Figura 19. Implementação das Mudanças



Fonte: Autoria própria

#### 5.2.2.1 Classificação das Mudanças – CM

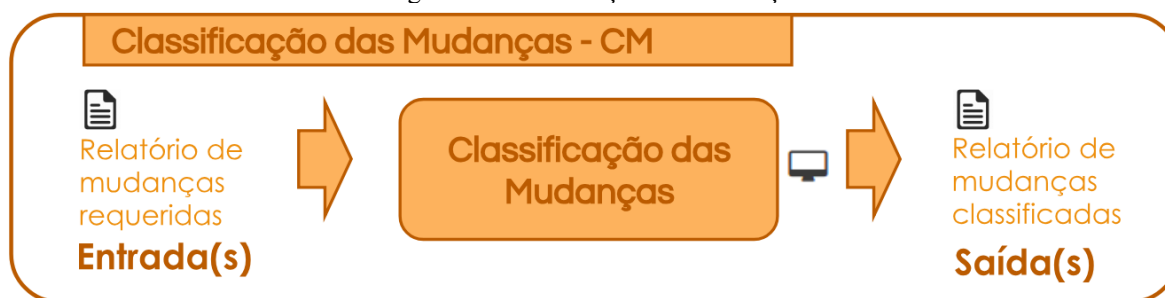
Nesta tarefa (Figura 20), as mudanças especificadas na fase de Planejamento da Evolução deverão ser classificadas como simples ou complexas com base no trabalho realizado por Noy e Klein (2003). Caso o motivo da evolução esteja relacionado à terminologia de referência da ontologia, não será necessária a classificação das mudanças. Em contrapartida, mudanças decorrentes de evoluções motivadas por mudanças nas diretrizes do domínio ou pela mudança do foco da ontologia, deverão ser classificadas. A classificação de mudanças poderá ser feita de maneira similar para qualquer um desses dois motivos.

As mudanças deverão ser classificadas como simples ou complexas. Segundo Noy e Klein (2006), essa classificação leva em consideração a quantidade de ações que determinada mudança necessita para alcançar seu objetivo final, ou seja, caso seja necessária a execução de mais de uma ação, a mudança é classificada como complexa, nos casos em que apenas uma ação é suficiente para alcançar o objetivo da mudança, classifica-se a mesma como simples. Por exemplo, considerando a necessidade de remoção de um termo de uma ontologia em evolução porque o mesmo se tornou obsoleto na nova versão da diretriz utilizada como fonte de conhecimento dessa ontologia, a ação de remover o termo pode ser classificada como uma



operação simples, tendo em vista que apenas uma ação é executada. Caso se tivesse optado por mover uma classe dessa ontologia, a operação seria classificada como complexa tendo em vista que em uma operação de mover, um conjunto composto por duas operações simples é executado: remoção e adição.

Figura 20. Classificação das Mudanças



Fonte: Autoria própria

Após a classificação, um relatório com as mudanças classificadas deve ser fornecido. Esse relatório deverá ser construído a partir do relatório de mudanças especificadas na fase de Planejamento das Mudanças. Desse modo, a classificação se tornará um processo menos oneroso tendo em vista que a descrição das mudanças poderá auxiliar na tarefa de classificação das mesmas como simples ou complexas. Na Tabela 11 pode ser observado um exemplo de relatório de mudanças classificadas a serem aplicadas em uma ontologia sobre a DRC em processo de evolução. A Tabela 11 foi criada a partir da Tabela 6, usada como exemplo de relatório de mudanças especificadas para uma ontologia em evolução por mudanças em suas diretrizes de domínio.

Tabela 11. Exemplo de relatório de mudanças classificadas para serem implementadas em uma ontologia em evolução por mudanças em suas diretrizes de domínio

<b>Ontologia em Evolução:</b> ontoDecideDRC			
<b>Motivo da Evolução:</b> Mudanças nas diretrizes do domínio da ontologia			
<b>Descrição do conceito</b>	<b>Mudança a ser aplicada</b>	<b>Descrição da Mudança</b>	<b>Tipo de Mudança</b>
TFG1	Alteração	Alterar valor da propriedade de dado “TFG1” para $\geq 95$	Complexa
DoençaRenalCronica Estágio5ComTransplante	Adição	Adicionar na hierarquia como conceito filho do conceito “Estadiamento”	Simple
MDRDSimplificada	Remoção	Remover conceito e todas as	Simple

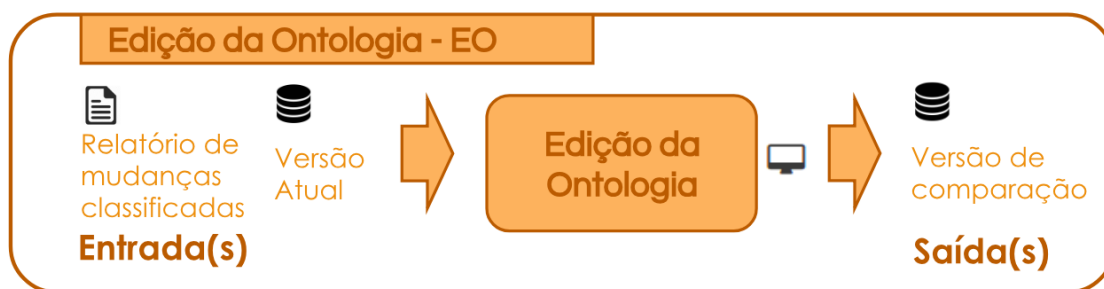
Fonte: Autoria própria.

A classificação das mudanças permite que se tenha uma visão mais clara do nível de complexidade dessas mudanças. Isso permite que o Engenheiro de ontologias possa decidir sobre quais mudanças devem ser priorizadas, por exemplo. Assim, o relatório de mudanças classificadas tem como finalidade direcionar a tarefa onde as mudanças serão implementadas na ontologia após terem sido representadas e classificadas, a edição da ontologia.

### 5.2.2.2 Edição da ontologia - EO

Para essa tarefa, a ontologia em evolução e o relatório de mudanças especificadas e classificadas deverão ser fornecidos como entrada para as realização da atividade de edição da ontologia (Figura 21). Para a realização dessa atividade, recomenda-se fortemente o uso de um *software* de edição de ontologias. Todos os exemplos dados nesta seção farão referência ao editor de ontologias Protégé, tendo em vista que o mesmo possui todos os recursos necessários para a edição de ontologias em OWL, linguagem que foi utilizada para a construção da ontologia que será referenciada nesses exemplos. É importante que a atividade de edição siga o relatório de mudanças classificadas, pois em caso de problemas relacionados à aplicação de mudanças desconhecidas, encontrar o local exato do erro para que o mesmo seja reportado e resolvido pode se tornar um processo oneroso.

Figura 21. Edição da ontologia



Fonte: Autoria Própria

O motivo da evolução impacta diretamente nas tarefas a serem realizadas no processo de edição da ontologia. Assim, conforme na tarefa de Especificação das Mudanças, a edição da Ontologia deverá ser direcionada pelo motivo da evolução da ontologia. Considerando que evoluções ocasionadas por mudança de foco ou mudanças nas diretrizes de domínio da

ontologia possuem mudanças conceituais semelhantes, ou seja, mudanças que têm como base operações de adição, alteração e remoção, a edição da ontologia para a implementação de mudanças em processos evolutivos ocasionados por esses motivos, será o mesmo. A evolução motivada por mudanças na terminologia de referência da ontologia deverá seguir atividades distintas e específicas para o apoio à realização de alterações na terminologia utilizada na ontologia ou padronização terminológica total da ontologia em evolução. Adiante as tarefas de edição de acordo com o motivo da evolução são detalhadamente abordadas.

### **I. Edição da ontologia em evolução por mudança de foco ou mudanças em suas diretrizes de domínio**

Levando em conta a evolução ocasionada por mudança de foco ou mudanças nas diretrizes de domínio da ontologia, poderão ser realizados 3 tipos principais de operações de mudanças: adição, alteração e remoção de conceito. As mudanças relacionadas à remoção de conceitos deverão ser aplicadas antes das adições e alterações. Isso é importante pois evita-se o manuseio de conceitos que não serão mais utilizados e que podem prejudicar o processo de edição ao dificultar a visualização da ontologia para aplicação das demais mudanças.

#### **a) Deleção de conceito**

A deleção de um conceito pode ser classificada como deleção de conceito ou deleção de subconceito. O Engenheiro de ontologias deverá decidir se os subconceitos relacionados ao conceito a ser deletado deverão ser mantidos ou removidos junto com o seu conceito pai. A decisão de manter esses subconceitos deverá acompanhar a definição do novo local na hierarquia de classes da ontologia para onde tais subconceitos deverão ser encaminhados. Dessa forma, a decisão de manter esses subconceitos, envolve adicionar um novo conceito no topo da hierarquia desses conceitos, ou movê-los para serem subconceitos de um conceito existente. No primeiro caso, deverá ser criada uma nova classe na ontologia que possa ser usada como conceito pai dos subconceitos que pertenciam ao conceito deletado, configurando-se assim a realização de uma operação simples. No segundo caso, os subconceitos deverão sofrer mudanças complexas tendo em vista que os mesmos deverão ser movidos do seu local na hierarquia de conceitos. As operações de adição e de mover conceitos serão abordadas nas próximas subseções.

Caso se decida deletar os subconceitos, a deleção do conceito principal deverá ser precedida pela deleção dos seus subconceitos. O processo de verificação da existência de subconceitos associados ao conceito sobre o qual deverão ser realizadas operações está representado no Algoritmo 1.

---

Algoritmo 1. temSubconceito( $c, O$ )

---

```
for all  $e \in O$  do  
  if  $\exists e \in c$  then  
    true  
  else  
    false  
  end if  
end for
```

---

Fonte: Autoria própria

Se na ontologia existir uma entidade pertencente ao conceito que deve ser deletado, ou seja, se abaixo da hierarquia desse conceito existir um mais subconceitos, o retorno do algoritmo será *true*, caso contrário, o retorno será *false*. A deleção de conceitos que possuem subconceitos é um processo mais complexo quando comparado ao processo de deleção de conceitos sem subconceitos relacionados. Detectada a existência de subconceitos, o processo de deleção será recursivo, pois enquanto houverem subconceitos os mesmos deverão ser deletados antes da deleção do conceito principal. A deleção prévia dos subconceitos é importante pois a deleção de um conceito sem a remoção dos subconceitos a ele relacionados causa a dispersão desses subconceitos.

Em OWL os subconceitos sem um conceito pai tornam-se filhos da classe *thing*, classe padrão das ontologias desenvolvidas por meio do uso dessa Linguagem, causando assim, problemas de corretude na categorização dos conceitos da ontologia em evolução. A deleção de subconceitos pode ser representada por meio do Algoritmo 2. O Algoritmo 3 representa o processo de deleção de conceitos. Nele, caso existam subconceitos relacionados ao conceito principal sobre o qual está sendo aplicada a deleção, os mesmos serão removidos enquanto existirem. Caso contrário, ocorrerá a deleção simples do conceito em questão.

---

**Algoritmo 2.** deletarSubconceito( $c$ )

---

```
for all  $subc \in c$  do  
    deletarSubconceito(subc)  
end for
```

---

Fonte: Autoria própria

---

**Algoritmo 3.** deletarConceito( $c, O$ )

---

```
for all  $e \in O$  do  
    if temSubconceito( $c, O$ ) then  
        deletarSubconceito(subc)  
    else  
        deletarConceito( $c$ )  
    end if  
end for
```

---

Fonte: Autoria própria

---

### **b) Adição de conceito**

Em OWL, os conceitos a serem adicionados na ontologia em evolução podem ser convertidos em classes, instâncias ou propriedades. A criação desses componentes na ontologia deverá se dar exatamente nessa ordem, tendo em vista que indivíduos estão vinculados a classes e as propriedades determinam os relacionamentos entre os indivíduos. Se o conceito a ser adicionado for traduzido como classe, antes de se iniciar a tarefa de adição é importante a classificação desses conceitos como superconceitos ou subconceitos.

Essa classificação ajudará a definir se o novo conceito será uma classe ou subclasse na ontologia, o que facilitará a identificação do local onde a mesma será adicionada. Por exemplo, em uma ontologia sobre a DRC é necessária a adição de um novo conceito referente à criação de um novo estágio da doença. Nessa ontologia, os estágios da DRC estão organizados como subconceitos de uma classe chamada “Estadiamento”. Assim, o novo estágio a ser criado na ontologia deverá ser classificado como subconceito. Por outro lado, um conceito a partir do qual se possa organizar hierarquicamente as causas da DRC pode ser classificado como um superconceito tendo em vista que as causas deverão ser elencadas como subconceitos do conceito recém adicionado. Desse modo, na ontologia, esse conceito será

uma superclasse. O processo de adição de um conceito pode ser representado por meio do Algoritmo 4. Conforme o Algoritmo, se o conceito a ser adicionado for um subconceito, este será incluído na ontologia nessa condição, sendo o mesmo traduzido como uma subclasse. Caso a adição não seja de um subconceito, verifica-se se o conceito possui subconceitos a ele vinculados. Se possuir, o conceito principal será adicionado e os seus subconceitos deverão ser inseridos depois. Caso o conceito a ser adicionado não seja um subconceito ou não possua-os, o mesmo apenas deverá ser adicionado no local apropriado.

---

**Algoritmo 4.** adicionarConceito( $c, O, local$ )

---

```
if eSubconceito( $c$ ) then
    adicionarSubconceito( $c, O, local$ )
else if temSubconceito( $c, O$ ) then
    addConceito( $c, O, local$ )
    for all  $subc \in c$  do
        adicionarSubconceito( $subc, O, local$ )
    end for
else
    addConceito( $c, O, local$ )
end if
```

---

Fonte: Autoria própria

Após a criação das classes, poderá ser dado início ao processo de criação de instâncias. Antes de se iniciar essa tarefa, deverão ser identificadas as classes às quais pertencerão as instâncias que devem ser criadas. Portanto, uma análise prévia com o intuito de se identificar os conceitos aos quais as instâncias a serem criadas estão vinculadas, deverá ser realizada. Após identificar o domínio da instância a ser criada, o processo de edição poderá ser realizado. Como exemplo, no domínio da DRC, a TFG de um paciente pode ser classificada em diversos níveis, essa classificação ajuda a determinar o estágio da doença no qual o paciente se encontra. Tendo em vista que TFG é uma classe pertencente a uma ontologia sobre a DRC, os seus tipos podem ser convertidos em instâncias no contexto do modelo ontológico dessa ontologia. Assim, a classe TFG terá os possíveis níveis nos quais a mesma pode ser classificada, definidos como instâncias na ontologia sobre a DRC.

Os relacionamentos entre os conceitos convertidos em classes ou instâncias serão criados na ontologia como propriedades. Uma análise detalhada para que se possa identificar o tipo de relacionamento entre esses conceitos deverá ser realizada. Uma propriedade poderá ser usada para relacionar diversos conceitos. Tendo em vista que para classificar a DRC é necessário que se saiba o nível da TFG, o relacionamento entre o conceito que representa o estágio 1 da DRC e a TFG nível 1 poderá ser convertido em uma propriedade chamada “*eObtidoPor*”. Assim, na ontologia, uma regra contendo essa relação será representada da seguinte forma: DRCEstagio1 *eObtidoPor* TFGNivel1. Essa propriedade poderá ser usada para relacionar os demais estágios da DRC e os níveis da TFG. Após serem identificados os tipos de relacionamentos, todas as propriedades a serem criadas deverão fazer parte do relatório de mudanças a serem aplicadas.

### **c) Operação de mover conceito**

Mover conceitos é uma operação complexa do processo de edição da ontologia em evolução porque exige a realização de um conjunto de operações para se alcançar a finalidade da operação. A primeira atividade a ser realizada para a execução de uma operação de mover é a identificação dos conceitos sobre os quais a operação será realizada. A entrada da tarefa de edição da ontologia é um relatório de mudanças especificadas e classificadas. Assim, as operações a serem aplicadas sobre os conceitos já serão conhecidas. Portanto, a identificação dos conceitos a serem movidos será uma tarefa menos onerosa.

Após a identificação, deverá ser feita a análise dos conceitos a serem movidos com o intuito de identificar os que possuem subconceitos. Essa tarefa é importante pois evita que o processo de mover conceitos prejudique a organização hierárquica da ontologia. Desse modo, todos os subconceitos deverão ser movidos junto com seu conceito pai. A próxima tarefa a ser realizada é a definição do local de destino do conceito. É importante que isso seja previamente definido de modo que se evite o gasto de tempo extra com a localização do destino do conceito. O processo de mover conceitos pode ser representado por meio do Algoritmo 5.

---

**Algoritmo 5.** moverConceito(*c*, *O*, *localDeDestino*)

---

```
if temSubconceito(c, O) then  
    for all subc ∈ c do  
        deletarConceito(subc, O)  
        adicionarConceito(subc, O, localDeDestino)  
    end for  
end if  
deletarConceito(c, O)  
adicionarConceito(c, O, localDeDestino)
```

---

Fonte: Autoria própria

## **II. Edição da ontologia em evolução por mudanças relacionadas à sua terminologia de referência**

Conforme discutido na seção sobre a fase de planejamento da evolução, as mudanças a serem aplicadas na ontologia por mudanças relativas à sua terminologia de referência poderão levar à alteração terminológica da ontologia para refletir as mudanças ocorridas na terminologia, o que caracterizará a padronização parcial ou adaptativa dos termos da ontologia em evolução; ou à padronização terminológica total da ontologia devido à necessidade de realizar a sua padronização inicial ou substituição da terminologia de referência atual. Esses dois caminhos a serem seguidos para a edição da ontologia estão detalhados nas subseções seguintes.

### **a) Edição da ontologia para padronização parcial**

A partir do relatório de mudanças especificadas será possível se identificar os conceitos cujo termo usado para representá-lo sofreu mudança. Assim, o relatório de mudanças especificadas será o guia de aplicação das mudanças. A alteração dos conceitos na ontologia deverá ser iniciada pelos subconceitos pois alterações em conceitos pais são refletidas nos conceitos filhos e esses impactos podem causar erros de difícil identificação. O início do processo de alteração da ontologia pelos subconceitos favorece a visualização das mudanças e dos seus impactos.



## **b) Edição da ontologia para padronização total**

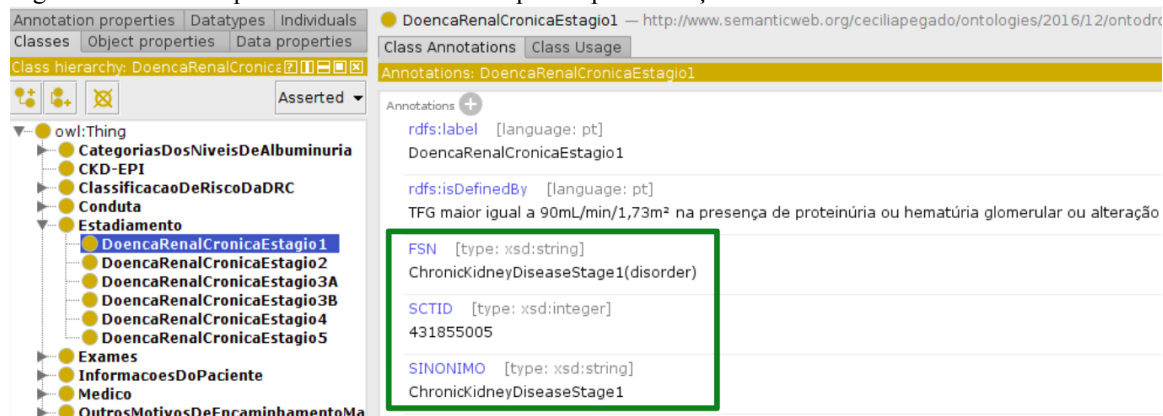
A terminologia SNOMED CT será utilizada para exemplificar as tarefas a serem aplicadas em um processo de padronização total de uma ontologia desenvolvida em OWL utilizando o editor Protégé. O processo de edição da ontologia para padronização total também deverá ser iniciado a partir dos conceitos folhas. Em geral, as terminologias clínicas possuem um identificador único para cada termo. Isso evita ambiguidades e erros causados pelo processamento dos conceitos errados por possuírem a descrição muito similar a de algum outro conceito. Assim, além do termo, o identificador do mesmo também deverá ser importado para a ontologia. Conforme discutido em seções anteriores, em SNOMED CT os termos são classificados de acordo com a sua aceitabilidade, podendo assim ser considerados preferíveis ou aceitáveis. Os termos preferíveis são os mais usados pela comunidade pertencente a determinado domínio. Desse modo, para ontologias que utilizam SNOMED CT recomenda-se que essa aceitabilidade seja considerada na escolha do termo a ser usado para representar um conceito na ontologia. Em SNOMED CT, o FSN é o termo que possui aceitabilidade preferível.

Para as ontologias nas quais os conceitos não estão representados em inglês, recomenda-se que sejam criadas *annotations* para a acomodação dos elementos capturados na terminologia. Por exemplo, em uma ontologia que definiu SNOMED CT como terminologia padrão, para o FSN deve ser criada uma *annotation* com nome sugestivo na qual para cada termo deverá ser inserida a sua descrição preferível, desse modo, essa *annotation* deverá ser definida para receber um valor do tipo *String*. O mesmo deverá ser feito para o SCTID, o identificador único dos termos pertencentes à SNOMED CT, porém a *annotation* deverá ser do tipo *Integer*. Para ontologias definidas em linguagens diferentes do inglês é importante que o termo original seja mantido, assim, a descrição principal do termo poderá ser mantida na linguagem original e a *annotation* com o FSN (para o caso de ontologias que utilizam SNOMED CT como terminologia de referência) poderá ser usada para representação dos conceitos na ontologia apenas quando necessário.

Para o processamento dos termos recomenda-se fortemente o uso do seu identificador. Isso é importante por que o identificador é único o que reduz o risco de erros ocasionados pelo processamento do termo incorreto por possuir a descrição semelhante à de outro termo.

Desse modo, a *annotation* que possui o valor do identificador poderá ser referenciada quando for necessário realizar o processamento dos conceitos. Caso deva ser considerado outro termo para representação dos conceitos além do FSN, uma *annotation* para acomodar um sinônimo poderá ser criada. Em SNOMED CT, sinônimos também possuem aceitabilidade. Considerando esse fato, recomenda-se o uso do sinônimo com aceitabilidade preferível. Na Figura 22 pode ser observado um exemplo onde foram criadas três *annotations* contendo o FSN, o SCTID e o sinônimo preferível do termo para o conceito “DoencaRenalCronicaEstagio1”, pertencente a uma ontologia sobre a DRC que utiliza a terminologia SNOMED CT. As *annotations* que receberam o FSN e o sinônimo são do tipo *String* enquanto a que recebe o SCTID é do tipo *Integer*.

Figura 22. Exemplos de *annotations* para padronização da classe “DoencaRenalCronica”



Fonte: Adaptado da ferramenta Protégé.

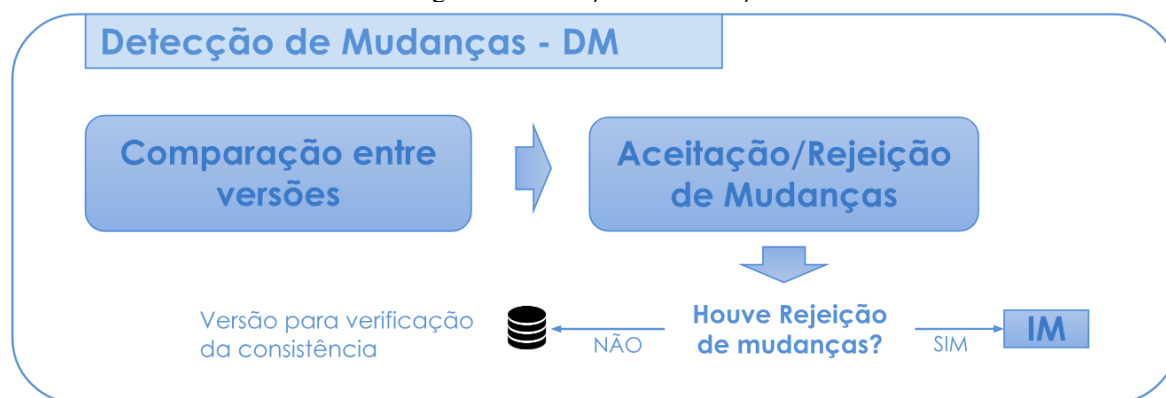
Esse processo deverá ser realizado para todos os termos da ontologia. Após a edição, a ontologia alterada é fornecida como saída da tarefa e será usada como uma versão de comparação a ser confrontada com a versão anterior à aplicação das devidas mudanças. Por meio dessa comparação é possível que se saiba exatamente as mudanças que foram realizadas. Considerando que a comparação entre as versões possui como finalidade identificar com exatidão todas as alterações sofridas pela ontologia, a ela foi dado o nome Detecção de Mudanças (DM). DM é a fase posterior à de implementação das mudanças e possui como principal objetivo fornecer subsídios para a identificação e posterior resolução das inconsistências ocorridas após a aplicação das mudanças na ontologia em processo evolutivo.

### 5.2.3 Detecção de mudanças - DM

A fase de detecção de mudanças (Figura 23) possui como objetivo identificar e analisar as mudanças que foram efetivamente aplicadas de modo a verificar se as mesmas estão em conformidade com o relatório de mudanças especificadas. É nesse subprocesso que a equipe responsável pela curadoria de mudanças verifica se as mesmas foram aplicadas corretamente, se alguma mudança indevida foi aplicada ou se alguma mudança requerida não foi realizada. Um *log* contendo todas as mudanças aplicadas deve ser obtido para que a análise das mudanças possa ser realizada.

A Detecção de Mudanças é um pré-requisito para a realização do Tratamento de Inconsistências, assim, apenas em evoluções por mudança de foco, ou mudanças nas diretrizes do domínio da ontologia será necessária a realização dessa fase do *framework*. Evoluções ocasionadas por mudanças relativas à terminologia de referência da ontologia não precisarão passar por tratamento de inconsistências, tendo em vista que o modelo lógico da ontologia não deverá sofrer alterações em evoluções decorrentes desse motivo.

Figura 23. Detecção de Mudanças



Fonte: Autoria própria.

Conforme a Definição 18 (seção 5.1) uma mudança  $m$  é detectada quando comparando-se a versão anterior  $Va$  de uma ontologia com a sua nova versão  $Vn$  são identificadas mudanças, sejam elas requeridas ou induzidas, em uma ou mais entidades da ontologia em evolução. O processo de detecção de mudanças pode ser representado por meio do Algoritmo 6.

---

Algoritmo 6. DetectaMudancas( $Ov1$ ,  $Ov2$ )

---

```
for all  $e1 \in Ov1$  do  
  for all  $e2 \in Ov2$  do  
    if  $e1 \neq e2$  then  
      returns true  
    end if  
  end for  
end for
```

---

Fonte: Autoria própria.

Conforme no Algoritmo, todas as entidades da versão anterior  $Va$  de uma ontologia  $O$  são comparadas às entidades da versão alterada  $Vn$ . Caso sejam identificadas diferenças nas entidades comparadas, o algoritmo retorna *true* assumindo a existência de mudanças entre as versões comparadas.

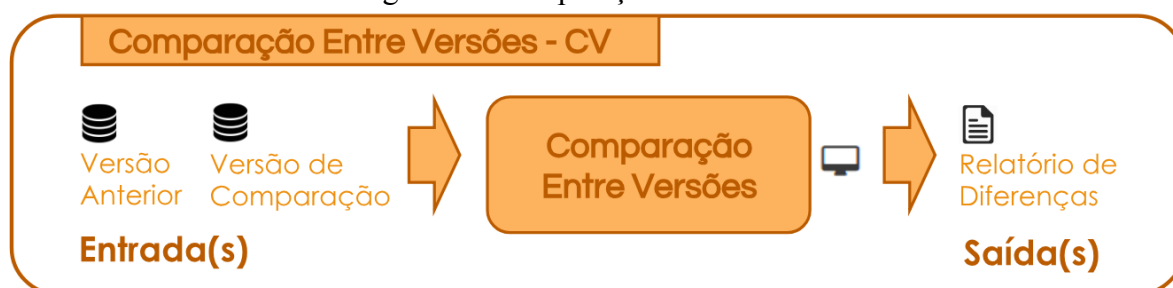
O subprocesso DM auxilia na identificação de inconsistências causadas pelas mudanças realizadas. Após a aplicação das mudanças requeridas, as entidades da ontologia podem sofrer alterações induzidas por essas mudanças e passar a não satisfazer uma ou mais condições de consistência às quais estão vinculadas. A Comparação Entre as Versões e a Aceitação ou Rejeição de Mudanças são as tarefas que deverão ser realizadas na fase de Detecção de Mudanças.

### 5.2.3.1 Comparação entre Versões (CV)

Na atividade de Comparação entre Versões (Figura 24), a versão alterada da ontologia em evolução, fornecida como saída da fase de Implementação das Mudanças é comparada à sua versão sem alterações. A comparação da versão anterior da ontologia com a sua versão atual deve ser realizada com o apoio de uma ferramenta *diff* (seção 4.2.3), pois geralmente ontologias do domínio biomédico possuem tamanhos consideravelmente grandes quando comparadas a ontologias de outros domínios o que dificulta a identificação de mudanças (GROSS, PRUSKI e RAHM, 2016). O uso de uma ferramenta *diff* ajudará a reduzir custos com tempo e com recursos humanos, por exemplo. Ferramentas *diff* também auxiliam na visualização das mudanças. Algumas alterações podem passar despercebidas quando é feita apenas a visualização da hierarquia de conceitos.

A visualização de todas as mudanças ocorridas entre as versões da ontologia em processo de evolução facilita a identificação das inconsistências pois permite que os responsáveis pela evolução possam ter acesso às entidades afetadas e conseqüentemente aos possíveis erros causados por essas mudanças. Após a comparação, um relatório de diferenças deve ser produzido contendo todas as mudanças realizadas, para ser analisado pela curadoria na atividade de Aceitação/Rejeição de Mudanças ou ARM (Figura 25).

Figura 24. Comparação Entre Versões



Fonte: Autoria própria.

Na Figura 25 pode ser observado um exemplo de *log* de mudanças fornecido pelo *plug-in* OWL Differences, ferramenta *diff* criada para ser utilizada no Protégé. Nesse exemplo, duas versões de uma ontologia sobre a DRC em evolução por mudança de foco são comparadas. O *log* de mudanças fornece, além das entidades afetadas, a operação que foi realizada nessas entidades bem como todas as mudanças decorrentes dessas alterações.

Figura 25. Descrição da remoção da classe "Nefrologista" no plug-in OWL Differences da ferramenta Protégé

	Description	Baseline Axiom
Deleted: Medico	Deleted	Class: Nefrologista
Deleted: Melfalan		
Deleted: Menor_igual		
Deleted: Menor_que		
Deleted: Metformina		
Deleted: Metotrexate		
Deleted: Monoparesias		
Deleted: Mulheres_Gravidas		
Deleted: Médicos da Atenção Primária		
Deleted: Nefrologista	Deleted	Médicos da Atenção Primária DisjointWith Nefrologista
Deleted: Neoplasia		
Deleted: Noção		
Deleted: Náuseas	Deleted	Nefrologista CN_pl "Nefrologistas"
Deleted: Obesidade		
Deleted: Obstrução do Trato Urinário		
Deleted: Obter	Deleted	Nefrologista CN_sg "Nefrologista"
Deleted: Opióides		
Deleted: Paciente		
Deleted: Parda		
Deleted: Penicilina	Deleted	Nefrologista Definição "Nefrologia é uma especialidade do âmbito da Medicina Interna que se dedica ao diagnóstico e ao tratamento das doenças do rim;"^^string
Deleted: Perda da Concentração		
Deleted: Pericardite		
Deleted: Peritonite	Deleted	Nefrologista Sinônimo "Médico nefrologista"^^string
Deleted: Peso Corporal		
Deleted: Pleurite		

Fonte: OWL Differences - Protégé.

Como pode ser visualizado na Figura, a remoção do conceito “Nefrologista” causou a remoção de todas as suas referências na ontologia. Esses impactos são detalhadamente exibidos no *log* de mudanças de uma ferramenta *diff*. A partir desse *log*, o relatório de diferenças para apoiar a tarefa de análise das mudanças para aceitação ou rejeição das mesmas poderá ser criado. Os ontologistas poderão decidir se criarão um relatório de diferenças ou se utilizarão o *log* de mudanças fornecido pela ferramenta *diff* para a realização dessa tarefa, tendo em vista que a criação do relatório poderá ser uma tarefa trabalhosa dependendo da quantidade de alterações realizadas na ontologia. Entretanto, o relatório de mudanças poderá ser útil para análises durante o processo de evolução, caso seja necessário, ou poderá ser utilizado em evoluções posteriores.

### **5.2.3.2 Aceitação/Rejeição de Mudanças**

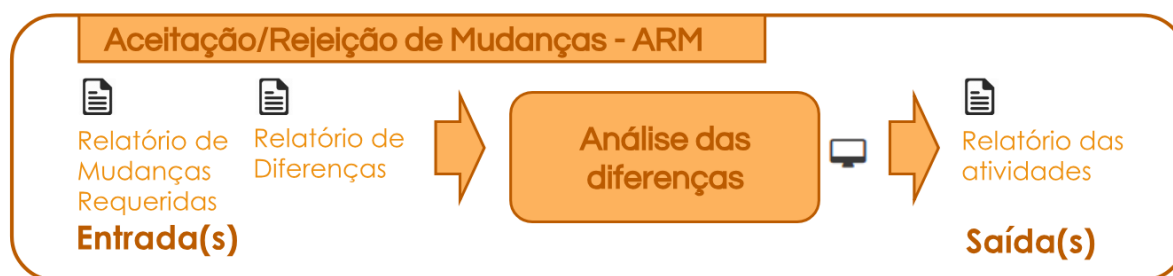
O relatório contendo as mudanças especificadas fornecido como saída do primeiro subprocesso do *framework* (PE) bem como o relatório de diferenças entre a versão anterior e a versão de comparação fornecido pela atividade de Comparação entre Versões são elementos essenciais para a execução da atividade de aceitação ou rejeição de mudanças (Figura 26). Por meio do uso desses relatórios será possível verificar se as mudanças sofridas pela ontologia estão em conformidade com aquelas que foram requeridas. A análise das mudanças é uma tarefa de alto nível, tendo em vista que a análise considerará apenas a conformidade das mudanças aplicadas com as requeridas para evitar que aquelas que não foram especificadas permaneçam na ontologia. Assim, durante o processo não serão exigidos detalhamentos, relacionados ao tipo de mudanças dos conceitos nos quais deverão ser feitas as análises, por exemplo.

Após a análise, um relatório das atividades realizadas nessa tarefa deverá ser apresentado como saída. Nele deverá conter informações sobre os envolvidos no processo de aceite ou rejeite e, em caso de aceite de todas as mudanças, uma observação acerca desse fato; em caso de rejeição de alguma mudança, as mesmas deverão estar detalhadamente descritas com informações sobre as entidades cujas mudanças foram rejeitadas com vistas a facilitar o processo de reimplementação das mesmas.

Caso todas as mudanças sejam aceitas a versão alterada da ontologia estará pronta para ser encaminhada para a fase onde as inconsistências consequentes das alterações realizadas

que foram detectadas em seu conteúdo serão tratadas no subprocesso nomeado Tratamento de Inconsistências ou TI. Caso haja rejeição de mudanças, o processo de evolução deverá voltar para o subprocesso IM, onde as mudanças aplicadas incorretamente deverão ser reimplementadas, as indesejadas desfeitas e as requeridas mas não aplicadas, aplicadas conforme solicitado no plano de evolução. De modo que o retorno a esse subprocesso não cause um aumento tão significativo no tempo necessário para a realização de todo o processo evolutivo, um relatório de atividades, contendo todas as mudanças rejeitadas e cujo intuito é direcionar o desenvolvedor para o local exato onde os problemas foram encontrados, deverá ser fornecido como entrada para esse subprocesso.

Figura 26. Aceitação/Rejeição de Mudanças

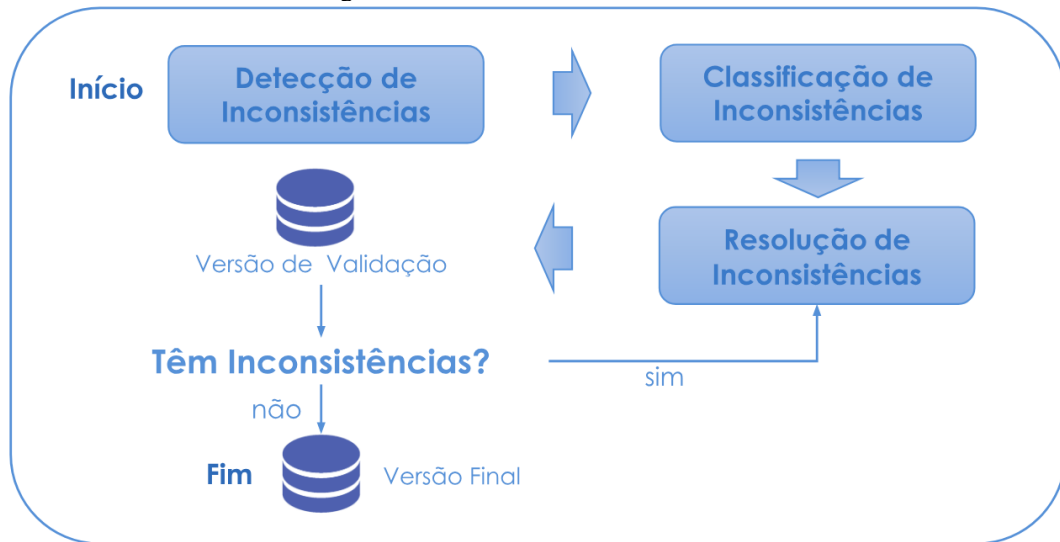


Fonte: Autoria própria

#### 5.2.4 Tratamento de Inconsistências - TI

Neste subprocesso (Figura 27), a ontologia é verificada tendo como objetos de análise suas condições de consistência. Conforme a Definição 19 (seção 5.1), uma condição de consistência é uma verdade absoluta determinada tendo como base um domínio específico. É a partir dessa verdade absoluta ou axioma que uma entidade da ontologia será considerada consistente. Após a análise para identificação das inconsistências, a ontologia deverá ser tratada de modo que as inconsistências ocorridas sejam resolvidas, e as questões de competência que definem os limites da ontologia no âmbito do seu domínio sejam corretamente respondidas como forma de avaliar se esta continua cumprindo a finalidade para a qual foi criada. O processo de tratamento de inconsistências também tem como objetivo permitir que os elementos que dependem da ontologia (outras ontologias, Sistemas de Informação etc.) continuem operando dentro dos padrões de consistência exigidos pelas regras do domínio.

Figura 27. Tratamento de Inconsistências



Fonte: Autoria própria.

Como pode ser visto na Figura 27, o subprocesso TI está dividido em 3 atividades principais que estão interligadas e fornecem subsídios para suas efetivas realizações. A primeira fornece suporte para a detecção de inconsistências, a segunda para a classificação dessas inconsistências e a terceira para o tratamento das inconsistências detectadas e classificadas.

O processo de tratamento de inconsistências é iniciado pela detecção das mesmas. O relatório de mudanças detectadas fornecido como saída do processo de detecção de mudanças será usado para apoio à detecção de inconsistências. Conforme discutido na seção anterior as ferramentas *diff* fornecem a descrição dos impactos causados pelas mudanças. Essas descrições auxiliam na detecção das inconsistências porque ao se saber os elementos ontológicos que foram afetados pelas mudanças é mais fácil se identificar os erros ocasionados por elas.

Após a detecção, as inconsistências deverão ser classificadas em lógicas ou estruturais, essa classificação é importante pois será possível definir uma ordem para resolução dos problemas. A tarefa de resolução de inconsistências é a próxima desse subprocesso. Nela, os problemas estruturais deverão ser resolvidos antes dos lógicos pois muitas das inconsistências lógicas são decorrentes das estruturais. Assim, a resolução das inconsistências estruturais, poderá, indiretamente, resolver os problemas lógicos delas decorrentes. Cada uma dessas



etapas são detalhadamente explicadas nas próximas subseções.

#### 5.2.4.1 Detecção de Inconsistências

Após a aplicação de mudanças requeridas, as entidades da ontologia podem sofrer alterações induzidas por essas mudanças e passar a não satisfazer uma ou mais condições de consistência às quais estão vinculadas. Com a finalidade de apoiar a detecção dessas mudanças existem as ferramentas *diff*, que permitem a visualização das diferenças entre duas versões de uma ontologia. A visualização de todas as mudanças ocorridas entre as versões da ontologia em processo de evolução facilita a identificação das inconsistências pois permite que os responsáveis pela evolução possam ter acesso às entidades afetadas, o que reduz o tempo gasto com o processo tendo em vista que a detecção de mudanças não precisará ser realizada manualmente.

A existência de inconsistências em uma ontologia pode ser verificada pela análise de todas as suas entidades com base nas condições que determinam sua consistência. O Algoritmo 7 representa o processo de verificação da consistência de entidades pertencentes a uma ontologia em processo evolutivo tendo como base suas condições de consistência.

Conforme na Definição 21,  $T$  representa o conjunto de condições de consistência de uma ontologia. Em um conjunto de condições de consistência  $T$  estão todas os requisitos que permitem afirmar que uma ontologia encontra-se em estado de consistência. O Algoritmo 7 retorna *true* caso uma entidade que faça parte de uma ou mais condições de consistência da ontologia em análise, não satisfaça a essas condições, tornando a ontologia inconsistente.

---

**Algoritmo 7.** verificarConsistencia( $e$ )

---

```
for all  $\beta \in T$  do  
  if  $e \in \beta \wedge e$  não satisfaz  $\beta$  then  
    true  
  end if  
end for
```

---

Fonte: Autoria Própria

A detecção de inconsistências em uma ontologia  $O$  pode ser realizada por meio da análise de todas as entidades pertencentes ao conjunto de entidades dessa ontologia. Em caso de identificação de alguma entidade que não satisfaça à condição de consistência à qual está

vinculada assume-se que a mesma é uma entidade inconsistente. No Algoritmo 8, onde esse processo está representado, o Algoritmo 7 é usado para verificar se determinada entidade está em estado de inconsistência, caso esteja, adiciona-se tal entidade a um conjunto de entidades inconsistentes  $\alpha$  para posterior aplicação de correções que a permitam retornar a um estado de consistência.

---

**Algoritmo 8.** `detectarInconsistencias(O)`

---

```

for all  $e \in O$  do
  if verificarConsistencia(e) then
     $\alpha := \alpha \cup e$ 
  end if
end for

```

---

Fonte: Autoria Própria

Na Figura 28 é apresentada a atividade de detecção de inconsistências do *framework* como um processo organizado de acordo com a ordem de realização de cada uma das subtarefas que o compõem. Como entrada dessa fase deve ser fornecido o relatório de mudanças detectadas. A saída deverá ser um relatório com as inconsistências identificadas. Na Tabela 12 pode ser observado um exemplo de relatório de mudanças detectadas criado a partir do *log* de mudanças fornecido pela ferramenta *diff* na fase de Detecção de Mudanças. No exemplo, foram comparadas duas versões de uma ontologia sobre a DRC em evolução por mudança de foco. A coluna “Novos Axiomas” apresenta os impactos das mudanças aplicadas na entidade apresentada na coluna “Entidade Afetada”.

Tabela 12. Exemplo de relatório de mudanças detectadas em uma ontologia sobre a DRC em evolução por mudança de foco

---

**Ontologia em Evolução:** ontoDecideDRC

---

**Motivo da evolução:** Mudança de foco

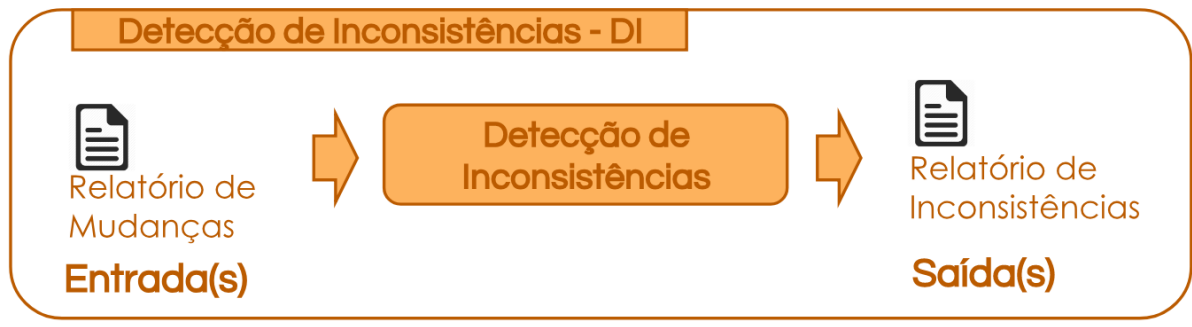
---

Entidade Afetada	Descrição da Mudança	Novos Axiomas
Conduta	Adição	-Class: Conduta; -rdfs:label "Conduta"; -AnnotationProperty: Conduta; -rdfs:isDefinedBy “Formas de manejo clínico do paciente com DRC conforme o estadiamento”
CondutaEstagio1	Adição	Class: CondutaEstagio1; CondutaEstagio1 SubClassOf Conduta

AcidoseMetabolica	Remoção	
Antibiotico	Remoção	
<i>isDefinedBy</i>	Alteração	<i>isDefinedby Range string</i>
Realiza	Alteração	<i>Realiza InverseOf E_ Realizado</i>

Fonte: Autoria Própria

Figura 28. Detecção de Inconsistências

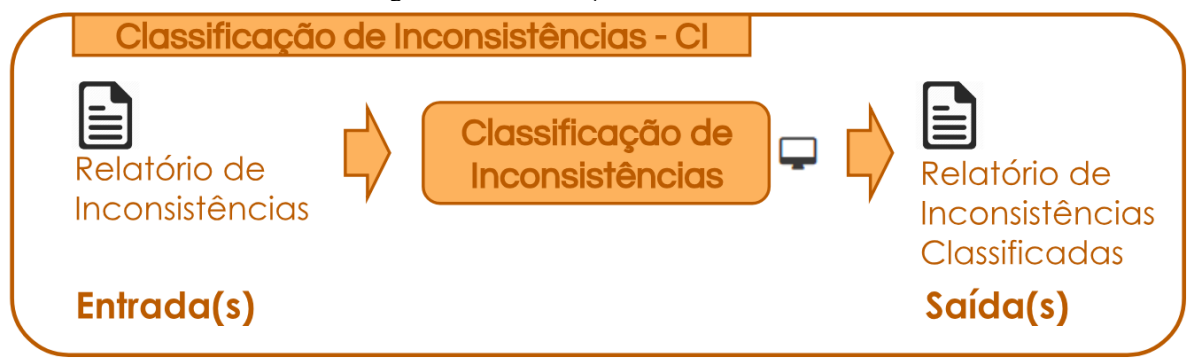


Fonte: Autoria própria

**5.2.4.2 Classificação de Inconsistências – CI**

Segunda tarefa da fase de Tratamento de Inconsistências (Figura 29), a classificação de inconsistências possui como objetivo direcionar a tarefa onde as inconsistências ocorridas na ontologia deverão ser resolvidas. Para a realização dessa fase, o relatório de inconsistências detectadas na fase anterior (seção 5.2.4.1) deverá ser utilizado. Nesta atividade, as inconsistências detectadas deverão ser classificadas como estruturais ou lógicas.

Figura 29. Classificação de Inconsistências



Fonte: Autoria própria

Conforme a Definição 24 (seção 5.1), as inconsistências estruturais são aquelas por meio das quais, os componentes da linguagem utilizada no desenvolvimento da ontologia são

afetados pelas mudanças. Por exemplo, em OWL, classes, indivíduos e propriedades são componentes centrais da linguagem para os quais os conceitos de determinado domínio são convertidos de modo que sejam representados ontologicamente. Nesse sentido, inconsistências estruturais no contexto de uma ontologia desenvolvida em OWL se manifestam por meio de erros nesses componentes. Nas inconsistências lógicas, as regras que definem a consistência da ontologia, definidas por meio de axiomas, são afetadas pelas mudanças aplicadas e passam a não satisfazer os requisitos definidos como condições de consistência do domínio representado.

Ao direcionar o tratamento de inconsistências da ontologia, a classificação permite tornar menos oneroso o processo de identificação das mudanças estruturais para que estas sejam aplicadas primeiro e os problemas relacionados às regras da ontologia relacionados a essas mudanças sejam conseqüentemente resolvidos. Por exemplo, em uma ontologia sobre a DRC em evolução por mudança de foco, determinou-se que a classe “Nefrologista” deveria ser removida. Tendo em vista que o foco anterior da ontologia exigia que fossem feitas diversas referências à essa classe, os impactos da deleção da mesma ocasionaram erros nos locais onde são realizadas essas referências. Esses erros podem ser classificados como inconsistências estruturais. Enquanto tais erros permanecerem na ontologia não poderão ser feitas inferências corretas sobre a regra na qual o conceito deletado está sendo referenciado. Esse problema pode ser classificado como uma inconsistência lógica.

A deleção das regras que utilizam a classe “Nefrologista” ou a alteração das mesmas para fazerem referência a outro conceito, são exemplos de estratégias de correção das inconsistências estruturais. Ao serem aplicadas essas correções, as inferências da ontologia retornarão a um estado de consistência, porém, esse estado de consistência estará relacionado a novas condições de consistência de modo que as inferências da ontologia possam refletir as alterações realizadas.

Após a classificação, deverá ser fornecido um relatório de inconsistências classificadas. Além de apoiar a realização da próxima fase, esse relatório poderá ser usado em evoluções posteriores para, por exemplo, prever a ocorrência de inconsistências baseando-se naquelas mais comuns. O uso de um *software* para ajudar nessa fase é altamente recomendado com vistas a dar praticidade ao processo e facilitar o reuso dos dados. O relatório de inconsistências classificadas deverá ser produzido a partir do relatório de inconsistências

detectadas fornecido como saída da tarefa de detecção de inconsistências (seção 5.2.4.1). Na Tabela 13 pode ser observado um exemplo de relatório de inconsistências classificadas de uma ontologia sobre a DRC em evolução por mudança de foco.

Tabela 13. Exemplo de relatório de inconsistências classificadas para uma ontologia sobre a DRC em evolução por mudança de foco

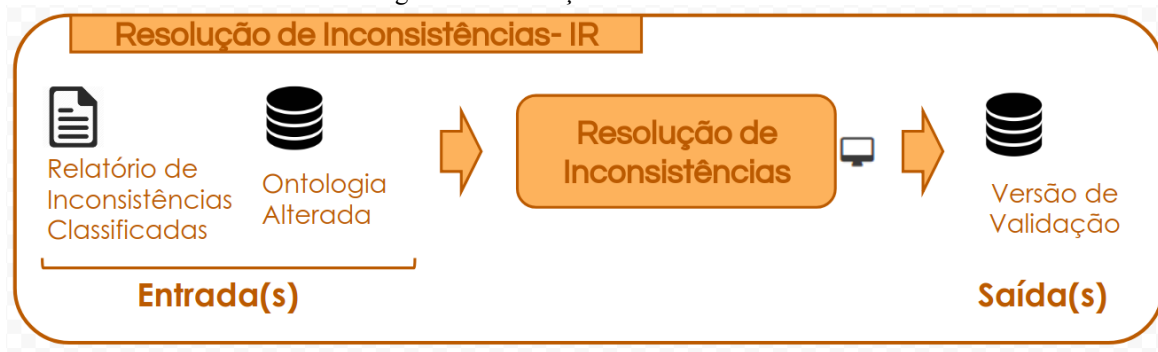
<b>Ontologia em Evolução:</b> ontoDecideDRC				
<b>Motivo da Evolução:</b> Mudança do foco da ontologia				
<b>Entidade Afetada</b>	<b>Tipo de Entidade</b>	<b>Mudança aplicada</b>	<b>Inconsistências</b>	
			<b>Estruturais</b>	<b>Lógicas</b>
EAplicadoNo	Propriedade	Alteração	-Erros nos axiomas que utilizam essa propriedade -Impossível encontrar a propriedade de objeto inversa a Aplicar	Impossível inferir sobre as regras que utilizam essa propriedade
Exames	Classe	Remoção	-Alteração da estrutura hierárquica dos exames organizados como subclasse da classe exames -Erros nas referências da classe exames em suas subclasses	Impossível inferir sobre as regras que utilizam essa classe
GrauDeRisco DaDRC	Classe	Remoção	Erros nos axiomas que fazem a classificação de risco da DRC	Impossível determinar o grau de risco de DRC do paciente

Fonte: Autoria própria

### 5.2.4.3 Resolução de inconsistências

A última atividade da fase de tratamento de inconsistências (Figura 30) apresenta estratégias de correção dos problemas identificados e classificados conforme abordado nas seções anteriores. Essas estratégias devem ser definidas com base nos tipos de erros por meio dos quais as inconsistências podem ser classificadas bem como com base no domínio representado na ontologia em evolução. Ainda que o *framework* definido seja voltado para ontologias Biomédicas, sabe-se que tal área detém diversas subáreas de amplitudes diferentes e que exigem maneiras distintas de representação por meio dos recursos ontológicos. Esse fato define a complexidade da ontologia e conseqüentemente ajuda na definição dos métodos de resolução das inconsistências. O processo de remoção de inconsistências de uma ontologia pode ser realizado por meio da busca e correção das inconsistências detectadas.

Figura 30. Resolução de Inconsistências



Fonte: Autoria própria

O processo de resolução está representado no Algoritmo 9, onde para todas as inconsistências  $i \in \alpha$  detectadas em uma ontologia  $O$ , deve-se verificar todas as entidades  $e \in E \in O$  e uma(s) solução(ões) necessária(s) (representada(s) pelo método `removeInconsistencias(e,i)` deve(m) ser aplicada(s) àquelas onde existam inconsistências de acordo com o domínio da ontologia em processo de tratamento de inconsistências. Conforme a Definição 23 (seção 5.1),  $\alpha$  é igual a um conjunto de entidades  $E$  que por terem sofrido mudanças, detêm inconsistências devido as quais não é possível satisfazer a qualquer condição de consistência da ontologia em processo evolutivo.

Conforme o Algoritmo 9, para todas as entidades de uma ontologia  $O$  verifica-se a ocorrência de alguma das inconsistências identificadas. Caso a entidade verificada tenha sido afetada por inconsistências, uma solução direcionada para o tipo de erro ocorrido e para o domínio da ontologia, deverá ser aplicada. Tal solução deverá ser aplicada até que a entidade volte a satisfazer a condição ou condições de consistência às quais pertence.

Essa atividade apenas será necessária para ontologias em evolução por mudança de foco ou por alterações nas suas diretrizes de domínio. Ontologias em evolução por mudanças relativas à sua terminologia de referência não sofrem alterações que venham a afetar as regras que determinam a sua consistência, portanto, não será necessário tratar inconsistências em ontologias em evolução por esse motivo.

---

**Algoritmo 9.** `removeInconsistencias( $O, \alpha$ )`

---

**for all**  $i \in \alpha$  **do**

```

for all  $e \in O$  do
  if  $\exists i \in e$  then
    repeat
      removeInconsistencias( $e, i$ )
    until verificarConsistencia( $e$ ) returns false
  end if
end for
end for

```

---

Fonte: Autoria própria

Nesta atividade recomenda-se fortemente o acompanhamento especializado, pois durante o processo de resolução das inconsistências o desenvolvedor de ontologias pode ter dúvidas relativas ao domínio da ontologia em evolução por não possuir conhecimento especializado o suficiente para determinar a correção de problemas cuja solução depende de conhecimentos profundos sobre o domínio abordado, e que não poderão ser conhecidas apenas por meio do uso do relatório de inconsistências classificadas, que é um documento técnico com a finalidade de apresentar as inconsistências sem detalhamentos relativos a como resolvê-las. O estudo de caso descrito no capítulo 6, que aborda a evolução de uma ontologia sobre a DRC, teve o acompanhamento de uma médica especialista em Nefrologia, que compõe o grupo de pesquisadores do laboratório onde esta pesquisa foi realizada. Além do relatório de inconsistências e do acompanhamento especializado, nessa fase pode ser necessário o uso das fontes de conhecimento da ontologia.

As estratégias de correção de inconsistências estão diretamente relacionadas aos tipos de erros causados por elas. Desse modo, neste trabalho, as formas de correção dos problemas que comprometeram a consistência da ontologia serão abordadas levando em conta os três tipos de erros que causam inconsistências e ambiguidades na ontologia em processo evolutivo. Conforme Fahad e Qadir (2008), esses erros são chamados de erros circulatorios, erros de partição e erros de inconsistências semânticas. A partir da identificação desses erros nos problemas ocorridos na ontologia, pode-se definir formas de correção dos mesmos. Dessa forma, antes de aplicar as correções é importante que se identifique em qual dos tipos de erros, dentre os três abordados no *framework*, o problema ocorrido se encaixa.

Conforme a Definição 27 (seção 5.1), os erros circulatorios ocorrem quando uma classe de uma ontologia é definida como subclasse de si mesma em qualquer nível da

hierarquia dessa ontologia. Para resolver inconsistências relacionadas a esse erro, será necessária a análise da hierarquia da ontologia iniciando pela classe base. A presença do especialista do domínio representado na ontologia durante esse processo é essencial pois a taxonomia dos conceitos de uma ontologia reflete a taxonomia desses conceitos no domínio do discurso ou de interesse. Assim, tendo em vista que os ontologistas podem não conhecer o domínio abordado o suficiente para estabelecer a estrutura hierarquica correta da ontologia, torna-se imprescindível a presença do especialista de domínio para auxiliar nesse processo. A correção dos erros da ontologia deverá ser feita por meio da edição da mesma após a identificação correta do local onde ocorreu o problema. A resolução dos erros circulatorios poderá ser feita seguindo-se a ordem: análise – identificação – correção.

De acordo com a Definição 28 (seção 5.1), os erros de partição dependem do tipo de decomposição de superclasses em subclasses. A decomposição citada nessa definição pode ser classificada como disjunta ou exaustiva. Considerando o exposto na Definição 31 (seção 5.1), os erros de partição advindos da decomposição disjunta ocorrem quando são definidos conceitos como subconceitos de conceitos disjuntos, essa definição é aplicável a classes e instâncias de uma ontologia em evolução (FAHAD e QADIR, 2008).

A solução para esse problema pode ser feita seguindo-se dois caminhos distintos e dependentes do domínio representado. A primeira solução possível envolve desfazer a relação de disjunção entre os conceitos que detêm subconceitos comuns, assim, tais conceitos não serão mais disjuntos e portanto poderão compartilhar os mesmos subconceitos. Entretanto, caso a relação de disjunção entre os conceitos não possa ser desfeita, pode-se aplicar a segunda solução possível, a qual está relacionada à análise da hierarquia dos conceitos partindo do conceito base ou raiz com a finalidade de corrigir as decomposições das superclasses em subclasses e estabelecer a organização taxonômica correta dos conceitos representados de acordo com o domínio de interesse. A decisão sobre qual caminho deverá ser seguido para corrigir os erros de partição ocasionados pela decomposição disjunta deverá ser acompanhada da opinião do especialista do domínio da ontologia, pois as ligações corretas entre os conceitos estão diretamente relacionadas ao domínio abordado.

Os erros de partição também podem ser advindos da decomposição exaustiva. Conforme a Definição 32 (seção 5.1), esses erros ocorrem quando são feitas decomposições ou partições exaustivas de uma classe em muitas subclasses porém nem todas as instâncias da



classe raiz pertencem às suas subclasses (FAHAD e QADIR, 2008). Para resolver inconsistências ocasionadas por esse tipo de erro existem duas soluções possíveis.

A primeira exige a remoção da instância ou das instâncias pertencentes à classe raiz que não são instâncias de todas as suas subclasses. Essa solução fará com que não mais existam na classe base, instâncias que não podem estar vinculadas às suas subclasses. Caso a remoção das instâncias não possa ser realizada, a segunda solução possível envolve a revisão da hierarquia dos conceitos que possuem esse erro a partir do conceito raiz com a finalidade de identificar quais subconceitos não podem estar vinculados a uma ou mais instâncias do conceito base e assim, decidir entre a remoção ou a aplicação de uma operação de mover sobre o subconceito que não pode utilizar as mesmas instâncias da classe raiz. Essa solução possui como finalidade redefinir a hierarquia à qual pertencem os conceitos afetados e reestabelecer a completude da decomposição ou particionamento.

Conforme na Definição 33, os erros de inconsistências semânticas ocorrem quando um conceito é classificado como subclasse de outro ao qual ele não pertence. Ou seja, o significado de um conceito é incorretamente interpretado fazendo com que a sua organização hierárquica seja estabelecida de forma incorreta. O primeiro passo para solucionar esse problema é a identificação, na ontologia, de todos os locais onde o mesmo ocorreu. Para auxiliar nessa tarefa poderá ser usado o relatório de inconsistências identificadas e classificadas (FAHAD E QADIR, 2008).

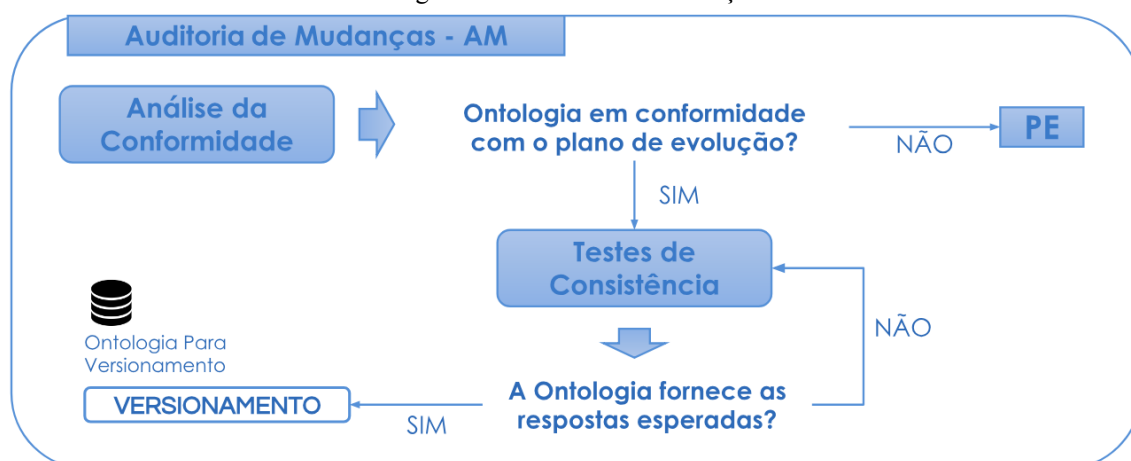
A correção poderá ser seguida da identificação e poderá considerar dois caminhos possíveis. O primeiro envolve a realização de uma operação de mover sobre o subconceito com o intuito de colocá-lo no local correto considerando a similaridade do seu significado com a do conceito destino. A segunda alternativa de resolução desse problema é a deleção do conceito e será realizada apenas caso não se consiga encaixar o conceito em uma nova hierarquia de classes. Ou seja, a segunda alternativa de correção será aplicada como solução residual.

### **5.2.5 Auditoria de Mudanças - AM**

Na Auditoria de Mudanças (Figura 31) é feita a análise da conformidade da ontologia com o plano de evolução e os testes de verificação da consistência da mesma. Assim, o subprocesso AM está dividido em duas atividades chamadas Análise da Conformidade e

Testes de Consistência. Tais atividades deverão ser aplicadas nessa ordem. Os testes de consistência da ontologia apenas poderão ser realizados caso a mesma esteja em conformidade com o plano de evolução, ou seja, a segunda atividade do subprocesso AM é dependente da primeira.

Figura 31. Auditoria de Mudanças



Fonte: Autoria própria

Conforme a Figura 31, caso a ontologia não esteja em conformidade com o que foi planejado para o processo evolutivo, a mesma deverá ser encaminhada para o Planejamento da Evolução, onde, de acordo com as considerações feitas pelo auditor, será feito um novo relatório de mudanças a serem aplicadas considerando apenas aquelas que não foram implementadas da maneira correta. Desse modo, a ontologia deverá passar novamente por todas as fases do *framework*.

Caso a ontologia esteja de acordo com o que foi planejado na fase inicial do *framework*, a mesma poderá ser encaminhada para a atividade onde a sua consistência será verificada. Os testes deverão ser realizados por meio do uso de consultas baseadas nas questões que definem a competência da ontologia no contexto do seu domínio. Essas consultas deverão ser aplicadas até que a ontologia forneça as respostas esperadas. O fornecimento dessas respostas permite determinar se a consistência da ontologia foi mantida. Assim, caso para todas as questões traduzidas, a ontologia forneça o retorno esperado a mesma poderá ser encaminhada para o versionamento.

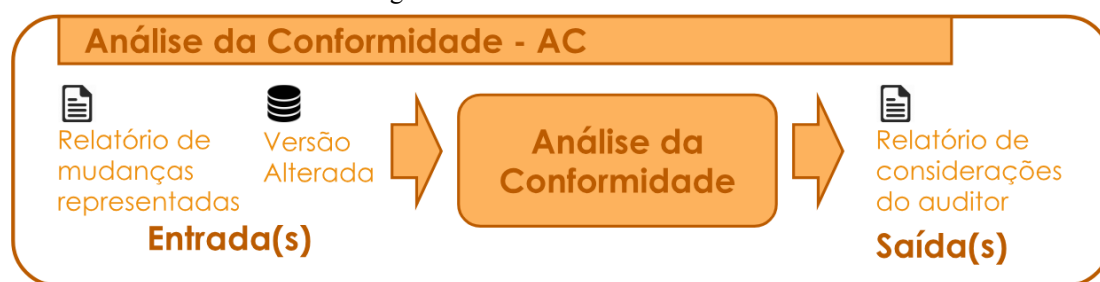
Apenas ontologias em evolução por mudanças em suas diretrizes de domínio ou por mudança de foco deverão passar pelos testes de consistência. Aquelas que estão em evolução

para refletir mudanças ocorridas em sua terminologia de referência apenas serão submetidas à análise da conformidade com o plano de evolução.

### 5.2.5.1 Análise da conformidade

A análise da conformidade (Figura 32) com o plano de evolução deverá considerar a correte e completude da aplicação das mudanças. Por esse motivo, tal atividade deverá receber como entrada, o relatório de mudanças representadas e a versão alterada da ontologia em evolução. O relatório de mudanças representadas, fornecido como saída do subprocesso Representação de Mudanças, deverá ser usado para direcionar a análise das mudanças aplicadas na ontologia e assim verificar a conformidade das alterações planejadas com as que foram efetivamente aplicadas.

Figura 32. Análise da Conformidade



Fonte: Autorial Própria

Para análise deverá ser considerada a correte da mudança, ou seja, uma mudança será invalidada caso não tenha sido aplicada da forma correta. Por exemplo, em uma ontologia sobre a DRC em evolução por mudança de foco, a propriedade planejada para ter a descrição “EObtidoPor” foi efetivamente criada com a descrição “EobtidoNo”. Devido ao erro, semanticamente a propriedade não mais reflete a finalidade para a qual foi criada tendo em vista que o uso da mesma com a descrição incorreta só seria útil em um contexto diferente daquele para o qual a mesma foi planejada.

A completude das mudanças também deverá ser considerada durante a análise da conformidade da ontologia com o seu plano de evolução. Segundo esse objeto de análise, a ontologia deverá conter todas as mudanças planejadas. Mudanças que não foram aplicadas deverão ser sinalizadas utilizando o relatório de mudanças representadas para que possam ser reimplementadas. Uma cópia do relatório de mudanças representadas poderá ser usado para

auxiliar no processo de adição de comentários nas mudanças incorretas ou não aplicadas. Tais mudanças estarão invalidadas até serem analisadas e adicionadas a um novo plano de evolução para que sejam reimplementadas. Desse modo, o relatório com as considerações do auditor será usado para auxiliar no projeto de um novo relatório de mudanças a serem aplicadas.

#### 5.2.5.2 Testes de consistência

Terminado o processo de análise, caso não tenham sido encontrados problemas em relação à conformidade das alterações da ontologia com as mudanças planejadas, os testes de verificação da consistência da ontologia poderão ser realizados. A Figura 33 ilustra o processo no qual a consistência da ontologia é testada. Os testes de consistência utilizarão as Questões de Competência da ontologia. Assim, as mesmas deverão ser convertidas em consultas por meio de alguma linguagem que possa ser utilizada para essa finalidade. Devido a sua alta expressividade, o que permite a tradução de um número maior de questões de competência, recomenda-se o uso da Linguagem SPARQL (BORTOLATO e PAULO, 2014). A criação das questões de competência bem como a sua tradução para a linguagem de consulta escolhida são subtarefas a serem realizadas antes do processo de testes.

A criação das QCs da ontologia deverá ser realizada pelo especialista do domínio da mesma. A finalidade para a qual a ontologia foi criada deverá direcionar a criação dessas questões. Por exemplo, uma QC de uma ontologia sobre a DRC pode ser escrita do seguinte modo: “Quais os estágios da DRC?”. Após criadas, as QCs deverão ser encaminhadas para que os ontologistas possam realizar a tradução das mesmas. O processo de tradução poderá ou não utilizar um *software* específico. Em contrapartida, a aplicação das consultas na ontologia deverá ser feita por meio de um software que possua essa finalidade. Caso a linguagem escolhida tenha sido SPARQL, recomenda-se o uso do *plug-in* SPARQL *Query* que pode ser usado na ferramenta Protégé. A consulta SPARQL criada a partir da QC usada como exemplo acima (“Quais os estágios da DRC?”) pode ser visualizada na Listagem 1.

---

Listagem 1. Consulta SPARQL para a QC “Quais os estágios da DRC?”

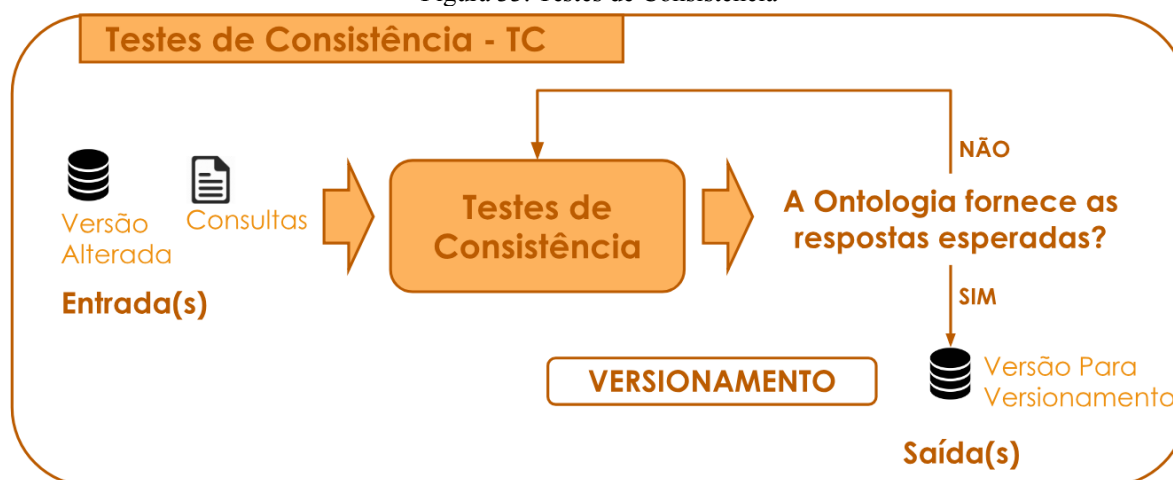
---

```
SELECT ?subject
WHERE {
  {?subject rdfs:subClassOf ontodrc:Estadiamento } }
```

---

Fonte: Autoria Própria

Figura 33. Testes de Consistência



Fonte: Autoria própria

Durante os testes de consistência da ontologia, caso não sejam obtidas respostas para determinadas consultas, a correção da ontologia deverá ser realizada como um processo concomitante aos testes. Após a correção, a consulta para a qual não foi fornecida resposta, deverá ser reaplicada. Esse processo deverá se repetir até que a resposta esperada seja fornecida. Caso a ontologia esteja em evolução por mudanças em suas diretrizes de domínio, esta poderá ser validada utilizando questões aplicadas em evoluções anteriores. Porém, caso o motivo da evolução da ontologia seja sua mudança de foco, novas questões deverão ser necessariamente criadas refletindo o novo foco da mesma.

Caso o processo evolutivo tenha ocorrido de acordo com o plano de evolução e a consistência da ontologia tenha sido mantida, a mesma poderá ser adicionada à ferramenta de versionamento escolhida para armazenar suas versões.

### 5.3 Conclusão do Capítulo

Para facilitar o entendimento do *framework*, na seção 5.1 deste capítulo foram apresentadas as definições e anotações utilizadas ao longo da descrição mesmo. Na seção 5.2 foram apresentados os processos do *framework* construído bem como as atividades que os compõem. Na descrição de cada uma das etapas do *framework* foram consideradas três perspectivas de evolução às quais a ontologia em evolução pode ser submetida. Tais perspectivas são baseadas nos três motivos de evolução identificados no estudo de caso utilizado para a obtenção de requisitos para a construção desse *framework*. Esse estudo está

descrito na seção 4.1.2. As perspectivas de evolução citadas são a mudança de foco da ontologia, as mudanças realizadas em suas diretrizes de domínio e as mudanças relacionadas à sua terminologia de referência.

A mudança de foco de uma ontologia acontece quando é necessário representar conceitos relativos a um conteúdo distinto do que está sendo abordado atualmente, ainda que pertencentes ao mesmo domínio da ontologia. Mudanças nas diretrizes da ontologia podem ocorrer caso sejam necessárias mudanças em sua estrutura hierárquica ou lógica motivadas por alterações em suas fontes de conhecimento. As mudanças na terminologia de referência da ontologia ocorrem caso tenham sido aplicadas mudanças nos termos relativos ao domínio representado na ontologia.

O *framework* está dividido em 5 fases denominadas Planejamento da Evolução, Implementação das Mudanças, Detecção de Mudanças, Tratamento de Inconsistências e Auditoria de Mudanças. No Planejamento da Evolução devem ser realizadas duas atividades: a Análise das Considerações do Auditor e a Representação das Mudanças. Se o Planejamento da Evolução da ontologia estiver sendo realizado pela primeira vez durante o processo evolutivo atual, ou seja, caso o processo evolutivo não esteja sendo executado com a finalidade de corrigir inconformidades com o plano de evolução, deverá ser realizada a Captura das Mudanças. Caso contrário, as observações feitas durante o processo de análise da conformidade da ontologia com o plano de evolução, deverão ser analisadas para que os problemas detectados sejam corrigidos por meio da submissão da ontologia a um novo processo evolutivo.

Na atividade de Captura das Mudanças, aquelas a serem implementadas na ontologia deverão ser identificadas em suas fontes de conhecimento. Após a captura deverá ser realizada a Representação das Mudanças a serem aplicadas. Representar as mudanças que a ontologia sofrerá em seu processo evolutivo é importante para facilitar a sua implementação quando a edição da ontologia for realizada. Ao serem representadas, as mudanças deverão ter seu tipo especificado, tais tipos podem ser alteração, adição ou remoção de conceitos da ontologia. Para ontologias em evolução por mudança de foco os tipos de mudanças poderão ser adições de novos conceitos, alterações de conceitos existentes, de modo a manter-se algum conteúdo pertencente ao foco anterior da ontologia, porém útil para o novo foco e; remoção de conceitos que não fazem parte do novo foco da ontologia. Mudanças em ontologias em

evolução por mudanças em suas diretrizes de domínio poderão ser especificadas como adições de novos conceitos, alterações de conceitos existentes e remoção de conceitos postos em desuso pela diretriz. Em contrapartida, mudanças relativas à terminologia de referência da ontologia poderão ser especificadas como inclusões de novos conceitos adicionados à terminologia padrão utilizada pela ontologia em evolução, melhoramento de termos existentes na terminologia e remoção de termos considerados obsoletos pelo padrão.

Após especificadas, as mudanças poderão ser implementadas na ontologia por meio do uso de algum *software* de edição de ontologias. A ferramenta Protégé, apresenta-se como aplicação eficaz para a realização dessa atividade. Antes da edição é recomendada a classificação das mudanças especificadas durante o planejamento da evolução em mudanças simples ou complexas. As mudanças simples são caracterizadas como alterações que necessitam de apenas uma ação para atingir seu objetivo final. Em contrapartida, as mudanças complexas precisam que mais de um ação seja realizada para que o objetivo final da mudança seja alcançado (KLEIN e NOY, 2003). A classificação das mudanças em simples ou complexas possui como finalidade identificar a complexidade de cada mudança e assim, auxiliar os ontologistas durante a definição da prioridade das mudanças a serem executadas na ontologia.

Após a implementação das mudanças na ontologia é importante que a mesma seja analisada para que se identifique a ocorrência de erros relacionados à implementação de mudanças não planejadas, aplicação incorreta de mudanças requeridas ou não aplicação de mudanças que foram planejadas. Essa análise é realizada na fase de Detecção de Mudanças. Tal fase é composta por duas atividades que se complementam: a Comparação entre Versões e a Aceitação ou Rejeição de Mudanças. Para que seja realizada a análise das mudanças com o intuito de aceitar ou rejeitar mudanças aplicadas, deverá ser realizada a comparação entre as versões da ontologia (versão anterior e versão atual/com alterações). A comparação entre as versões pode ser facilitada pelo uso de uma ferramenta *diff*. Um relatório com as mudanças deferidas deverá ser fornecido como saída da fase de Detecção de Mudanças. Esse relatório é subsídio imprescindível para a realização do tratamento das inconsistências ocorridas na ontologia.

Após a implementação das mudanças na ontologia, as regras que definem a sua consistência sofrem alterações decorrentes dessas mudanças e conseqüentemente podem

passar a não satisfazer as condições de consistência da mesma, essas condições de consistência auxiliam na identificação de inconsistências na ontologia, tendo em vista que se uma condição de consistência não for satisfeita, um problema de inconsistência poderá ter ocorrido. A fase de Tratamento de Inconsistências do *framework* está dividida em três atividades denominadas Detecção de Inconsistências, Classificação de Inconsistências e Resolução de Inconsistências.

A detecção das inconsistências ocorridas deverá utilizar o relatório de mudanças deferidas fornecido pela fase de Detecção de Mudanças. Assim, as mudanças realizadas deverão direcionar a busca por problemas de inconsistências na ontologia em evolução. Após identificadas recomenda-se a classificação das inconsistências em lógicas ou estruturais. Inconsistências estruturais ocorrem nos elementos da linguagem utilizada para construir a ontologia em evolução. Em uma ontologia construída em OWL, por exemplo, poderiam ser identificadas inconsistências estruturais caso tenham ocorrido erros em suas classes, propriedades ou instâncias. Inconsistências lógicas estão relacionadas à não possibilidade de realizar inferências sobre as regras da ontologia em evolução devido às inconsistências estruturais, que acometem tais regras. A resolução das inconsistências estruturais reduzem o número de problemas lógicos tendo em vista que, em sua maioria, problemas lógicos são advindos dos estruturais. Assim, a classificação das inconsistências em estruturais ou lógicas permite que os ontologias estabeleçam soluções para os problemas ocorrido do modo menos oneroso possível.

Após classificadas a solução para as inconsistências poderá ser aplicada por meio da edição da ontologia. Tal edição deve ser realizada por meio do uso de algum *software* de edição de ontologias. No *framework* construído foram sugeridas maneiras de resolução de inconsistências. Tais soluções estão diretamente relacionadas aos erros que causam essas inconsistências, que podem ser erros circulatorios, erros de partição ou erros de inconsistência semântica.

Os erros circulatorios podem ocorrer caso uma classe de uma ontologia em evolução seja definida como subclasse de si mesma em qualquer nível da hierarquia dessa ontologia. Os erros de partição dependem do tipo de decomposição de superclasses em subclasses, que pode ser disjunta ou exaustiva. Erros de partição ocasionados pela decomposição disjunta ocorrem quando ontologistas criam instâncias que pertencem a diversas subclasses disjuntas ou quando



definem uma classe como uma subclasse de classes disjuntas. Os erros de partição ocasionados pela decomposição exaustiva ocorrem quando ontologistas fazem uma decomposição ou partição exaustiva de uma classe em muitas subclasses mas nem todas as instâncias da classe base pertencem às suas subclasses. Os erros de inconsistência semântica ocorrem quando ontologistas criam uma hierarquia incorreta de classes por meio da classificação de determinado conceito como subclasse de outro ao qual esse conceito não pertence.

A resolução das inconsistências advindas desses erros está diretamente relacionada ao erro ocorrido. Por exemplo, a correção das inconsistências advindas dos erros circulatorios, deve ser realizada pela análise da hierarquia na qual houve o erro e posterior reorganização dessa hierarquia com o intuito de remover a definição errônea da classe envolvida como subclasse dela mesma. A solução para os erros de partição também estão relacionadas ao erro. Por exemplo, considerando que esses erros podem ser ocasionados por partição disjunta ou exaustiva, para solucionar aqueles relacionados à partição disjunta é necessário desfazer a relação de disjunção entre os conceitos que detêm subconceitos comuns, ou, para o caso da relação de disjunção entre os conceitos não poder ser desfeita, deve-se realizar a análise da hierarquia dos conceitos partindo do conceito raiz, com a finalidade de corrigir as decomposições das superclasses em subclasses e estabelecer a organização hierárquica correta dos conceitos representados de acordo com o domínio do discurso.

Para solucionar erros de inconsistência semântica deverá ser feita uma operação de mover sobre o subconceito que se encontra em uma posição errada na hierarquia segundo o domínio de interesse, com o intuito de colocá-lo no local correto considerando a similaridade do seu significado com a do conceito destino. Outra alternativa viável para a resolução desse problema é a deleção do conceito. Essa alternativa deverá ser implementada apenas caso não seja possível adicionar-se o conceito a uma nova hierarquia de classes. Desse modo, a segunda alternativa de correção será executada como solução residual.

A última fase do *framework* é a Auditoria de Mudanças. Nela deverá ser realizada a análise da conformidade da ontologia com o plano de evolução, ou seja, as mudanças planejadas e a ontologia obtida como resultado da evolução deverão ser analisados com o intuito de verificar a completude e corretude das mudanças requeridas. Após essa análise, caso tenham sido identificadas inconformidades relacionadas às mudanças realizadas e as

planejadas, a ontologia deverá ser encaminhada para a primeira fase do *framework*. Um relatório contendo as considerações do auditor deverá ser encaminhado junto com a ontologia de modo que as mudanças em discondância com o plano de evolução sejam analisadas e refeitas. Em contrapartida, caso a ontologia evoluída esteja em conformidade com o plano de evolução, a mesma deverá ser submetida aos testes onde a sua consistência será verificada. Recomenda-se que nesses testes sejam utilizadas as Questões de Competência da ontologia. Tais questões deverão ser traduzidas em uma linguagem de consulta com expressividade suficiente para representar as QCs da ontologia. Caso não sejam apresentadas respostas para as consultas, as mesmas deverão ser aplicadas na ontologia até que as respostas esperadas para as QCs sejam fornecidas.

Considerando que os problemas lógicos de uma ontologia em evolução estão relacionados às mudanças realizadas em suas regras, a fase de Tratamento de Inconsistências apenas será realizada para ontologias em evolução por mudança de foco ou mudanças em suas diretrizes de domínio. Ontologias em evolução por mudanças relativas à sua terminologia de referência não precisam ser submetidas ao tratamento de inconsistências e conseqüentemente não precisarão ter sua consistência testada na fase de auditoria de mudanças.

#### **5.4 Notas sobre Capítulo**

Algumas atividades do *framework* podem não ser necessárias para os casos de evolução para refletir um número pequeno de mudanças. Por exemplo, a classificação de mudanças em simples ou complexas. Nesse contexto, o *framework* pode ser adaptado à necessidade dos ontologistas de modo que se evite a realização de atividades que, por não serem essenciais para o realidade abordada, podem apenas aumentar o tempo gasto com o processo evolutivo. Os erros para os quais foram fornecidas sugestões de correção na seção de tratamento de inconsistências não são os únicos tipos de erros de podem ocorrer. No *framework* construído nesta dissertação foram abordados apenas os mais comuns com base na literatura utilizada como referência. Assim, é importante que os ontologistas estejam cientes que erros mais complexos e conseqüentemente de resolução mais difícil, podem ocorrer.

As ferramentas utilizadas nos exemplos são apenas sugestões. É interessante que as ferramentas a serem usadas no processo evolutivo sejam escolhidas pelos ontologistas

considerando a realidade da ontologia em evolução em relação ao domínio e linguagem por meio da qual a ontologia foi desenvolvida, por exemplo.

## Capítulo 6

### 6 Estudo de Caso

O *framework* desenvolvido foi validado por meio de um estudo de caso onde o mesmo foi aplicado à ontologia ontoDecideDRC (TAVARES, 2016). Ainda que a mesma ontologia tenha sido previamente utilizada em um estudo de caso (seção 4.1.2) para levantamento de requisitos úteis para a construção do *framework* construído, os resultados esperados pelo atual estudo de caso não poderiam ter sido manipulados tendo em vista que neste estudo foram utilizadas ferramentas de validação distintas das utilizadas anteriormente. No estudo de caso para levantamento de requisitos a validação da ontologia foi realizada apenas com o acompanhamento do especialista de domínio, enquanto que, neste estudo, as Questões de Competência da ontologia foram traduzidas em consultas SPARQL e aplicadas à mesma para verificação da sua consistência. Os processos de verificação e validação bem como de auditoria foram acompanhados pelo especialista do domínio da ontologia.

Além disso, as fases e atividades do *framework* foram definidas considerando as três perspectivas de evolução baseadas nos três motivos que podem levar à evolução de ontologias identificados no estudo de caso para levantamento de requisitos. Esses motivos são a mudança de foco da ontologia, mudanças em suas diretrizes de domínio e mudanças relativas à sua terminologia de referência. Assim, com vistas a demonstrar a aplicabilidade do *framework* diante desses três motivos, duas cópias da ontoDecideDRC foram submetidas a evolução por mudança de foco e mudanças em suas diretrizes de domínio, respectivamente e a versão da ontologia após mudança de foco passou por evolução tendo como motivo a padronização terminológica total dos seus conceitos.

Os materiais e ferramentas utilizadas para a realização do estudo de caso são descritas nas próximas subseções.

#### I. Doença Renal Crônica

A DRC é definida como um conjunto de anormalidades na estrutura ou função dos rins e baseia-se em alterações na Taxa de Filtração Glomerular (TFG) e/ou na presença de lesão

parenquimatosa que se mantém por pelo menos três meses. Considera-se portador de DRC, qualquer indivíduo que, independente da causa, apresente  $TFG < 60 \text{ mL/min/1,73m}^2$  ou a  $TFG > 60 \text{ mL/min/1,73m}^2$  associada a pelo menos um marcador de dano renal parenquimatoso (por exemplo, proteinúria) presente há pelo menos 03 meses (UNA-SUS/UFMA, 2014). A DRC pode ser classificada em 06 estágios de acordo com o grau de redução da TFG conforme pode ser observado na Figura 34. O tratamento da DRC deve ser intensificado de acordo com a evolução da doença.

Figura 34. Prognóstico da DRC por TFG e categorias de albuminúria

**Prognóstico da IRC por TFG e Classificação da Albuminúria: KDIGO 2012**

				Classificação da albuminúria persistente Descrição e variação		
				A1	A2	A3
				Normal a levemente elevado <30 mg/g <3 mg/1mmol	Moderadamente elevado 30-300 mg/g 3-30 mg/1mmol	Severamente elevado >300 mg/g >30 mg/1mmol
Classificação da TFG (mL/min/1,73 m <sup>2</sup> ) Descrição e variação	G1	Normal ou elevado	≥90			
	G2	Levemente reduzido	60-89			
	G3a	Leve a moderadamente reduzido	45-59			
	G3b	Moderado a severamente reduzido	30-44			
	G4	Severamente reduzido	15-29			
	G5	Falência renal	<15			

Verde: baixo risco (se nenhum outro marcador de doença renal, sem IRC); Amarelo: risco moderadamente elevado; Laranja: risco alto; Vermelho: risco muito elevado.

Fonte: KDIGO(2012)

## II. A ontologia ontoDecideDRC

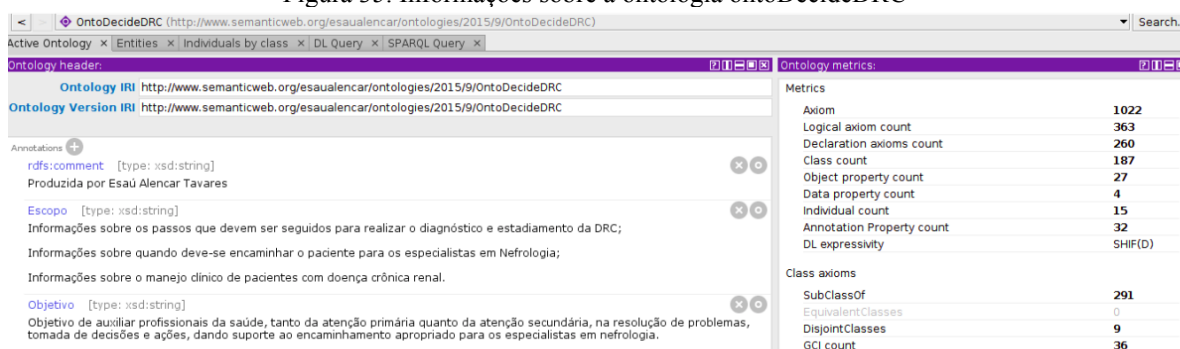
A ontoDecideDRC (TAVARES, 2016) , ontologia submetida a processo de evolução no estudo de caso, foi formulada para implementar conceitos relacionados aos processos necessários ao diagnóstico, estadiamento, encaminhamento e tratamento de pacientes com DRC e para ser construída passou por etapas de conceitualização, formalização e implementação. Na conceitualização foi feita a classificação e descrição dos dados através da elaboração do Léxico Ampliado da Linguagem (LAL), linguagem de representação utilizada para organizar o vocabulário sobre o domínio em estudo.

Durante a etapa de formalização foi feito o mapeamento dos termos conforme as primitivas básicas de construção de ontologias, ou seja, os termos do domínio a ser representado na ontologia foram convertidos em classes, propriedades, hierarquias, restrições ou axiomas de acordo com suas características e funções (TAVARES, 2016). Na implementação, a ontologia foi desenvolvida utilizando-se para isso a *Web ontology Language-Description Logic* (OWL-DL). Nessa fase, foi utilizada a ferramenta Protégé.

Constitui-se objetivo da ontoDecideDRC, auxiliar os médicos da atenção primária e Nefrologistas na resolução de questões relativas à tomada de decisão fornecendo informações sobre os procedimentos de diagnóstico, estadiamento da DRC e o momento correto para o encaminhamento de pacientes para serem assistidos pelos profissionais da Nefrologia (TAVARES, 2016).

Como resultados da construção da ontologia ontoDecideDRC tem-se 184 classes com propriedades de objeto, dados e anotação. Conforme informado nas métricas da ontologia fornecidas pela ferramenta Protégé, a mesma possui 1240 axiomas, dos quais 389 são axiomas lógicos. A ferramenta ainda classificou a expressividade da lógica de descrição da ontologia como “*SHIF(D)*” pois a mesma “possui intersecção de conceitos, restrições universais, classes disjuntas, conceitos complexos de negação, propriedades inversas, propriedades funcionais, propriedades de dados, entre outras características” (TAVARES, 2016). As métricas e demais informações relativas à ontologia especificadas pelo *Protégé* podem ser observadas na Figura 35.

Figura 35. Informações sobre a ontologia ontoDecideDRC



Fonte: Protégé

A ontoDecideDRC possui 13 classes principais. A classe “Paciente”, descreve as informações básicas por exemplo o gênero, cor da pele, idade e peso do paciente ou possível

paciente com DRC. A classe “DRC” apresenta conceitos básicos da DRC como fatores de risco e profissionais da saúde responsáveis por tratar o paciente dependendo do estágio da DRC em que este se encontra. A classe “Encaminhamento” representa as informações sobre os momentos e condições nas quais o paciente deve ser encaminhado para o médico nefrologista. Em “Medico” são apresentados os tipos de médicos que realizarão o acompanhamento do paciente: Médicos da atenção primária e Nefrologista. Na classe “Formulas” são representadas as fórmulas que os profissionais que acompanham os pacientes com DRC devem utilizar para calcular a TFG. Os estágios da DRC são subclasses da classe “ClassificacaoDaDRC”, as quais foram nomeadas do seguinte modo: “Estagio1”, “Estagio2”, “Estagio3A”, “Estagio3B”, “Estagio4”, “Estagio5ND” e “Estagio5D” (TAVARES, 2016).

A classe “Estrategias”, descreve os meios que podem ser aplicados para o tratamento e prevenção da DRC, possuindo como subclasses as classes “ManejoClinicoBasico” e “TerapiaDeSubstituicaoRenal”. A primeira detém conceitos sobre os procedimentos clínicos básicos que devem ser realizados para retardar a progressão da DRC. A segunda representa os procedimentos a serem realizados no caso de o paciente ser diagnosticado com DRC. “Exames” é a classe que apresenta os exames exigidos para apoio à decisão clínica durante o tratamento da DRC. “GrauDeRiscoDaDRC” foi criada para representar as probabilidades de o paciente possuir DRC. A classe “FatoresDeRisco” por sua vez traz os fatores que podem intensificar as chances de diagnóstico da DRC em determinado paciente. “Vacinao” representa as vacinas que o paciente deveria ter tomado ou terá que tomar caso não o tenha feito. A classe “Medicamentos” traz informações quanto aos medicamentos a serem utilizados pelo paciente bem como sua administração (TAVARES, 2016).

A estrutura conceitual da ontologia foi estabelecida por meio da definição de propriedades. As conexões entre os conceitos definidos em suas classes foram estabelecidas por 26 propriedades de objeto. Os valores numéricos foram definidos em 3 propriedades de dados e as informações detalhadas sobre as classes foram definidas em 4 propriedades de anotação que foram utilizadas além das propriedades pré-definidas pela linguagem. (TAVARES, 2016).

Após a definição das classes e propriedades da ontologia, a fim de complementar a modelagem semântica do domínio, foram desenvolvidas regras semânticas baseadas nos

conceitos da ontologia de forma a realizar-se inferências que permitam concluir um determinado fato. Tais inferências foram criadas no formato [Conjunto de Condições] -> [Ação] (TAVARES, 2016). Tavares (2016) avaliou que a ontologia *ontoDecideDRC* contribuiu para a captura e estruturação das informações relacionadas a práticas clínicas voltadas para o tratamento da DRC.

### III. SNOMED CT

Lançada em 2002, a terminologia de referência clínica SNOMED CT foi originalmente desenvolvida pelo Colégio de Patologistas Americanos. A terminologia é resultado da união do sistema SNOMED *Reference Terminology* (SNOMED RT) com o sistema *Clinical Terms Version 3* (CTV3). Em 2007 o direito de propriedade intelectual da SNOMED CT foi adquirido pela IHTSDO que passou a ser responsável pela sua manutenção de distribuição (AIL-HABLANI, 2017). SNOMED CT é a mais abrangente terminologia clínica internacional e multilíngue do mundo (APRENDIS, 2016). Em seu último *release* (junho de 2018) a terminologia contabilizou 340.659 conceitos ativos (IHTSDO, 2018).

Países como Estados Unidos, Reino Unido, Canadá e Nova Zelândia designaram SNOMED CT como a terminologia de referência clínica recomendada para uso em Sistemas de Informação Clínica (CIS, do inglês, *Clinical Information Systems*). De acordo com a IHTSDO, SNOMED CT pode ser implementada para três finalidades: Registros Clínicos, Representação do Conhecimento e Agregação e Análise. A implementação da terminologia em Registros Clínicos se refere ao manuseio de dados de pacientes bem como ao suporte para a realização de serviços que incluem o armazenamento, recuperação e comunicação de dados em CIS baseados em SNOMED CT. A Representação do Conhecimento envolve a expressão de conhecimentos clínicos, tais como diretrizes clínicas e formas de tratamento por meio do uso da terminologia SNOMED CT. A implementação do padrão para Agregação e Análise diz respeito à recuperação de dados de um CIS com a finalidade de se realizar análises secundárias (LEE et al., 2013).

Padrão aberto global, SNOMED CT possui conteúdo clínico abrangente e cientificamente validado, fornece recursos para a representação consistente e processável do Registro Eletrônico de Saúde (RES) bem como possui uma estrutura que permite a sua utilização por outras normas clínicas internacionais. SNOMED CT facilita ainda a integração



de dados clínicos de múltiplas fontes que usam diferentes sistemas de código ou texto livre e possibilita a recuperação de informações baseadas em significado de fatos clínicos relevantes (SNOMED CT, 2018).

Com o uso da SNOMED CT é possível realizar-se a documentação e raciocínio com a informação clínica, armazenamento, recuperação, análise, compartilhamento, suporte à decisão e agregação de conceitos em diferentes níveis. A terminologia possui extensibilidade para atender requisitos locais e especializados. Em SNOMED CT os termos estão organizados em dois repositórios ou *browsers* internacionais: um cujos termos estão em inglês e outro onde os mesmos são representados em espanhol. Existem ainda as extensões locais, que são *browsers* onde são seguidas especificações clínicas locais e os conceitos estão representados na linguagem específica do país, que é chamado de país membro da terminologia<sup>29</sup> (SNOMED CT, 2018).

A utilização da SNOMED CT em aplicações de *software* pode ser feita por meio da representação de conteúdos clínicos relevantes de forma consistente, confiável e compreensiva como parte integrante da produção de informações eletrônicas de saúde. SNOMED CT é rica semanticamente e possui um vocabulário controlado que facilita o crescimento evolutivo da expressividade para dar suporte a requisitos em processo de evolução. SNOMED CT é mapeada para outros padrões internacionais e utilizada em mais de 50 países, dentre os quais está o Brasil, que passou a ser membro da terminologia em abril de 2018 (SNOMED CT, 2018).

O objetivo do mapeamento da SNOMED CT para outros padrões é estabelecer a relação dos mesmos com uma terminologia internacional com sistema de classificação e código que forneça benefícios como: reúso de dados, retenção de valores de dados quando estes forem migrados para formatos e esquemas de base dados mais atuais, evitar a entrada de dados várias vezes e o risco associado ao aumento de custos e erros e fornecer interoperabilidade entre terminologias internacionais (SNOMED CT, 2018).

Informações clínicas baseadas em SNOMED CT fornecem vantagens para profissionais da saúde, pacientes e para a população em geral ao mesmo tempo que apoiam o cuidado em saúde baseando-se em evidências. O uso do RES possibilita a comunicação e

<sup>29</sup><https://browser.ihstsdotools.org/>

aumenta a disponibilidade de informações clínicas relevantes. Quando o RES utiliza um meio de armazenamento de informações que permita a sua recuperação baseado-se em seu significado os benefícios são consideravelmente maiores (SNOMED CT, 2018).

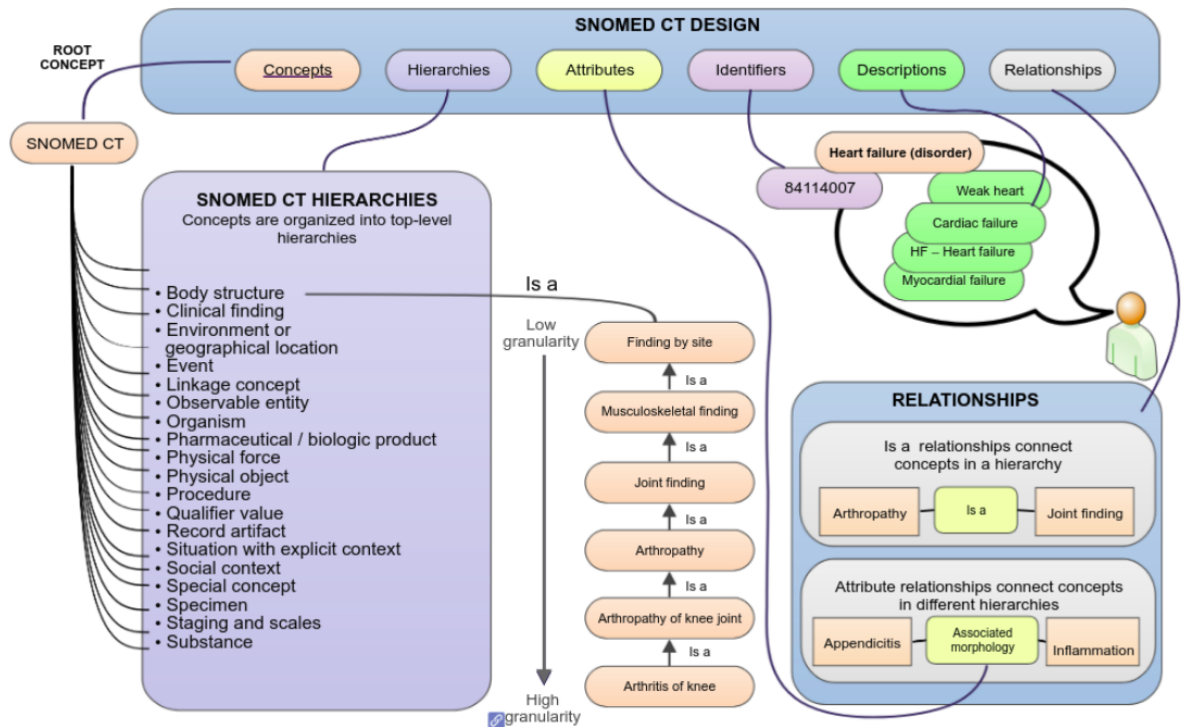
Para os pacientes, os benefícios do uso RES baseado em SNOMED CT incluem o registro consistente de informações clínicas, sistemas que ofereçam aconselhamento em tempo real, apoio ao compartilhamento de informações apropriadas com os envolvidos na prestação dos cuidados, o que permite a compreensão da informação de modo comum; análise precisa e abrangente que permite identificar pacientes que necessitam de acompanhamento ou mudanças de tratamento e remoção das dificuldades e limitações apresentadas pelas diferentes linguagens tendo em vista que SNOMED CT é uma terminologia clínica multilíngue (SNOMED CT, 2018).

Compreendem algumas das vantagens do uso RES baseado em SNOMED CT para as populações em geral, a disponibilização de dados relevantes para apoiar a pesquisa clínica e contribuir com evidências para melhorias dos tratamentos, acurácia e acesso a informações relevantes reduzindo custos com informações duplicadas e erros decorrentes desse problema, facilidade na identificação prévia de problemas de saúde emergentes, monitoramento da saúde da população e respostas para práticas clínicas em processo de mudança. O RES fundamentado na SNOMED CT também fornece apoio ao cuidado em saúde baseado em evidências apresentando como vantagem a possibilidade de estabelecer ligações entre registros clínicos e diretrizes e protocolos clínicos melhorados, a melhoria da qualidade do cuidado e redução dos custos com testes e tratamentos inapropriados e duplicados, limitando a ocorrência frequente de eventos adversos no cuidado em saúde (SNOMED CT, 2018).

Tecnicamente, SNOMED CT possui como característica a poli hierarquia, característica por meio da qual cada conceito pode estar relacionado a vários conceitos pais, que por sua vez representam um conceito mais geral; definições formais, categorias não residuais, granularidade múltipla, visões múltiplas consistentes por meio de diagramas, conceitos e sinônimos, representação de contexto, por meio de históricos possíveis que apoiam o processo de diagnóstico; evolução incremental, onde evita-se a perda das versões anteriores e reconhecimento das redundâncias por meio da capacidade da terminologia de reconhecer duplicações nas informações representadas (SNOMED CT, 2018).

O projeto da terminologia SNOMED CT pode ser observado na Figura 36. Conforme a Figura, SNOMED CT está estruturada em conceitos, hierarquias, atributos, identificadores, descrições e relacionamentos.

Figura 36. Projeto da terminologia SNOMED CT

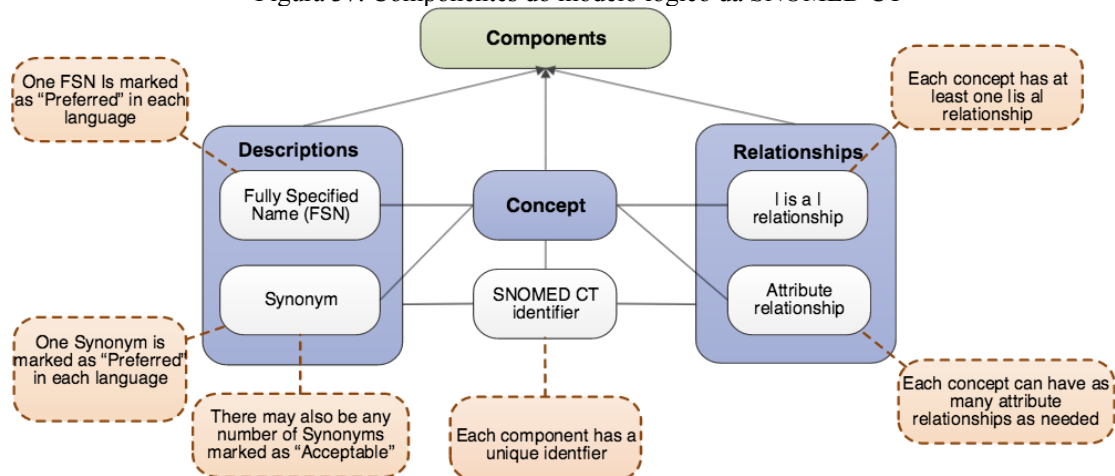


Fonte: SNOMED CT Starter Guide

Conforme a Figura 37, conceitos são os elementos principais da terminologia e os relacionamentos e descrições estão diretamente relacionados a ele. Um conceito possui um identificador único chamado de SNOMED CT *identifier* (SCTID), conceitos podem ter tantos relacionamentos quantos forem necessários, porém cada conceito tem que possuir no mínimo um relacionamento do tipo “Is a”.

Conceitos em SNOMED CT são organizados em hierarquias. Em uma hierarquia os conceitos variam de acordo com o seu nível de detalhamento ou granularidade, em SNOMED CT a granularidade é estabelecida por termo, os quais de acordo com o seu nível na hierarquia podem ser gerais (baixa granularidade) ou detalhados (alta granularidade). A relação de conceitos em uma hierarquia é estabelecida por meio da ligação entre os mesmos utilizando-se o tipo de relacionamento “Is a”.

Figura 37. Componentes do modelo lógico da SNOMED CT



Fonte: SNOMED CT – 5 Step Briefing

Ao todo, SNOMED CT possui 20 hierarquias de nível superior para organização estrutural de conceitos. “*Clinical finding*”, “*procedure*”, “*observable entity*”, “*body structure*” e “*organism*” são exemplos desses conceitos. Na Tabela 14, pode ser observada uma breve descrição sobre o conteúdo representado por eles.

Tabela 14. Exemplos de conceitos de nível superior em SNOMED CT

Conceito de nível superior	Descrição
<i>clinical finding</i>	Representa o resultado de uma observação, avaliação ou opinião clínica, inclui estados clínicos normais ou anormais e conceitos usados para representar diagnósticos. Por exemplo: dor de cabeça, respiração normal etc.
<i>procedure</i>	Representa as atividades realizadas para prover o cuidado em saúde. Inclui procedimentos invasivos, administração de medicamentos, educação, terapias, entre outros. Por exemplo: apendicectomia, fisioterapia etc.
<i>observable entity</i>	Representa um questionamento ou avaliação que pode ajudar na produção de um resultado. Por exemplo: pressão arterial sistólica, cor da íris etc.
<i>body structure</i>	Representa estruturas anatômicas normais ou anormais. Por exemplo: estrutura da valva mitral, adenossarcoma etc.
<i>organisms</i>	Representa organismos significantes para medicina humana ou animal. Por exemplo: streptococcus pyogenes, beagle etc.

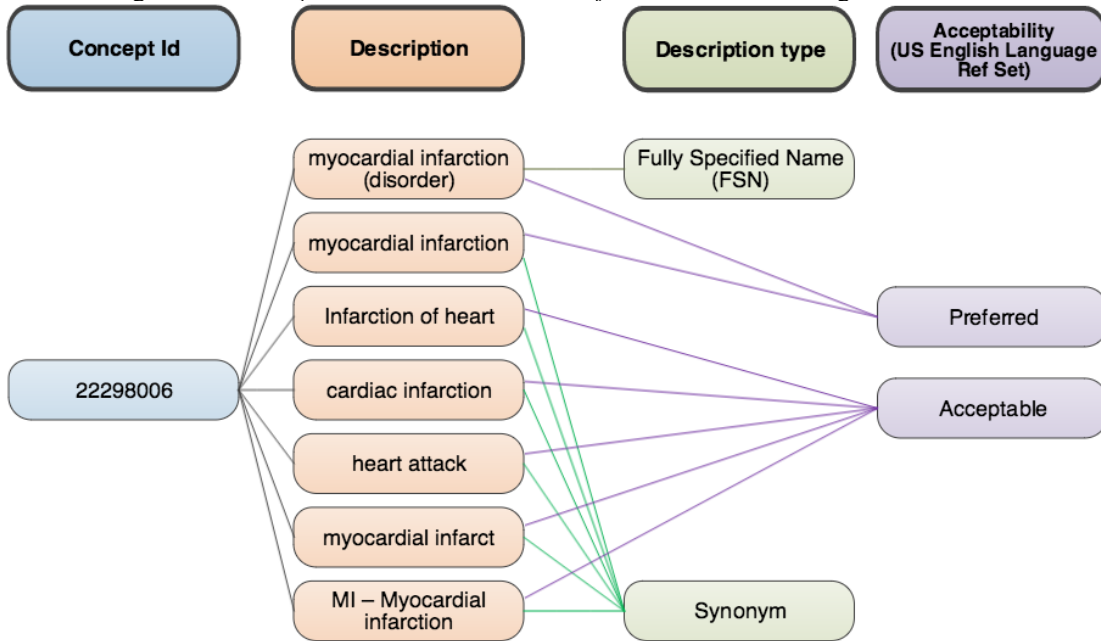
Fonte: SNOMED CT – Starter guide

Quanto às descrições, conforme a imagem, as mesmas podem ser classificadas em Nome Totalmente Especificado (FSN, do inglês *Fully Specified Name*) ou Sinônimos. O FSN representa a descrição única e não ambígua do significado de um conceito. Essa representação única é particularmente útil quando ocorre a referência de diferentes conceitos pela mesma palavra ou frase. Cada conceito pode ter apenas um FSN em cada linguagem ou dialeto. Um sinônimo representa um termo que pode ser usado para mostrar ou selecionar um conceito. Um conceito pode ter vários sinônimos (SNOMED CT, 2018).

O FSN e os sinônimos podem ser qualificados de acordo com a sua aceitabilidade. Em SNOMED CT a aceitabilidade de um termo se refere a o quão recomendado é seu uso. Essa aceitabilidade pode ser classificada como “*Preferred*” (Preferível) ou “*Acceptable*” (Aceitável). A marcação do FSN como variação preferível do termo, significa que o mesmo possui aceitabilidade preferida pela comunidade de profissionais da saúde do domínio no qual esse termo é utilizado. O FSN é portanto a variação do termo clínico mais recomendada. Os sinônimos são descrições alternativas dos termos. Pelo fato de um termo poder ter vários sinônimos, a classificação de aceitabilidade também é utilizada para classificar os sinônimos. Assim, em SNOMED CT um único sinônimo deve ser escolhido para possuir aceitabilidade “*Preferred*” e não existe limite em relação à quantidade de sinônimos com aceitabilidade “*Acceptable*” (SNOMED CT, 2018).

Na Figura 38, pode ser observado um exemplo onde as possíveis descrições para o termo “*Myocardial infarction*” têm sua aceitabilidade classificada. A descrição “*Myocardial infarction (disorder)*” é o FSN do termo. “*Miocardial infarction*” é o sinônimo cuja aceitabilidade foi definida como “*Preferred*”. Os demais sinônimos foram classificados como “*Acceptable*”.

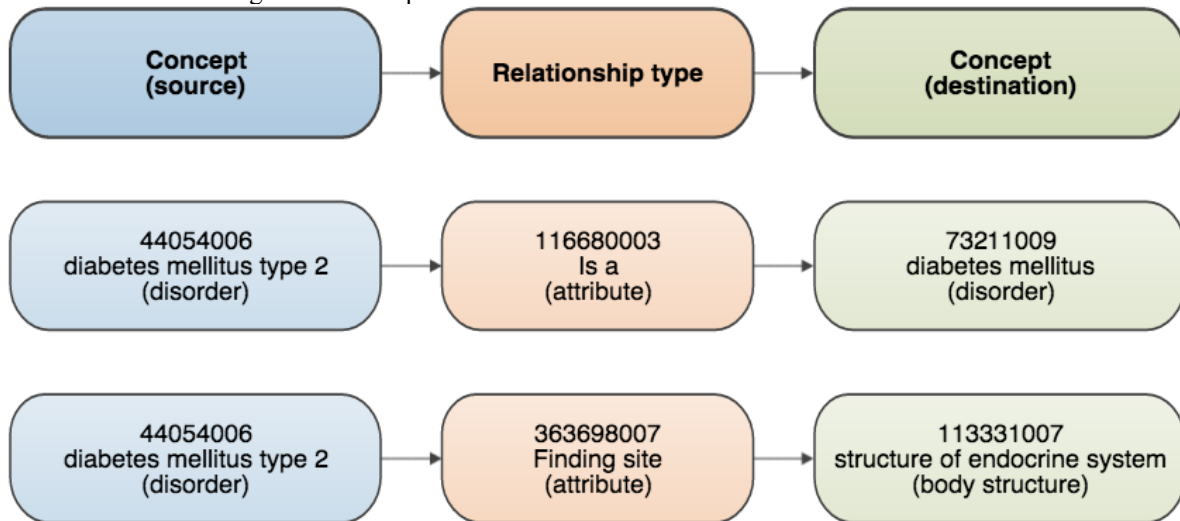
Figura 38. Descrição do termo "Myocardial Infarction" na terminologia SNOMED CT



Fonte: SNOMED CT – 5 Step Briefing

Os relacionamentos em SNOMED CT representam a associação entre dois conceitos. Eles são usados para definir de forma lógica o significado de um conceito de modo que este possa ser processado por computador. Existem os tipos ou atributos de relacionamentos. Um tipo de relacionamento é usado para representar o significado da associação entre o conceito origem e o conceito destino. “*Is a*”, “*Proper part of*” (inserido no *release* de junho de 2018) e “*Finding site*” são exemplos de atributos de relacionamentos existentes em SNOMED CT. Na Figura 39 podem ser observados dois exemplos de relacionamentos entre dois conceitos em SNOMED CT.

Figura 39. Exemplos de atributos de relacionamentos em SNOMED CT



Fonte: SNOMED CT – Starter Guide

No primeiro exemplo o uso do atributo “*Is a*” entre os conceitos significa que o conceito “*diabetes mellitus type 2 (disorder)*” (conceito origem) é um subconceito de “*diabetes mellitus (disorder)*” (conceito destino). No segundo exemplo o atributo “*Finding site*” é o tipo de relacionamento usado. Levando em conta o significado desse atributo, o relacionamento entre os dois conceitos do exemplo permite dizer que “*diabetes mellitus tipo 2 (disorder)*” (conceito origem) é uma doença que ocorre na estrutura do sistema endócrino que é a tradução do termo usado no exemplo da figura para o conceito “*structure of endocrine system (body structure)*” (conceito destino).

SNOMED CT possui expressões pré e pós-coordenados. As expressões pré-coordenadas são os termos já existentes na terminologia, enquanto as pós-coordenadas originam-se da combinação desses termos existentes. Eles são particularmente úteis quando é necessário o uso de um termo padronizado, porém o mesmo não existe na terminologia. Assim, um novo termo pode ser modelado utilizando-se para isso termos existentes no repositório da SNOMED CT. O SNOMED CT *identifier* das expressões pós-coordenadas é mais longo, pois contém informações acerca de quem o criou, por exemplo(SNOMED CT, 2018).

Conforme Richesson, Andrews e Krischer (2006) o uso da pós-coordenação é simples nas áreas onde o modelo de contexto clínico da SNOMED CT é completo e intuitivo.

Entretanto, nos casos em que propõe-se a criação de expressões correspondentes a qualificadores de conceito dependentes do contexto, tais como negação e sujeito de observação, o uso de pós-coordenação torna-se um processo complexo.

SNOMED CT pode ser utilizada como dicionário para análise de dados clínicos no RES e em *Data Warehouses* (DW). Nesse último caso, o uso da SNOMED CT como dicionário para análise de dados oferece como benefício a melhoria da qualidade dos dados por meio da sua clara representação e mapeamento dos seus códigos para o modelo SNOMED CT de codificação de conceitos além da possibilidade de reestruturação dos dados para otimizar a performance das consultas. É possível ainda fazer uso da terminologia ligando-a com bases de conhecimento e como base para documentação de informações clínicas. Para que se tenha benefícios decorrentes do uso da terminologia SNOMED CT é necessário que a mesma seja implementada como parte de uma aplicação. As características do *software* que usará o padrão bem como seus objetivos e justificativas são fatores determinantes para o sucesso do uso do mesmo (SNOMED CT, 2018).

#### IV. Ferramentas *Diff*

Após a aplicação das mudanças requeridas, as entidades da ontologia podem sofrer alterações induzidas por essas mudanças e passar a não satisfazer uma ou mais condições de consistência às quais estão vinculadas. Com a finalidade de apoiar a detecção dessas mudanças existem as ferramentas *diff*, que permitem a visualização das diferenças entre duas versões de uma ontologia. São exemplos dessas ferramentas a *PromptDiff* criada por Noy e Musen (2002), *Conto-Diff* criada por Hartung, Groß e Rahm (2013) e *OWL Differences*, criada como um *plug-in* para o Protégé<sup>30</sup>. Existem outros *plug-ins* do Protégé para visualização de mudanças: *Changes Tab*, *Version Log Generator* e *Changes Tracer*, avaliados por Khattak, Latif e Lee (2013) como boas ferramentas para esse fim. *ChangeDetection* é uma ferramenta stand alone para visualização de mudanças que também pode ser usada, porém como a captura de mudanças é a única funcionalidade da ferramenta, será necessário o uso de outras alternativas para, por exemplo, aplicar as correções das inconsistências identificadas, o que acarreta uso de excessivas ferramentas podendo causar aumento dos custos com tempo e

<sup>30</sup><https://protege.stanford.edu/>



recursos humanos durante o processo de evolução. *NeOn Toolkit*<sup>31</sup> é outra alternativa para edição de ontologias que possui um *plug-in* para visualização de mudanças, o *ChangeCapturing*<sup>32</sup>.

A visualização de todas as mudanças ocorridas entre as versões da ontologia em processo de evolução também facilita a identificação das inconsistências que venham a ocorrer na ontologia pois permite que os responsáveis pela evolução possam ter acesso às entidades afetadas, o que reduz o tempo gasto com o processo tendo em vista que a detecção de mudanças não precisará ser uma realizada manualmente.

Por ser uma ferramenta intuitiva e adaptável às necessidades do usuário, o Protégé foi utilizado em todas as atividades dos processos do *framework* que necessitaram do uso de um *software* para edição de ontologias durante o estudo de caso. Dessa forma, a ferramenta *diff* utilizada para apoiar a identificação de mudanças necessárias na fase de detecção de inconsistências foi o *plug-in* OWL *Differences*.

## **6.1 Uso do *framework* em uma ontologia em evolução por mudança de foco**

Nessa perspectiva de evolução, em vez de representar conceitos relacionados ao diagnóstico, estadiamento, encaminhamento e tratamento dos pacientes em todos os estágios da DRC, a *ontoDecideDRC* passou a representar conceitos até o encaminhamento do paciente para a atenção especializada, que deve ocorrer quando este estiver no estágio 3B da doença. O novo foco da ontologia foi sugerido pela médica especialista em Nefrologia que acompanhou o estudo de caso. Tal sugestão teve como base a experiência da especialista em presenciar a grande ocorrência de diagnósticos imprecisos causados por informações incorretas e inconsistentes no formulário de encaminhamento do Médico da Atenção Primária (MAP) para a atenção especializada no serviço público de saúde. A mudança de foco da ontologia ocasionou a sua mudança de nome. Assim a mesma passou a ser chamada de *ONTODRC*.

Seguindo os passos do *framework* para esta perspectiva de evolução, após a identificação do novo foco da ontologia, foram estudadas suas novas fontes de conhecimento. A partir desse estudo, as mudanças a serem consideradas em decorrência da mudança de foco

31<http://neon-toolkit.org/>

32[http://neon-toolkit.org/wiki/1.x/Change\\_Capturing.html](http://neon-toolkit.org/wiki/1.x/Change_Capturing.html)

da ontologia foram capturadas e especificadas conforme a atividade de representação das mudanças integrante da fase de Planejamento da Evolução do *framework*. Para adaptar a classe da ontologia que faz o estadiamento da DRC à diretriz internacional, criada e mantida pela KDIGO, foi incluído na metodologia deste trabalho, o documento de Avaliação e Gerenciamento da DRC (do inglês, CKD *Evaluation and Management*).

Devido à mudança de foco da ontoDecideDRC, além do uso do documento de Diretrizes Clínicas para o cuidado do paciente com DRC no SUS, que já havia sido utilizado para a criação da versão inicial da ontoDecideDRC, foi necessária a inclusão do documento de Protocolos de Encaminhamento da Atenção Básica para a Atenção Especializada - Endocrinologia e Nefrologia do Ministério da Saúde<sup>33</sup> como nova fonte de conhecimento da ontologia. As mudanças identificadas e especificadas foram dispostas em um relatório de mudanças conforme pode ser observado na Tabela 15.

Após especificadas, as mudanças foram aplicadas na ontologia utilizando a ferramenta Protégé. Após serem aplicadas as mudanças especificadas, a ontologia foi encaminhada para a fase de detecção das mudanças aplicadas, tal fase é importante pois detecta as mudanças induzidas, que são as mudanças que ocorrem como consequência das mudanças requeridas.

Ao fim da aplicação das mudanças, é natural que a ontologia passe a não fornecer respostas para questões relativas ao seu domínio, como por exemplo: “quais estágios da DRC devem ser tratados pelo Nefrologista?”. Assim, a versão anterior (sem alterações) e a versão atual (após mudanças) da ontologia foram comparadas de modo que, por meio da identificação de mudanças, pudessem ser encontradas as inconsistências causadoras dos problemas da ontologia. Para esta finalidade foi utilizada uma ferramenta *diff*.

Tabela 15. Lista de mudanças especificadas para aplicação na versão da ontologia ontoDecideDRC em evolução por mudança de foco

<b>Ontologia em Evolução:</b> ontoDecideDRC		
<b>Motivo da Evolução:</b> Mudança do foco da ontologia		
<b>Descrição da Entidade</b>	<b>Mudança a ser aplicada</b>	<b>Descrição da Mudança</b>
Nefrologista	Remoção	Remover conceito e todas as suas referências

<sup>33</sup>[http://dab.saude.gov.br/portaldab/biblioteca.php?conteudo=publicacoes/protocolos\\_atencao\\_basica\\_atencao\\_especializada](http://dab.saude.gov.br/portaldab/biblioteca.php?conteudo=publicacoes/protocolos_atencao_basica_atencao_especializada)

Estagio3B	Remoção	Remover conceito e todas as suas referências
Estagio4	Remoção	Remover conceito e todas as suas referências
Estagio5	Remoção	Remover conceito e todas as suas referências
Estagio1	Alteração	Alterar a descrição da classe para DoencaRenalCronicaEstagio1
Estagio2	Alteração	Alterar a descrição da classe para DoencaRenalCronicaEstagio2
Estagio3A	Alteração	Alterar a descrição da classe para DoencaRenalCronicaEstagio3A
Conduta	Adição	Adicionar na hierarquia de classes como conceito raiz
CondutaEstagio1	Adição	Adicionar na hierarquia como conceito filho do conceito “Conduta”
CondutaEstagio2	Adição	Adicionar na hierarquia como conceito filho do conceito “Conduta”
MotivosDeEncaminhamentoMaisComuns	Adição	Adicionar na hierarquia de classes como conceito raiz
DiabetesMellitus	Adição	Adicionar na hierarquia como conceito filho do conceito “MotivosDeEncaminhamentoMaisComuns”
EObtidoPor	Adição	Adicionar como propriedade da ontologia
Utiliza	Adição	Adicionar como propriedade da ontologia
EUtilizadoPor	Adição	Adicionar como propriedade da ontologia

Fonte: A autoria própria

O *plug-in* OWL *Differences*, disponível para uso na ferramenta Protegé foi adicionado ao mesmo e utilizado no processo de comparação. Ao comparar as versões foi fornecida uma lista de diferenças com todas as mudanças especificadas e induzidas por estas, ou seja, para cada mudança também foi fornecida uma lista de axiomas afetados. Na Tabela 16 podem observadas algumas das mudanças detectadas pela ferramenta *diff* bem como os axiomas afetados por elas.

Tabela 16. Diferenças entre a versão anterior e a versão alterada da ontoDecideDRC em evolução por mudança de foco

<b>Ontologia em Evolução:</b> ontoDecideDRC		
<b>Motivo da Evolução:</b> Mudança de foco da ontologia		
<b>Descrição da Entidade</b>	<b>Mudança Aplicada</b>	<b>Axiomas Afetados</b>

Nefrologista	Remoção	-Classe Nefrologista -Nefrologista <i>SubclassOf</i> Medico -MedicoDaAtencaoPrimaria <i>DisjointWith</i> Nefrologista -Nefrologista <i>sinonimo</i> MedicoNefrologista -Nefrologista <i>isDefinedBy</i> “Nefrologia é uma especialidade do âmbito da medicina interna...” ^^string
Estagio3B	Remoção	-Classe Estagio3B -Estagio3B <i>SubclassOf</i> ClassificacaoDaDRC <i>DisjointClasses</i> Estagio1, Estagio2, Estagio3A, Estagio4, Estagio5D, Estagio5ND, -Estagio3B <i>guideline</i> “Manter os valores de fósforo normais”
Estagio1	Remoção	-Classe Estagio1 -Estagio1 <i>SubclassOf</i> ClassificacaoDaDRC -Estagio1 <i>SubclassOf</i> EClassificadoPor some T1 -Estagio1 <i>SubclassOf</i> ETratadoPor only MedicoDaAtencaoPrimaria -Estagio1 <i>definição</i> “Paciente com TFG maior ou igual a 90mL/min/1,73m <sup>2</sup> ...” ^^string
DoencaRenalCronicaEstagio1	Adição	-Classe DoencaRenalCronicaEstagio1 -DoencaRenalCronicaEstagio1 <i>SubclassOf</i> Estadiamento -DoencaRenalCronicaEstagio1 <i>SubclassOf</i> EObtidoPor only TFG1 - DoencaRenalCronicaEstagio1 <i>isDefinedBy</i> “TFG maior ou igual a 90mL/min/1,73m <sup>2</sup> na presença de proteinúria ou hematúria glomerular...” @pt
DiabetesMellitus	Adição	-Classe DiabetesMellitus -DiabetesMellitus <i>SubclassOf</i> MotivosDeEncaminhamentoMaisComuns -DiabetesMellitus <i>condicoesClinicasParaEncaminhamento</i> “Pacientes com TFG<30mL/min/1,73m <sup>2</sup> ...” ^^string
Conduta	Adição	-Classe Conduta -Conduta <i>isDefinedBy</i> “Formas de manejo clínico do paciente com DRC conforme o estadiamento da doença” ^^string -Conduta range string
EObtidoPor	Adição	<i>ObjectPropert:</i> EObtidoPor

Fonte: Autoria própria

No *log* de diferenças fornecido pela ferramenta *diff*, as classes alteradas por meio de renomeação aparecem com a descrição anterior como classes removidas e com a nova descrição como classes adicionadas, por esse motivo, a alteração prevista na linha 5 da Tabela

15 foi apresentada na tabela 16 como uma remoção seguida de adição. Tais operações podem ser observadas na linha 3 e 4, respectivamente.

Como saída da fase onde foi realizada a detecção das mudanças aplicadas por meio da comparação entre as versões da ontoDecideDRC, foi fornecida a ontologia alterada para que pudesse ser submetida ao subprocesso onde, conforme o *framework*, deve ser realizado o tratamento das inconsistências ocorridas. Tais inconsistências foram identificadas com base dos axiomas afetados pelas mudanças aplicadas em cada entidade. Desse modo, tendo a tabela de mudanças aplicadas como direcionadora do processo de identificação de inconsistências, as mesmas foram detectadas por meio da busca, na ontologia, do local onde se encontram os axiomas afetados.

Na Tabela 17, podem ser observadas algumas das inconsistências detectadas, na mesma, tais inconsistências já se encontram classificadas como estruturais ou lógicas conforme a localização do impacto ocorrido na estrutura ontológica.

Com base no relatório de inconsistências identificadas, foram aplicadas as devidas correções na ontologia utilizando a ferramenta Protégé. Os erros mais identificados foram os de partição ocasionados por decomposição disjunta, tendo em vista que algumas instâncias foram definidas como instâncias comuns para classes disjuntas. Por exemplo, a instância A1 estava sendo compartilhada por todos os estágios da DRC cujas diretrizes do domínio define como conceitos distintos.

Tabela 17. Inconsistências identificadas na versão da ontoDecideDRC em evolução por mudança de foco

<b>Ontologia em Evolução:</b> ontoDecideDRC			
<b>Motivo da Evolução:</b> Mudança do foco da ontologia			
<b>Entidade Afetada</b>	<b>Mudança aplicada</b>	<b>Inconsistências</b>	
		<b>Estruturais</b>	<b>Lógicas</b>
Nefrologista	Remoção	Erros nos axiomas relacionados à conduta dos pacientes	Impossível determinar quem trata o paciente com DRC a partir do estágio 3B
Estagio1	Alteração	-Impossível encontrar a classe Estagio1 <i>SubclassOf</i> ClassificacaoDaDRC -Impossível encontrar a classe Estagio1 <i>SubclassOf</i> EClassificadoPor some T1	-Impossível inferir que a classe Estagio1 é subclasse da classe ClassificacaoDaDRC -Impossível inferir que a classe Estagio1 é classificada por meio do nível 1 da TFG

		-Impossível encontrar a classe Estagio1 <i>SubclassOf ETratadoPor only MedicoDaAtencaoPrimaria</i>	-Impossível inferir que pacientes no estágio 1 da DRC devem ser tratados pelo Médico da Atenção Primária de Saúde
Estagio3B	Remoção	-Impossível encontrar a classe Estagio3B <i>SubclassOf ClassificacaoDaDRC</i> -Impossível encontrar a classe que possui a descrição “Manter os valores de fósforo normais” para a <i>annotation “guideline”</i>	-Impossível inferir que a classe Estagio3B é subclasse da classe ClassificacaoDaDRC -Impossível determinar a <i>guideline</i> a ser observada para pacientes no Estagio3B da DRC

---

Fonte: Autoria própria

Tendo em vista que, conforme as regras de classificação da doença, as categorias dos níveis de albuminúria (conceitos que detêm a instância A1) devem ser consideradas para determinar todos os estágios da doença, as mesmas passaram a não ser representadas como indivíduos e foram redefinidas como sendo subclasses da classe “CategoriasDosNiveisDeAlbuminuria”. Assim, os axiomas que determinam o estadiamento da DRC poderão fazer referência a esse conceito sem prejuízos à consistência da ontologia e sem que sejam causados problemas de ambiguidade.

Após a correção das inconsistências identificadas, considerando que a evolução foi motivada pela mudança de foco da ontologia, suas Questões de Competência foram redefinidas. Essa tarefa foi realizada pelo especialista do domínio da Nefrologia que acompanhou o processo de evolução da ontologia. As novas QCs da ontologia podem ser observadas na Tabela 18. A partir dessas QCs foram construídas as consultas SPARQL (Tabela 19) utilizadas para a execução dos testes de consistência da ontologia.

Tabela 18. Questões de competência da ontoDecideDRC após evolução por mudança de foco

<b>Número da QC</b>	<b>Descrição da QC</b>
QC01	O que é DRC?
QC02	O que é TFG?
QC03	Quais são as categorias dos níveis de albuminúria?
QC04	Quais as atribuições do Médico da Atenção Primária?
QC05	O Médico da Atenção Primária trata todos os estágios da DRC?
QC06	Quem deve conduzir a DRC G2?
QC07	Quais os estágios da DRC tratados pelo MAP?
QC08	Como conduzir o estágio 1 da DRC
QC09	Como conduzir o estágio 3A da DRC?
QC10	Qual o risco de DRC de um paciente cuja albuminúria foi classificada no nível 3?

Fonte: As QCs foram criadas pelo Médico Especialista do Domínio da Nefrologia

Tabela 19. Consultas SPARQL criadas a partir das questões de competência da versão da ontologia ontoDecideDRC em evolução por mudança de foco

<b>Número da QC</b>	<b>Consulta SPARQL</b>
QC01	<pre>SELECT ?isDefinedBy WHERE {   { ?subject rdfs:isDefinedBy ?isDefinedBy }   FILTER (?subject=ontodrc:DoencaRenalCronica) }</pre>
QC02	<pre>SELECT ?isDefinedBy WHERE {   { ?subject rdfs:isDefinedBy ?isDefinedBy }   FILTER(? subject=ontodrc:TaxaDeFiltracaoGlomerular) }</pre>
QC03	<pre>SELECT ?subject WHERE{   {?subject rdfs:subClassOf   ontodrc:CategoriasDosNiveisDeAlbuminuria } }</pre>
QC04	<pre>SELECT ?isDefinedBy WHERE {</pre>

```

    { ?subject rdfs:isDefinedBy ?isDefinedBy }
    FILTER(?subject=ontodrc:MedicoDaAtencaoPrimaria)}

QC05  ASK{
    {?condutas rdfs:subClassOf ontodrc:Conduta }
    {?condMAP owl:onProperty ontodrc:ERealizadoPor }
    {?condMAP owl:someValuesFrom
    ontodrc:MedicoDaAtencaoPrimaria }
    FILTER(?condutas = ?condMAP)}

QC06  SELECT DISTINCT ?conduta
    WHERE{
    { ?conduta rdfs:subClassOf ontodrc:CondutaEstagio2 }
    { ?restriction rdf:type owl:Restriction }
    { ?restriction owl:onProperty ontodrc:ERealizadoPor }}

QC07  SELECT DISTINCT ?entidade
    WHERE {
    ?entidade rdfs:subClassOf ontodrc:Estadiamento .
    ?restriction rdf:type owl:Restriction .
    ?restriction owl:onProperty ontodrc:ETratadoPor .
    ?restriction owl:someValuesFrom
    ontodrc:MedicoDaAtencaoPrimaria .}
    ORDER BY ?entidade

QC08  SELECT ?isDefinedBy
    WHERE {
    { ?subject rdfs:isDefinedBy ?isDefinedBy }
    FILTER(?subject=ontodrc:CondutaEstagio1)
    }

QC09  SELECT ?isDefinedBy
    WHERE {
    { ?subject rdfs:isDefinedBy ?isDefinedBy }
    FILTER(?subject=ontodrc:CondutaEstagio3A) }

QC10  SELECT *
    WHERE {
    ?entidade rdfs:subClassOf ?restriction .

```

---



---

```

?restriction rdf:type owl:Restriction .
?restriction owl:onProperty ontodrc:temRisco .
FILTER (?entidade=ontodrc:A3)

```

---

Fonte: Aatoria própria

Com a utilização do *plug-in SPARQL query* implementado para o Protégé, e do raciocinador para ontologias desenvolvidas em OWL, Hermit<sup>34</sup>, tais consultas foram aplicadas à ontologia de modo a verificar sua consistência. Algumas consultas não obtiveram respostas, o que levou ao retorno à correção de inconsistências para verificar o porquê e corrigir o problema. Após a correção, as consultas foram novamente aplicadas. Esse processo foi realizado até a obtenção da resposta esperada, o que permitiu concluir que as inconsistências foram corrigidas por completo. Nas Figuras 40 e 41 pode ser observado o retorno da ONTODRC para a consulta baseada na QC07 e 10 respectivamente. O retorno da ontologia para as demais consultas pode ser visto no Anexo A.

Após a auditoria, a ontologia foi encaminhada para o versionamento. A ferramenta de versionamento escolhida foi o Github<sup>35</sup>. Na Figura 42, a ontologia pode ser visualizada na ferramenta. A mesma também pode ser acessada por meio do seguinte endereço: <https://github.com/LIAALabUFPB/ONTODRC>.

Figura 40. Retorno da ONTODRC para a QC07 (Quais os estágios da DRC tratados pelo MAP?)

The screenshot shows a SPARQL query interface. At the top, there is a purple header with the text "SPARQL query:". Below this, the query is displayed in a monospaced font. The query includes several prefixes for RDF, OWL, RDFS, XSD, and the specific ontology (ontodrc). The main query is a SELECT statement that filters for entities that are subclasses of 'Estadiamento', are restrictions, and are instances of 'ETratadoPor' with some values from 'MedicoDaAtencaoPrimaria'. The results are ordered by the entity name. The results table has a single column labeled 'entidade' and contains three rows of data.

```

SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ontodrc: <http://www.semanticweb.org/ceciliapegado/ontologies/2016/12/ontodrc.owl#>
|
SELECT DISTINCT ?entidade WHERE {
  ?entidade rdfs:subClassOf ontodrc:Estadiamento .
  ?restriction rdf:type owl:Restriction .
  ?restriction owl:onProperty ontodrc:ETratadoPor .
  ?restriction owl:someValuesFrom ontodrc:MedicoDaAtencaoPrimaria .
}
ORDER BY ?entidade

```

entidade
DoencaRenalCronicaEstagio1
DoencaRenalCronicaEstagio2
DoencaRenalCronicaEstagio3A

Fonte: SPARQL *query* - Protégé

<sup>34</sup><http://www.hermit-reasoner.com/>

<sup>35</sup><https://github.com/>

Figura 41. Retorno da ONTODRC para a QC10 (Qual o risco de DRC de um paciente cuja albuminúria foi classificada no nível 3?)

```
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ontodrc: <http://www.semanticweb.org/ceciliapegado/ontologies/2016/12/ontodrc.owl#>

SELECT * WHERE {
  ?entidade rdfs:subClassOf ?restriction .
  ?restriction rdf:type owl:Restriction .
  ?restriction owl:onProperty ontodrc:temRisco .
  FILTER (?entidade=ontodrc:A3)
}
```

entidade
A3

restriction
● temRisco some Alto

Fonte: Adaptado do SPARQL query – Protégé

Figura 42. Ontologia ONTODRC no Github

LIAALabUFPB / ONTODRC Private

Watch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Ontologia OntoDecideDRC após a mudança do foco da mesma para o encaminhamento do paciente. O novo escopo da Ontologia compreende informações sobre os passos que devem ser seguidos para realizar o diagnóstico e estadiamento da DRC até o estágio 3A. [Edit](#)

Manage topics

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

 anapatriciagh Ontologia OntoDecideDRC após evolução por mudança de foco Latest commit da7eb79 22 minutes ago

Fonte: Github

## 6.2 Uso do *framework* em uma ontologia em evolução por mudanças em suas diretrizes de domínio

As Diretrizes Clínicas para o cuidado do paciente com DRC no SUS, principais fontes de conhecimento da ontoDecideDRC, não sofreram alterações no período durante o qual foram feitas as validações do *framework*. Porém em Junho de 2018, foi lançada a portaria nº 1.675<sup>36</sup>. Tal portaria altera a Portaria de Consolidação nº 3/GM/MS, de 28 de setembro de 2017, e a Portaria de Consolidação nº 6/GM/MS, de 28 de setembro de 2017 que dispõem sobre os critérios de organização, funcionamento e financiamento do cuidado da pessoa com Doença Renal Crônica - DRC no âmbito do Sistema Único de Saúde – SUS (MINISTÉRIO DA SAÚDE, 2018).

Assim, as alterações realizadas referentes ao conteúdo representado na ontologia foram identificadas e especificadas conforme sugere o *framework*, para que pudessem, em seguida, ser implementadas na ontologia. Durante o planejamento, também foram consideradas algumas mudanças sugeridas pelo especialista do domínio da ontologia. Na Tabela 20 podem ser observadas algumas das mudanças especificadas para serem implementadas na ontoDecideDRC.

Tabela 20. Lista de mudanças especificadas para aplicação na versão da ontologia ontoDecideDRC em evolução por mudanças em suas diretrizes de domínio

<b>Ontologia em Evolução:</b> ontoDecideDRC		
<b>Motivo da Evolução:</b> Mudanças nas diretrizes do domínio da ontologia		
<b>Descrição do conceito</b>	<b>Mudança a ser aplicada</b>	<b>Descrição da Mudança</b>
Medico	Alteração	Alterar a descrição da classe “Medico” para “EquipeMultiprofissionalDeAtencaoEspecializadaEmDRC”
Enfermeiro	Adição	Adicionar na hierarquia como conceito filho do conceito “EquipeMultiprofissionalDeAtencaoEspecializadaEmDRC”
Nutricionista	Adição	Adicionar na hierarquia como conceito filho do conceito “EquipeMultiprofissionalDeAtencaoEspecializadaEmDRC”
Psicologo	Adição	Adicionar na hierarquia como conceito filho do conceito “EquipeMultiprofissionalDeAtencaoEspecializadaEmDRC”
MDRD Simplificada	Remoção	Remover conceito e todas as suas

<sup>36</sup>[http://bvsms.saude.gov.br/bvs/saudelegis/gm/2018/prt1675\\_08\\_06\\_2018.html](http://bvsms.saude.gov.br/bvs/saudelegis/gm/2018/prt1675_08_06_2018.html)

MedicoDaAtencaoPrimaria	Alteração	referências da ontologia Mover conceito para um nível acima do seu local atual na hierarquia
ClassificacaoDaDRC	Alteração	Mudar a descrição da classe para “Estadiamento”

Fonte: Autoria própria

As mudanças especificadas foram classificadas em simples ou complexas e depois implementadas na ontologia. Na Tabela 21 podem ser observadas algumas mudanças após processo de classificação.

Tabela 21. Lista de mudanças especificadas e classificadas para aplicação na versão da ontologia ontoDecideDRC em evolução por mudanças em suas diretrizes de domínio

**Ontologia em Evolução:** ontoDecideDRC

**Motivo da Evolução:** Mudanças nas diretrizes do domínio da ontologia

Descrição do conceito	Mudança a ser aplicada	Descrição da Mudança	Tipo de Mudança
Medico	Alteração	Alterar a descrição da classe “Medico” para “EquipeMultiprofissionalDeAtencaoEspecializadaEmDRC”	Complexa
Enfermeiro	Adição	Adicionar na hierarquia como conceito filho do conceito “EquipeMultiprofissionalDeAtencaoEspecializadaEmDRC”	Simple
MDRD Simplificada	Remoção	Remover conceito e todas as suas referências da ontologia	Simple
MedicoDaAtencaoPrimaria	Alteração	Mover conceito para um nível acima do seu local atual na hierarquia	Complexa

Fonte: Autoria própria

Iniciando pelas complexas, as mudanças foram implementadas na ontologia utilizando o editor Protégé. A mesma ferramenta foi utilizada para que o *plug-in* OWL *Differences* pudesse ser utilizado com a finalidade de executar a atividade de detecção das mudanças aplicadas. Similarmente à evolução da ontologia por mudança de foco, a partir das mudanças detectadas foram identificados os impactos das mudanças e conseqüentemente as inconsistências causadas por esses impactos. Na Tabela 22 podem ser observadas algumas

das inconsistências identificadas e classificadas após a análise do *log* de diferenças fornecido pela ferramenta *diff*.

Tabela 22. Inconsistências identificadas na versão da ontoDecideDRC em evolução por mudanças em suas diretrizes de domínio

<b>Ontologia em Evolução:</b> ontoDecideDRC			
<b>Motivo da Evolução:</b> Mudanças nas diretrizes do domínio da ontologia			
<b>Entidade Afetada</b>	<b>Mudança aplicada</b>	<b>Inconsistências</b>	
		<b>Estruturais</b>	<b>Lógicas</b>
Medico	Alteração	-Erros nos axiomas de subclasse do conceitos filhos da classe Medico -Impossível determinar o conceito pai da classe Nefrologista	-Impossível inferir que os estágios da DRC são tratados por algum médico -Impossível inferir que os pacientes com DRC são tratados por algum médico
MDRD Simplificada	Remoção	-Impossível encontrar a classe MDRD Simplificada <i>SubclassOf</i> “Formulas”	-Impossível inferir sobre as regras que usam a fórmula
MedicoDaAtencao Primaria	Alteração decorrente da mudança da classe “Medico”	-Impossível encontrar a superclasse desse conceito	Impossível inferir sobre os axiomas que determinam as relações de subclasse e superclasse.
Nefrologista	Alteração decorrente da mudança da classe “Medico”	-Impossível encontrar a superclasse desse conceito	Impossível inferir sobre os axiomas que determinam as relações de subclasse e superclasse.

Fonte: Autoria própria

A renomeação da classe médico é entendida e representada pela ferramenta *diff* como uma deleção seguida da adição de um conceito com o novo nome dado à classe. Assim, os axiomas que utilizam a classe supostamente deletada conseqüentemente também são deletados. Por esse motivo, as classes que sofreram essas alterações indiretas também são exibidas na ferramenta *diff*. A adição do novo nome, auxilia na recuperação da consistência dessas regras permitindo portanto que as inferências sobre os conceitos envolvidos possam ser realizadas novamente.

As inconsistências identificadas foram corrigidas utilizando a ferramenta Protégé e tendo o relatório de inconsistências identificadas como direcionador. Após realizadas as devidas correções, foram aplicadas consultas SPARQL baseadas nas QCs da ontologia de

modo a verificar sua consistência do modo como sugere o *framework*. A aplicação das consultas também foi feita por meio da utilização do *plug-in* SPARQL *query* e do raciocinador HerMiT.

As questões de competência da ontologia podem ser observadas na Tabela 23. Como essa versão da ontologia faz parte do mesmo domínio da versão submetida a evolução por mudança de foco, algumas questões de competência como “O que é DRC?” e “O que é TFG?”, são similares para as duas versões. Assim, na Tabela 23 constam apenas as QCs próprias da versão em evolução por mudanças nas diretrizes de domínio da ontologia.

Tabela 23. Questões de competência da ontoDecideDRC

<b>Número da QC</b>	<b>Descrição da QC</b>
QC01	Quais as atribuições do Nefrologista?
QC02	O Nefrologista trata todos os estágios da DRC?
QC03	Como conduzir a DRC G5?
QC04	Quais os estágios da DRC tratados pelo Nefrologista?
QC05	Como conduzir a DRC G4?
QC06	Quem são integrantes da equipe multiprofissional de atenção especializada em DRC?
QC07	Quais as atribuições da equipe multiprofissional de atenção especializada em DRC?
QC08	Como conduzir o estágio 3 da DRC

Fonte: Autoria própria

Na Tabela 24 podem ser observadas as consultas SPARQL criadas a partir das QCs da ontoDecideDRC. Nas Figuras 43 e 44 estão os retornos para as consultas baseadas nas QCs 01 e 02, respectivamente. O retorno da ontologia para as demais consultas pode ser visto no Anexo B. A ontologia respondeu corretamente a todas as consultas realizadas. Após a auditoria, a mesma foi encaminhada para o versionamento no Github (Figura 45). A ontologia pode ser acessada na ferramenta por meio do seguinte endereço: <https://github.com/LIAALabUFPB/ontoDecideDRC>.

Tabela 24. Consultas SPARQL criadas a partir das questões de competência da versão da ontologia ontoDecideDRC em evolução por mudanças em suas diretrizes de domínio

<b>Número da QC</b>	<b>Consulta SPARQL</b>
---------------------	------------------------

```

QC01  SELECT ?isDefinedBy
      WHERE {
        { ?subject rdfs:isDefinedBy ?isDefinedBy }
        FILTER(?subject=ontodecidedrc:Nefrologista)
      }

QC02  SELECT DISTINCT ?entidade
      WHERE {
        ?entidade rdfs:subClassOf ontodecidedrc:Nefrologista .
        ?restriction rdf:type owl:Restriction .
        ?restriction owl:onProperty ontodecidedrc:Trata .
      }

QC03  SELECT ?conduta5n ?conduta5nd
      WHERE{
        { ?subject ontodecidedrc:Guideline ?conduta5n .
          FILTER(?subject=ontodecidedrc:Estagio5D) }
        UNION
        { ?subject ontodecidedrc:Guideline ?conduta5nd .
          FILTER(?subject=ontodecidedrc:Estagio5ND) }
      }

QC04  SELECT DISTINCT ?entidade ?restriction
      WHERE {
        ?entidade rdfs:subClassOf ?restriction .
        ?restriction rdf:type owl:Restriction .
        ?restriction owl:onProperty ontodecidedrc:ETratadoPor .
        ?restriction owl:someValuesFrom
ontodecidedrc:Nefrologista .
      }
      GROUP BY ?restriction ?entidade
      ORDER BY ?entidade

QC05  SELECT ?conduta4
      WHERE{ { ?subject ontodecidedrc:Guideline ?conduta4 .
        FILTER(?subject=ontodecidedrc:Estagio4) }
      }

```

```

QC06  SELECT ?subject
      WHERE {
        { ?subject rdfs:subClassOf
          ontodecidedrc:EquipeMultiprofissionalDeAtencaoEspecializada
          EmDRC }}
QC07  SELECT ?isDefinedBy
      WHERE {
        { ?subject rdfs:isDefinedBy ?isDefinedBy }
        FILTER(?
          subject=ontodecidedrc:EquipeMultiprofissionalDeAtencaoEspec
          ializadaEmDRC)
        }
QC08  SELECT ?conduta3a ?conduta3b
      WHERE{
        { ?subject ontodecidedrc:Guideline ?conduta3a .
          FILTER(?subject=ontodecidedrc:Estagio3A) }
        UNION
        { ?subject ontodecidedrc:Guideline ?conduta3b .
          FILTER(?subject=ontodecidedrc:Estagio3B) }
        }

```

Fonte: Autoria própria

Figura 43. Retorno da ontoDecideDRC para a QC01 (Quais as atribuições do Nefrologista?)

```

SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ontodecidedrc: <http://www.semanticweb.org/esaualeucar/ontologies/2015/9/OntoDecideDRC#>

SELECT ?isDefinedBy
WHERE {
  { ?subject rdfs:isDefinedBy ?isDefinedBy }
  FILTER(?subject=ontodecidedrc:Nefrologista)
}

```

isDefinedBy

"O profissional de Nefrologia é especializado no tratamento clínico das doenças do sistema urinário, principalmente relacionadas às

Fonte: SPARQL query – Protégé



Figura 44. Retorno da ontoDecideDRC para a QC02 (O Nefrologista trata todos os estágios da DRC?)

```
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ontodecidedrc: <http://www.semanticweb.org/esaualeucar/ontologies/2015/9/OntoDecideDRC#>

SELECT DISTINCT ?entidade
WHERE {
?entidade rdfs:subClassOf ontodecidedrc:Nefrologista .
?restriction rdf:type owl:Restriction .
?restriction owl:onProperty ontodecidedrc:Trata .
}
```

entidade
● Trata some (Estagio3B or Estagio4 or Estagio5D or Estagio5ND)

Fonte: SPARQL query – Protégé

Figura 45. Ontologia ontoDecideDRC no Github



Fonte: Github

### 6.3 Uso do *framework* em uma ontologia em evolução por mudanças em sua terminologia de referência

A versão após mudança de foco foi a utilizada para passar por evolução por mudanças na terminologia de referência. No processo de concepção da ontoDecideDRC, para padronizar as descrições dos termos da ontologia foram utilizadas as Diretrizes Clínicas para o Cuidado do Paciente com DRC no SUS como base para a criação de um Léxico Ampliado da Linguagem (LAL) (TAVARES, 2016). No entanto, com a mudança de foco, a ontologia passou a ter mais de uma fonte de conhecimento e assim, passaria a ter que usar bases diferentes para a representação terminológica. Por esse motivo, a equipe de evolução optou

por implementar a SNOMED CT como padrão único de representação dos termos da ontologia.

Alguns termos próprios do contexto da ontologia como por exemplo, o conceito utilizado para descrever as formas de conduta de pacientes em cada estágio da doença (“CondutaEstagio1”, “CondutaEstagio2”...), não existem no repositório de termos SNOMED CT. Assim, os termos inexistentes na terminologia foram classificados e organizados em um relatório de modo a serem adicionados na terminologia como expressões pós-coordenadas (seção 2.5). Após a decisão de criação das expressões pós-coordenadas, segundo a Sociedade Brasileira de Informática em Saúde (SBIS), o Brasil passou a ser país membro da terminologia SNOMED CT a partir de abril de 2018 e atualmente, está sendo realizada a definição de SCTIDs para conceitos exclusivos do cuidado em saúde brasileiro. Ainda segundo informações da SBIS, uma extensão da SNOMED CT para termos clínicos utilizados no Brasil será lançada em 2019. Considerando essas informações, a equipe responsável pela evolução da ontologia para padronização total decidiu aguardar o lançamento da extensão de modo que a ontologia pudesse utilizar os termos adequados e não fosse necessária a utilização de termos em uma língua distinta. Porém, de modo que o estudo de caso diante dessa perspectiva de evolução não fosse prejudicado, as atividades realizadas para padronização utilizando a edição internacional do repositório de termos SNOMED CT apenas para os termos encontrados, está descrita nos próximos parágrafos. Foi criada uma *annotation* do tipo *boolean* chamada “conceitoPadronizado” para identificar os conceitos para os quais não foram encontrados termos correspondentes na terminologia.

Seguindo as atividades sugeridas pelo *framework*, tendo em vista que a terminologia SNOMED CT é internacional e a ontologia em evolução foi construída em Português, após a escolha da terminologia, os conceitos da ontologia foram traduzidos. A língua escolhida foi o inglês, tendo em vista que essa é a linguagem do repositório SNOMED CT com maior número de termos. De modo que fosse realizada a tradução correta e assim se pudesse reduzir o tempo gasto com a escolha do termo no repositório durante as buscas, o processo de tradução foi auxiliado pela versão em inglês do documento de Avaliação e Gerenciamento da DRC (do inglês, *CKD Evaluation and Management*) fornecido e mantido pela KDIGO.

Após a tradução, foram realizadas as buscas pelos termos da terminologia por meio de palavras-chave presentes nos conceitos da ontologia após serem traduzidos. Os termos encontrados na terminologia referentes aos conceitos da ontologia foram organizados em uma tabela. Foram trazidos para a ontologia o SCTID e o FSN referentes a cada um dos seus conceitos. Na Tabela 25 podem ser observados alguns dos termos mapeados.

Tabela 25. Relatório de mudanças a serem aplicadas na ontologia ONTODRC em evolução para padronização terminológica total

<b>Ontologia em Evolução: ONTODRC</b>		
<b>Motivo da Evolução: Padronização terminológica total com SNOMED CT</b>		
<b>Descrições</b>		<b>SCTID</b>
<b>Anterior</b>	<b>Atual</b>	
Estadiamento	<i>Staging of disease (attribute)</i>	260760003
Doença Renal Crônica	<i>Chronic Kidney Disease (disorder)</i>	709044004
Doença Renal Crônica Estágio 1	<i>Chronic Kidney Disease Stage 1 (disorder)</i>	431855005
Doença Renal Crônica Estágio 2	<i>Chronic Kidney Disease Stage 1 (disorder)</i>	431856006
Doença Renal Crônica Estágio 3A	<i>Chronic kidney disease stage 3A (disorder)</i>	700378005
Diabetes Mellitus	<i>Diabetes Mellitus</i>	73211009
Doença Policística Renal	<i>Congenital cystic kidney disease (disorder)</i>	82525005
Hipertensão Arterial Sistêmica	<i>Hypertensive disorder, systemic arterial (disorder)</i>	38341003
Litíase Renal	<i>Kidney stone (disorder)</i>	95570007
Médico da Atenção Primária	<i>Primary care physician (occupation)</i>	446050000
Taxa de Filtração glomerular	<i>Glomerular filtration rate (observable entity)</i>	80274001
Peso	<i>Body weight (observable entity)</i>	27113001
Gênero	<i>Gender (observable entity)</i>	263495000
Idade	<i>Age (qualifier value)</i>	397669002
Raça	<i>Race (observable entity)</i>	103579009
Paciente	<i>Patient (person)</i>	116154003
Alto Risco	<i>High risk (qualifier value)</i>	723509005
Baixo Risco	<i>Low risk (qualifier value)</i>	723505004
Risco Moderado	<i>Moderate risk of (contextual qualifier) (qualifier value)</i>	25594002
EQU_EAS_UrinaTipo1	<i>Urine finding (finding)</i>	301830001
Relação Albuminúria Creatininúria	<i>Albumin/creatinine ratio measurement (procedure)</i>	250745003

A1	<i>Microalbuminuria (finding) Grade A2 albuminuria (disorder)</i>	312975006
A2	<i>Microalbuminuria (finding)</i>	735623008
A3	<i>Grade A3 albuminuria (disorder)</i>	735624002

Fonte: Autoria própria

Após o mapeamento, foi realizada a edição da ontologia utilizando a ferramenta Protégé. Foram criadas *annotations* para comportar cada elemento correspondente ao conceito identificado na terminologia. Por exemplo, para o FSN foi criada uma *annotation* chamada “FSN” do tipo *String*. Após a criação, para cada conceito da ontologia foi adicionado o valor do FSN e o do SCTID em suas respectivas *annotations*. O uso das *annotations* para adição dos elementos da terminologia referentes ao conceito da ontologia, permite que a URI (do inglês, *Uniform Resource Identifier*) do conceito seja mantida na linguagem original, pois a possibilidade de renderizar a ontologia para que os conceitos sejam visualizados por alguma *annotation*, dispensa a necessidade de renomear os mesmos.

A manutenção da ontologia na linguagem original é importante pois conserva a representação principal dos termos conforme seu idioma de origem e não restringe o uso da ontologia. Além disso, as consultas na ontologia poderão ser realizadas com o uso do idioma original do termo, tendo em vista que a URI é a descrição do conceito considerada pelas linguagens de consulta. De modo a evitar erros de consistência e consequentemente não atrapalhar o processo de raciocínio da ontologia, todos os termos foram adicionados na mesma sem espaços ou caracteres especiais. Na Figura 46 pode ser observado o resultado da padronização do termo “DoencaRenalCronica”. Na imagem, a ontologia foi renderizada pela *annotation* FSN.

Figura 46. Classe “DoencaRenalCronica” da ontologia ONTODRC após padronização com SNOMED CT

The screenshot shows the Protégé interface for the class 'ChronicKidneyDisease(disorder)'. The URL is <http://www.semanticweb.org/ceciliapegado/ontologies/2016/12/ontodrc.owl#DoencaRenalCronica>. The class annotations are:

- Hematúria persistente; ou
- Alterações anatômicas que provoquem lesão ou perda de função renal; ou
- Perda rápida da função renal (> 5 /min/1,73 m2 em seis meses, com uma TFG < 60 /min/1,73 m2, confirmado em dois exames); ou p potencial patológico (céreos, largos, graxos, epiteliais, hemáticos ou leucocitários).

(MINISTÉRIO DA SAÚDE, 2015)

The identifiers are:

- FSN [type: xsd:string]  
ChronicKidneyDisease(disorder)
- SCTID [type: xsd:integer]  
709044004

Fonte: Protégé

Após o término da padronização a ontologia foi encaminhada para auditoria, onde foi realizada a análise da conformidade da mesma com o seu plano de evolução. Nesse perspectiva de evolução a análise da conformidade tem como finalidade verificar se todos os termos foram padronizados. Todas as atualizações referentes à padronização da ontologia foram disponibilizadas no Github.

#### 6.4 Conclusão do capítulo

Neste capítulo foi realizada a evolução da ontologia OntoDecideDRC(TAVARES, 2016) com a finalidade de validar a aplicabilidade do *framework* diante de três perspectivas por meio das quais uma ontologia em evolução poderá ser submetida, de acordo com o motivo pelo qual a mesma encontra-se em evolução. Os três motivos de evolução que levaram à definição das perspectivas citadas foram identificados no estudo de caso realizado pra a obtenção de requisitos para a construção do *framework*. Tal estudo de caso encontra-se descrito na seção 4.1.2. Considerando os motivos de evolução citados, duas cópias da ontologia OntoDecideDRC foram submetidas a evolução. Uma delas teve a evolução motivada pela sua mudança de foco e a outra por mudanças em suas diretrizes de domínio.

A ontologia OntoDecideDRC aborda conceitos relativos ao diagnóstico, estadiamento, encaminhamento e tratamento de pacientes com DRC. Ao passar por processo evolutivo para refletir os conceitos referentes ao seu novo foco, a ontologia passou a abordar conceitos relacionados ao cuidado realizado antes do encaminhamento do paciente para a atenção especializada, ou seja, o novo foco da ontologia está voltado para os procedimentos realizados

com o paciente durante a Atenção Primária de Saúde. Após serem implementadas as mudanças exigidas pelo seu novo foco, a ontologia foi encaminhada para a fase de Detecção de Mudanças, onde as versões da mesma foram comparadas e analisadas. Como não ocorreram rejeições de mudanças, a ontologia e o relatório de mudanças deferidas foram encaminhados para fase de Tratamento de Inconsistências. Resolvidas as inconsistências identificadas e classificadas, a ontologia foi encaminhada para a fase de Auditoria de Mudanças. Pela sua alta expressividade, a linguagem SPARQL foi escolhida para traduzir as questões de competência da ontologia e testar sua consistência. A github foi a ferramenta utilizada para versionar a ontologia.

Um processo similar foi realizado para a evolução da OntoDecideDRC por mudanças em suas diretrizes de domínio. Apenas as QCs utilizadas para testar a consistência dessa cópia da ontologia são, em partes (algumas QCs foram reusadas pelo fato das ontologias pertencerem ao mesmo domínio), diferentes das utilizadas para testar a consistência da cópia da ontologia que teve sua evolução motivada por mudança de foco. Na verificação da consistência de ambas as cópias da ontologia, foi necessária a aplicação múltipla de algumas QCs para as quais a ontologia não forneceu a resposta esperada de imediato.

A ontologia resultante da evolução por mudança de foco, ONTODRC, foi submetida à evolução motivada por alterações relativas à terminologia de referência da mesma. Alterações relacionadas à terminologia de referência em uma ontologia podem ser feitas para refletir, na ontologia que a utiliza, alterações ocorridas na terminologia, ou para realizar a padronização total da mesma. Nesse estudo de caso, a ONTODRC foi submetida à evolução para realizar a padronização total da sua nomenclatura. Alguns conceitos próprios do domínio de interesse não foram identificados na terminologia. De modo a identificar os termos não padronizados na ontologia, foi criada uma *annotation* booleana para a qual foi atribuído o valor *true* caso o termo tivesse sido padronizado e *false* para aqueles para os quais não foram detectados termos correspondentes. Considerando que não houveram alterações relativas às regras da ontologia, no processo de Auditoria de Mudanças apenas a conformidade da ontologia em relação ao plano de evolução foi verificada. Todas as versões resultantes dos processos evolutivos realizados para validar o *framework* estão sendo versionadas na ferramenta Github.

## 6.5 Notas sobre o capítulo

O versionamento da ontologia foi realizado utilizando o github por ser uma ferramenta de fácil acesso pelos usuários da mesma e por permitir a colaboração de todos os usuários com permissão para alterar a ontologia. Existem ferramentas próprias de compartilhamento e visualização de ontologias que também podem ser utilizadas. A principal contribuição da validação do *framework* foi a possibilidade de visualização e entendimento do seu funcionamento em uma perspectiva prática, tendo em vista que desse modo, as limitações do mesmo podem ser exploradas visando a implementação de melhorias.

# Capítulo 7

## 7 Conclusões

### 7.1 Contribuições

O *framework* desenvolvido possui como principal benefício o fornecimento de recursos conceituais organizados como um conjunto de processos para apoiar a evolução de ontologias Biomédicas, desde a representação da mudança até a validação da ontologia, considerando três perspectivas de evolução advindas da mudança do foco, mudanças nas diretrizes de domínio da ontologia e mudanças relativas à sua terminologia de referência. A organização do *framework* como um conjunto de fases compostas por atividades ordenadamente executadas, colabora para um processo evolutivo organizado e possivelmente com menos erros, pois a execução individualizada de cada tarefa que compõe determinada fase do *framework*, permite a visão completa da mesma o que possibilita enxergar com maior facilidade as falhas que venham a ocorrer.

A estrutura na qual estão dispostas as fases do *framework* foi motivada pelo fato do mesmo ser voltado para ontologias Biomédicas, pois a maior complexidade, ocasionada pelo grande número de conceitos, relacionamentos e regras desse tipo de ontologias, dificulta a visualização das mudanças necessárias em seu conteúdo. Assim, acreditando-se oferecer, essa estrutura, um maior detalhamento em cada etapa da evolução, optou-se por utilizá-la com a finalidade de auxiliar na visualização das mudanças necessárias bem como na resolução das inconsistências ocorridas.

Planejar a evolução da ontologia é crucial para a sua efetividade, pois o planejamento funciona como um guia de todo o processo evolutivo, facilitando a identificação de falhas durante a execução do mesmo. A fase de Planejamento da Evolução, primeira fase do *framework*, foi pensada para ter a finalidade de representar corretamente as mudanças a serem aplicadas e nortear a evolução de modo a garantir que as mudanças planejadas sejam efetivamente implementadas. Por esse motivo é de extrema importância a utilização do plano de evolução na última etapa do *framework*, onde verifica-se se a ontologia reflete as



mudanças requeridas durante o planejamento do processo evolutivo.

O processo de Planejamento da Evolução é composto pelas atividades de captura das mudanças e representação das mesmas. A captura das mudanças a serem aplicadas na ontologia permite que se identifique as fontes de conhecimento da ontologia em evolução e que a sua análise evite que mudanças necessárias não sejam implementadas. A atividade de representação das mudanças possui como finalidade a especificação das mudanças a serem realizadas em relação ao seu tipo, ou seja, mudanças que refletem adições, alterações ou remoções de conceitos deverão ser identificadas no plano de evolução. A representação das mudanças desse modo fornece maior clareza na interpretação das mudanças antes que estas sejam efetivamente aplicadas. Assim, o *framework* construído permite que se tenha uma visão mais clara e ampla das mudanças requeridas, o que torna mais fácil o processo de aplicação dessas mudanças.

O uso do *framework* também poderá facilitar o processo de identificação das inconsistências decorrentes das mudanças aplicadas, pois a atividade de Classificação de Inconsistências facilita sua visualização e colabora com a definição das estratégias de correção mais adequadas. A visualização das mudanças semanticamente representadas bem como das inconsistências classificadas também permite que as mesmas sejam usadas em evoluções posteriores, facilitando a identificação de mudanças necessárias na ontologia e de inconsistências com base nas já ocorridas, bem como na formulação das melhores formas de aplicação das mudanças e resolução de inconsistências.

Durante a descrição do *framework* bem como em sua validação realizada por meio do estudo de caso descrito no capítulo 6, diversas ferramentas para apoio às atividades do processo evolutivo foram sugeridas, dentre elas estão a ferramenta Protégé, editor de ontologias desenvolvido e mantido por pesquisadores da Universidade de *Stanford*; ferramentas *diff* como *PromptDiff* criada por Noy e Musen (2002), *Conto-Diff* criada por Hartung, Groß e Rahm (2013) e *OWL Differences*, criada como um *plug-in* para o Protégé; *plug-ins* utilizados como ferramentas *diff* em editores como Protégé para visualização de mudanças também fazem parte das ferramentas sugeridas. São exemplos desse tipo de *plug-ins* o *Changes Tab*, *Version Log Generator* e *Changes Tracer*. A ferramenta *NeOn Toolkit* é outra alternativa para edição de ontologias, a mesma pode ser usada para visualizar mudanças

por meio do *plug-in ChangeCapturing*.

Para apoio ao processo de padronização terminológica de ontologias em evolução para refletir alterações ocorridas em sua terminologia de referência, foram sugeridos o BioPortal *Import plug-in*, disponível para a ferramenta Protégé. Essa ferramenta permite que os usuários importem classes de ontologias que estejam disponíveis no BioPortal, um repositório para armazenar e gerenciar ontologias durante sua vida útil. Com a mesma finalidade também foi sugerida a *Snow OWL Meaningful Query*, uma plataforma semântica gratuita *online* para processamento de dados em saúde que permite, dentre outras funcionalidades, o acesso a conceitos de mais de 70 terminologias incluindo SNOMED CT.

A detecção das mudanças efetivamente implementadas na ontologia em evolução é imprescindível para auxiliar na fase de Tratamento de Inconsistências ocorridas na mesma. Neste trabalho, o uso das ferramentas *diff* foram sugeridas como elementos facilitadores do processo de detecção de mudanças e conseqüentemente de detecção das inconsistências. Sabendo-se quais foram as entidades afetadas pelas mudanças, bem como os axiomas afetados por elas (informação fornecida pelas ferramentas *diff*) o processo de identificação de inconsistências se tornará menos trabalhoso. A fase de Tratamento de Inconsistências foi concebida com a intenção de possibilitar uma melhor visualização dos problemas ocorridos na estrutura taxonômica e lógica da ontologia, considerando que nas ontologias Biomédicas, a grande quantidade de conceitos representados dificulta a visualização dos problemas decorrentes das mudanças. Desse modo, além da detecção de inconsistências, que tem como finalidade identificar os erros ocorridos na ontologia a partir das mudanças detectadas, ao tratar as inconsistências recomenda-se que o as mesmas sejam classificadas antes de serem corrigidas.

Conforme o *framework*, as inconsistências da ontologia em evolução deverão ser classificadas em lógicas ou estruturais. Essa classificação permite que as inconsistências estruturais, muitas vezes causadoras das inconsistências lógicas sejam resolvidas antes e assim, os problemas lógicos delas decorrentes sejam automaticamente resolvidos. Esse processo pode então vir a reduzir os custos relativos ao tempo e/ou recursos humanos durante o processo evolutivo.

Um estudo de caso foi realizado com a finalidade principal de demonstrar o

comportamento do *framework* diante das três perspectivas de evolução definidas a partir dos motivos de evolução identificados e assim verificar o seu desempenho quando aplicado a um caso do mundo real. A validação das ontologias resultantes do estudo de caso utilizou as suas questões de competência em conjunto com a linguagem SPARQL, linguagem de consulta para RDF (tendo em vista que as ontologias foram desenvolvidas em OWL) que permite a tradução de um grande número de QCs devido à sua alta expressividade. As QCs das versões da ontologia, OntoDecideDRC e ONTODRC (resultado da evolução por mudança de foco), foram definidas pela médica especialista em Nefrologia que acompanhou o estudo de caso. O método de validação utilizado mostrou-se eficaz tendo em vista que as ontologias utilizadas no estudo de caso forneceram as respostas esperadas para as QCs. Com base nos resultados dos experimentos realizados no estudo de caso, pode-se concluir que o *framework* permitiu manter organizado e compreensivo o processo evolutivo das cópias da ontologia utilizada bem como oferece os recursos necessários para a evolução apropriada de ontologias Biomédicas.

A utilização da terminologia SNOMED CT no estudo de caso onde a ontologia OntoDecideDRC foi submetida à evolução para padronização conceitual total, possibilitou uma visão ampla dos benefícios e funcionamento do padrão de termos clínicos e permitiu ainda a obtenção de conhecimentos relativos à sua utilização, o que levou a um melhor aproveitamento dos seus recursos.

Os relatórios fornecidos como saída das atividades que compõem os subprocessos do *framework*, além de subsidiarem a realização das atividades às quais são dados como entradas, podem ser utilizados como ferramentas de predição de mudanças, por meio das quais as mudanças recorrentes podem ser identificadas, o que pode ajudar na definição prévia de estratégias de aplicação de mudanças bem como de correção de erros identificados durante a validação da ontologia.

## **7.2 Trabalhos Futuros**

As diretrizes pertencentes a qualquer domínio sofrem mudanças para incorporar novas descobertas ocorridas ao longo do tempo. Com o domínio biomédico não é diferente. As diretrizes clínicas para o cuidado de pacientes com DRC no SUS, por exemplo, têm sofrido mudanças a cada dez anos. Considerando esse fato, propõe-se como trabalhos futuros uma

abordagem mais profunda e detalhada pelo *framework* construído e descrito nesta dissertação, do processo de evolução motivado por mudanças nas diretrizes de domínio da ontologia.

Tal detalhamento tem como finalidade auxiliar o Engenheiro de ontologias durante o processo de evolução por meio da definição e formalização de métodos de representação e implementação das mudanças ocorridas nas diretrizes de domínio da ontologia em evolução. Para tanto, serão utilizadas como base, as mudanças ocorridas nas versões existentes dessas diretrizes como forma de identificar padrões de mudanças para buscar modos de representação e implementação das mesmas.

Também se apresentam como trabalhos futuros desta dissertação, um detalhamento maior da fase de Tratamento de Inconsistências com vistas a fornecer soluções para um número maior de erros que podem advir das inconsistências ocorridas após a implementação das mudanças.

No estudo de caso realizado, os relatórios com as mudanças representadas foram construídos utilizando ferramentas de edição de texto para modelagem de tabelas. Considerando as dificuldades encontradas durante a execução dessa atividade por ser um processo em grande parte manual e por não oferecer a possibilidade de controle das requisições de mudanças, propõe-se como trabalhos futuros, a modelagem e construção de uma ferramenta de *software* para solicitação e representação das mudanças requeridas pelos usuários da ontologia em evolução.

Os trabalhos manuais realizados durante o mapeamento dos termos da ontologia no repositório do padrão de termos clínicos SNOMED CT, motivaram uma proposta de trabalho futuro a qual tem como finalidade, a construção de um *plug-in* para o editor de ontologias Protégé que possa ser utilizado para realizar o mapeamento automático dos termos da ontologia em evolução de acordo com o lançamento dos novos *releases* da terminologia.

Tendo em vista que as grandes ontologias Biomédicas em geral são mantidas de forma colaborativa por usuários geograficamente distribuídos, como trabalhos futuros pretende-se adaptar o *framework* para que este possa dar suporte ao desenvolvimento colaborativo e assim apoiar a o processo de evolução de ontologias Biomédicas distribuídas.

## **Bibliografia**

ABGAZ, Y. M.; JAVED, M.; PAHL, C. A framework for change impact analysis of ontology- driven content-based systems. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 7046 LNCS, p. 402–411, 2011.

ABGAZ, Y. M.; JAVED, M.; PAHL, C. Analyzing impacts of change operations in evolving ontologies. **CEUR Workshop Proceedings**, v. 890, 2012.

AITKEN J.S.; WEBBER B.L.; BARD J.B. Part-of relations in anatomy ontologies: a proposal for RDFS and OWL formalisations. **Pac Symp Biocomput. Bethesda**. p.166-77. 2004.

AL-HABLANI, B. The Use of Automated SNOMED CT Clinical Coding in Clinical Decision Support Systems for Preventive Care. **Perspectives in Health Information Management**, v. 14, p. 1–8, 2017.

APRENDIS. **SNOMED CT**. Disponível em: <<http://aprendis.gim.med.up.pt/index.php/SNOMED-CT>>. Acesso em: 19 dez. 2018.

BAYOUDHI, N.; SASSI, L. 'e JAZIRI, W. OWL 2 DL ontology inconsistencies prediction In: **WIMS '17 Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics** , 7., 2017, Amantea, Italy, 2017, Artigo nº 25.

BODENREIDER, O. Biomedical ontologies in action: role in knowledge management, data integration and decision support. **Yearbook of medical informatics**, v. 3841, p. 67–79, 2008.

BORTOLATO, F. **Ligando Dados Governamentais Abertos: uma ontologia do Processo Legislativo de São Paulo Dissertação**. 2014. Dissertação (Mestrado em Engenharia de Computação) - Instituto de Pesquisas Tecnológicas do Estado de São Paulo, São Paulo.

BORTOLATO, F.; PAULO, S. **Instituto de Pesquisas Tecnológicas do Estado de São Paulo Ligando Dados Governamentais Abertos: uma ontologia do Processo Legislativo de São Paulo**. 2014.

BREITMAN, K.; LEITE, J. Ontologias–Como e porque criá-las. **Jornada de Atualização em Informática - Minicursos - SBC**, p. 1–50, 2004.

BUENO, T. C. **Uma metodologia de representação do conhecimento para construção de ontologias em sistemas baseados em conhecimento**. 2005. 174 f. Tese (Doutorado em Engenharia) - Universidade Federal de Santa Catarina, Florianópolis. 2005.

CAMBRIDGE DICTIONARY. **Framework**. Disponível em: <<https://dictionary.cambridge.org/pt/dicionario/ingles/framework/>>. Acesso em: 30 jan. 2019.

CLARKSON; M. Representation of Parts within the Foundational Model of Anatomy ontology. **International Conference on Biological ontology and BioCreative**. Oregon. 2016.

CONITEC. **Manual de elaboração de Diretrizes Clínicas irá preencher lacunas sobre as melhores práticas na elaboração de diretrizes em saúde**. Disponível em: <<http://conitec.gov.br/manual-de-elaboracao-de-diretrizes-clinicas-ira-preencher-lacunas-sobre-as-melhores-praticas-na-elaboracao-de-diretrizes-em-saude>>. Acesso em: 20 jan. 2019.

CUI, L.; TAO, S.; ZHANG, G. Biomedical ontology Quality Assurance Using a Big Data Approach. **ACM Transactions on Knowledge Discovery from Data (TKDD)**, New York, v.10, n.4, 2016.

FAHAD, M.; QADIR, M. A. A framework for ontology evaluation. **CEUR Workshop Proceedings**, v. 354, p. 148–156, 2008.

FLOURIS, G. et al. ontology change: Classification and survey. **The Knowledge Engineering Review**, v. 23, n. 2, p. 117-152, 2008.

FLOURIS, G. *et al.* Inconsistencies, Negations and Changes in ontologies. **Proceedings of the 21st National Conference on Artificial Intelligence AAAI06**, p. 1295–1300, 2006.

FLOURIS, G.; PLEXOUSAKIS, D.; ANTONIOU, G. Evolving ontology Evolution. SOFSEM 2006: **Theory and Practice of Computer Science**. Anais...2006.

GAWITH, M. et al. ontology Evolution Approaches in Medical Domain. **International Journal of Biology and Biomedical Engineering**, v. 10, p. 219-225, 2016.

GENESERETH, M. R.; NILSSON, N. J. **Logical Foundations of Artificial Intelligence**. 1. ed. San Mateo: Morgan Kaufmann Publishers, 1987.

GROS, A.; PRUSKI, C.; RAHM, E. Evolution of biomedical ontologies and mappings: Overview of recent approaches. **Computational and Structural Biotechnology Journal**, v. 14, p. 333–340, 2016.

GRUBER, T. R. A translation approach to portable ontology specifications. **Knowledge Acquisition**, vol. 5, pp. 199–220, 1993.

GRUBER, T. R. Toward Principles for the Design of ontologies Used for Knowledge Sharing [internet]. **International Journal of Human-Computer Studies**. 1995, v. 43. p. 907-928.

GUARINO, N. Formal ontology and Information Systems. **Proceedings of FOIS'98, Trento**, Italy, 6-8 June 1998. Amsterdam, IOS Press, pp. 3-15.

HAASE, P. *et al.* **A Framework for Handling Inconsistency in Changing Ontologies**International Semantic Web Conference. Anais...2005.

HAASE, P.; STOJANOVIC, L. Consistent evolution of owl ontologies.The Semantic Web: Research and Applications In: **ESWC' 05 Proceedings of the Second European conference on The Semantic Web: research and Application**, 2., 2005. Heraklion, Greece. 2005, p. 182-197.

HARTUNG, M; GROß, A. e RAHM, E. Conto-Diff: Generation of complex evolution mappings for life science ontologies. **Journal of Biomedical Informatics**. v. 46, p. 15-32, 2013.

HARTUNG, M.; KIRSTEN, T.; RAHM, E. Analyzing the evolution of life science ontologies and mappings. **Data Integration in the Life Sciences. DILS 2008. Lecture Notes in Computer Science**, v. 5109, p. 11–27, 2008.

HORRIDGE M. et al. WebProtégé: a collaborative Web-based platform for editing biomedical ontologies. **Bioinformatics**. Oxford Journals, Oxford, v. 30, n. 16, p. 2384-2385, 2014.

HORRIDGE, M. *et al.* **A Practical Guide To Building OWL ontologies Using Protégè 4 and CO-ODE Tools Edition 1.3**. 1 ed. Stanford: Matrix, 2011.

HORROCKS, I.; PATEL-SCHNEIDER, P. F.; WELTY, C. A. **The Description Logic Handbook: Theory, Implementation and Applications**. p. 458–486, 2007.

IHTSDO. **SNOMED CT Concept Model**. Disponível em: <<https://confluence.ihtsdotools.org/display/DOCSTART/6.+SNOMED+CT+Concept+Model>>. Acesso em: 12 nov. 2018.

ISOTANI, S.; BITTENCOURT, I. I. Engenharia de ontologias. *In: Dados Abertos Conectados*. 1. ed. [s.l.] Novatec, 2015. p. 176.

JAVED M., ABGAZ YM., PAHL C. A Pattern-Based Framework of Change Operators for ontology Evolution. In: **Meersman R., Herrero P., Dillon T. (eds) On the Move to Meaningful Internet Systems: OTM 2009 Workshops. OTM 2009**. Lecture Notes in Computer Science, vol 5872. Springer, Berlin, Heidelberg. 2009.

KHATTAK, A. M.; LATIF, K.; LEE, S. Change management in evolving web ontologies. **Knowledge-Based Systems**, v. 37, p. 1–18, 2013.

\_\_\_\_. Change management in evolving web ontologies. **Knowledge-Based Systems**, v. 37, n. May, p. 1–18, 2013b.

KIDNEY DISEASE: IMPROVING GLOBAL OUTCOMES (KDIGO) CKD WORK GROUP. KDIGO 2012 Clinical Practice Guideline for the Evaluation and Management of Chronic Kidney Disease. **Kidney inter., Suppl.** 2013; 3: 1–150.

KLEIN, M. Change Management for Distributed ontologies. **Amsterdam: SIKS Dissertation Series**, 2004.

KLEIN, M.; NOY, N. F. A component-based framework for ontology evolution. **Workshop on ontologies and Distributed Systems at IJCAI-03**, Acapulco, Mexico, 2003.



- LASSILA, O.; MCGUINNESS, D. L. **The Role of Frame-Based Representation on the Semantic Web**. LABORATORY, K. S. (Org.). Stanford: Stanford University, 2001. Disponível em: <<http://www.ida.liu.se/ext/epa/ej/etai/2001/018/01018-etaibody.pdf>>. Acesso em: 04 mar 2018.
- LEE, D. *et al.* A survey of SNOMED CT implementations. **Journal of Biomedical Informatics**, v. 46, p. 87–96, 2013.
- MATTOS, M. C.; SIMÕES, P. W. T. A.; FARIAS R. F. A metodologia Methontology na construção de ontologias. **Rev de Inic Cient.** v. 5, n.1, p.1-12, 2010.
- MINISTÉRIO DA SAÚDE. **Protocolo de Encaminhamento da Atenção Básica para a Atenção Especializada** Ministério da Saúde Universidade Federal do Rio Grande do Sul. [s.l: s.n.].
- MORAIS, E. A. M.; AMBRÓSIO, A. P. L. **ontologias: Conceitos, usos, tipos, metodologias, ferramentas e linguagens**. Instituto de Informática, Universidade Federal de Goiás. 2007.
- MOTA, M. R. A. (2013) **Mapeamento sistemático sobre o uso de ontologias em informática médica**. 132 f. Dissertação (Mestrado). Universidade Federal da Paraíba, João Pessoa.
- NOY, N. F.; CHUGH, A.; LIU, W. e MUSEN, M. A. A Framework for ontology Evolution in Collaborative Environments. **International Semantic Web Conference (ISWC)**. 2006.
- NOY, N. F.; MCGUINNESS, D. L. **ontology Development 101: A Guide to Creating Your First ontology**. Stanford: Stanford University, 2001. Disponível em: [https://protege.stanford.edu/publications/ontology\\_development/ontology101-noy-mcguinness.html](https://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html). Acesso em: 04 mar. 2018.
- NOY, N. F. e MUSEN, M. A. PromptDiff: a fixed-point algorithm for comparing ontology versions. **Proceedings of the eighteenth national conference on artificial intelligence (AAAI)**. p. 744–50. 2002.

OCHS, C. *et al.* Summarizing and visualizing structural changes during the evolution of biomedical ontologies using a Diff Abstraction Network. **Journal of Biomedical Informatics**, v. 56, p. 127–144, 2015.

OLIVER, D. E. *et al.* Representation of Change in Controlled Medical Terminologies. **Artificial Intelligence in Medicine**, v. 15, n. 1, p. 53–76, 1999.

PINTO, H. S.; MARTINS, J. P. ontologies: How can They be Built?. **Knowledge and Information Systems**. v. 6, n. 4, p. 441-464, 2004.

PLESSERS, P.; TROYER, O. DE. Resolving inconsistencies in evolving ontologies. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 4011 LNCS, p. 200–214, 2006.

PLESSERS, P.; TROYER, O. DE; CASTELEYN, S. Understanding ontology evolution: A change detection approach. **Web Semantics: Science, Services and Agents on the World Wide Web**, v. 5, n. 1, p. 39–49, mar. 2007.

RAUTENBERG, S.; TODESCO, J. L.; GAUTHIER, Fernando A. O. Processo de desenvolvimento de ontologias: uma proposta de ferramenta. **Rev. Tecnol.** v. 30, n. 1, p. 133-144, 2009.

REIS, J. C. DOS *et al.* Understanding semantic mapping evolution by observing changes in biomedical ontologies. **Journal of Biomedical Informatics**, v. 47, p. 71–82, 2014.

REIS, J. C. DOS *et al.* DyKOSMap: A framework for mapping adaptation between biomedical knowledge organization systems. **Journal of Biomedical Informatics**, v. 55, p. 153–173, 2015.

RICHESSON, R. L.; ANDREWS, J. E.; KRISCHER, J. P. Use of SNOMED CT to Represent Clinical Research Data : A Semantic Characterization of Data Items on Case Report Forms in Vasculitis Research. **Journal of the American Medical Informatics Association**, v. 13, n. 5, p. 536–546, 2006.

ROSSE, C. e MEJINO Jr., J. L. V. A reference ontology for biomedical informatics: The foundational model of anatomy. **Journal of Biomedical Informatics**. v. 36, n. 6, p. 478–500, 2003.

RUSSEL, S. J. e NORVIG, P. **Inteligência Artificial**. 3. ed. Rio de Janeiro: Elsevier, 2013.

SNOMED CT. **SNOMED CT & Other Terminologies, Classifications & Code Systems**. Disponível em: <http://www.snomed.org/snomed-ct/mapping-to-other-terminologies>. Acesso em: 21 Jun. 2017.

SNOMED CT. **What is SNOMED CT?**. Disponível em: <http://www.snomed.org/snomed-ct/what-is-snomed-ct>. Acesso em: 21 Jun 2017.

SOUSA, A. P. DE *et al.* **Evolução de Ontologias em Saúde: Um Estudo de Caso no Domínio da Doença Renal Crônica**. Encontro Nacional de Inteligência Artificial e Computacional. **Anais...**Uberlândia-MG: 2017.

SOWA, J. F. Knowledge Representation: Logical, Philosophical, and Computational Foundations. **Brooks Cole Publishing**. v. 27, n. 2, p. 286-295, 2000.

STOJANOVIC, L. (2004) **Methods and tools for ontology Evolution**. 2004. 249 f. Tese (Doutorado) – Universitaet Fridericiana zu Karlsruhe, Karlsruhe.

STOJANOVIC, L. et al. User-driven ontology evolution management. **Lecture Notes in Computer Science**. v. 6, n.4, p. 285–300, 2002.

STOJANOVIC, L. et al. ontology Evolution as Reconfiguration-Design Problem Solving. **Proceedings of the 2nd international conference on Knowledge capture**. p. 162–171, 2003.

STUDER, R.; BENJAMINS, R. V.; FENSEL, D. Knowledge Engineering: Principles and methods. **Data & Knowledge Engineering**, v. 25, p. 161-197, 1998.

TAVARES, E. A. **Uma abordagem para suporte à decisão clínica baseada em semântica no domínio da nefrologia**. 2016. 211 f. Dissertação (Mestrado em Informática). Universidade Federal da Paraíba, João Pessoa. 2016.

UNA-SUS/UFMA, U. F. DO M. Especialização em Nefrologia Multidisciplinar 1. *In*: **Nefrologia**. [s.l: s.n.]. p. 15–39.

W3C. **Simple part-whole relations in OWL ontologies.** Disponível em: <https://www.w3.org/2001/sw/BestPractices/OEP/SimplePartWhole/>. Acesso em 04 mar. 2018.

W3C. **OWL Web ontology Language.** Disponível em: <https://www.w3.org/TR/owl-features/>. Acesso em 20 ago. 2016.

ZABLITH, F. **Dynamic ontology Evolution.** ISWC 2008 Doctoral consortium. Anais...2008.

## Anexo A

Nesta seção estão dispostos os resultados das consultas SPARQL definidas a partir das QCs da versão da ontoDecideDRC submetida à evolução no estudo de caso por mudança de foco.

Figura 47. Retorno obtido após a execução da consulta SPARQL baseada da QC01 (O que é DRC?)

```
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ontodrc: <http://www.semanticweb.org/ceciliapegado/ontologies/2016/12/ontodrc.owl#>

SELECT ?isDefinedBy
WHERE {
  { ?subject rdfs:isDefinedBy ?isDefinedBy }
  FILTER(?subject = ontodrc:DoencaRenalCronica)
}

isDefinedBy
"A doença Renal Crônica pode ser definida por: - Uma Taxa de filtração glomerular (TFG) normal ou próxima do normal,
```

Fonte: SPARQL query – Protégé

Figura 48. Retorno obtido após a execução da consulta SPARQL baseada da QC02 (O que é TFG?)

```
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ontodrc: <http://www.semanticweb.org/ceciliapegado/ontologies/2016/12/ontodrc.owl#>

SELECT ?isDefinedBy
WHERE {
  { ?subject rdfs:isDefinedBy ?isDefinedBy }
  FILTER(?subject = ontodrc:TaxaDeFiltraçãoGlomerular)
}

isDefinedBy
"Teste para medir a função renal e determinar o estágio de doenças renais"@pt
```

Fonte: SPARQL query – Protégé

Figura 49. Retorno obtido após a execução da consulta SPARQL baseada da QC03 (Quais são as categorias dos níveis de albuminúria?)

```
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ontodrc: <http://www.semanticweb.org/ceciliapegado/ontologies/2016/12/ontodrc.owl#>

SELECT ?subject
WHERE {
  {?subject rdfs:subClassOf ontodrc:CategoriesDosNiveisDeAlbuminuria }
}
```

subject
A2
A3
A1

Fonte: SPARQL query – Protégé

Figura 50. Retorno obtido após a execução da consulta SPARQL baseada da QC04 (Quais as atribuições do Médico da Atenção Primária?)

```
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ontodrc: <http://www.semanticweb.org/ceciliapegado/ontologies/2016/12/ontodrc.owl#>

SELECT ?isDefinedBy
WHERE {
  { ?subject rdfs:isDefinedBy ?isDefinedBy }
  FILTER(?subject = ontodrc:MedicoDaAtencaoPrimaria)
}
```

isDefinedBy
"Médicos que atuam nos cuidados de saúde primários fazendo a intervenção precoce na história natural das doenças potenciadas"

Fonte: SPARQL query – Protégé

Figura 51. Retorno obtido após a execução da consulta SPARQL baseada da QC05 (O Médico da Atenção Primária trata todos os estágios da DRC?)

SPARQL query:	
<pre>PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; PREFIX owl: &lt;http://www.w3.org/2002/07/owl#&gt; PREFIX rdfs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX xsd: &lt;http://www.w3.org/2001/XMLSchema#&gt; PREFIX ontodrc: &lt;http://www.semanticweb.org/ceciliapegado/ontologies/2016/12/ontodrc.owl#&gt;  ASK{   {?condutas rdfs:subClassOf ontodrc:Conduta }   {?condMAP owl:onProperty ontodrc:ERealizadoPor }   {?condMAP owl:someValuesFrom ontodrc:MedicoDaAtencaoPrimaria }   FILTER(?condutas = ?condMAP)}</pre>	
	Result
False	

Fonte: SPARQL query – Protégé

Figura 52. Retorno obtido após a execução da consulta SPARQL baseada da QC06 (Quem deve conduzir a DRC G2?)

SPARQL query:	
<pre>PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; PREFIX owl: &lt;http://www.w3.org/2002/07/owl#&gt; PREFIX rdfs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX xsd: &lt;http://www.w3.org/2001/XMLSchema#&gt; PREFIX ontodrc: &lt;http://www.semanticweb.org/ceciliapegado/ontologies/2016/12/ontodrc.owl#&gt;  SELECT DISTINCT ?conduta WHERE {   { ?conduta rdfs:subClassOf ontodrc:CondutaEstagio2 }   { ?restriction rdf:type owl:Restriction }   { ?restriction owl:onProperty ontodrc:ERealizadoPor } }</pre>	
	conduta
● ERealizadoPor some MedicoDaAtencaoPrimaria	

Fonte: SPARQL query – Protégé

Figura 53. Retorno obtido após a execução da consulta SPARQL baseada da QC08 (Como conduzir o estágio 1 da DRC)

```
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ontodrc: <http://www.semanticweb.org/ceciliapegado/ontologies/2016/12/ontodrc.owl#>

SELECT ?isDefinedBy
WHERE {
  { ?subject rdfs:isDefinedBy ?isDefinedBy }
  FILTER(?subject=ontodrc:CondutaEstagio1)
}

isDefinedBy
"Tratamento dos fatores de risco modificáveis de progressão da DRC e doença cardiovascular de acordo com as recomendações"
```

Fonte: SPARQL query – Protégé

Figura 54. Retorno obtido após a execução da consulta SPARQL baseada da QC09 (Como conduzir o estágio 3A da DRC?)

```
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ontodrc: <http://www.semanticweb.org/ceciliapegado/ontologies/2016/12/ontodrc.owl#>

SELECT ?isDefinedBy
WHERE {
  { ?subject rdfs:isDefinedBy ?isDefinedBy }
  FILTER(?subject=ontodrc:CondutaEstagio3A)
}

isDefinedBy
"Tratamento dos fatores de risco modificáveis de progressão da DRC e doença cardiovascular de acordo com as recomendações"
```

Fonte: SPARQL query – Protégé



## Anexo B

Nesta seção estão dispostos os resultados das consultas SPARQL definidas a partir das QCs da versão da ontoDecideDRC submetida à evolução no estudo de caso por mudanças em suas diretrizes de domínio.

Figura 55. Retorno obtido após a execução da consulta SPARQL baseada da QC03 (Como conduzir a DRC G5?)

```
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ontodecidedrc: <http://www.semanticweb.org/esaualeucar/ontologies/2015/9/OntoDecideDRC#>

SELECT ?conduta5n ?conduta5nd
WHERE { { ?subject ontodecidedrc:Guideline ?conduta5n .
FILTER(?subject=ontodecidedrc:Estagio5D)}
UNION
{ ?subject ontodecidedrc:Guideline ?conduta5nd .
FILTER(?subject=ontodecidedrc:Estagio5ND)}
}

conduta5n
"- Recomenda-se medir os níveis de cálcio, fósforo, PTH e fosfatase alcalina. - Manter os valores de fósforo

conduta5nd
"- Manter os valores de fósforo normais - Realizar mensalmente os exames creatinina, ureia, cálcio, fósforo
```

Fonte: Adaptado do SPARQL query – Protégé

Figura 56. Retorno obtido após a execução da consulta SPARQL baseada da QC04 (Quais os estágios da DRC tratados pelo Nefrologista?)

```
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ontodecidedrc: <http://www.semanticweb.org/esaualeucar/ontologies/2015/9/OntoDecideDRC#>

SELECT DISTINCT ?entidade ?restriction
WHERE {
?entidade rdfs:subClassOf ?restriction .
?restriction rdf:type owl:Restriction .
?restriction owl:onProperty ontodecidedrc:ETratadoPor .
?restriction owl:someValuesFrom ontodecidedrc:Nefrologista .
}
GROUP BY ?restriction ?entidade
ORDER BY ?entidade
```

entidade	restriction
Estagio3B	● ETratadoPor some Nefrologista
Estagio4	● ETratadoPor some Nefrologista
Estagio5D	● ETratadoPor some Nefrologista
Estagio5ND	● ETratadoPor some Nefrologista

Fonte: SPARQL query – Protégé

Figura 57. Retorno obtido após a execução da consulta SPARQL baseada da QC05 (Como conduzir a DRC G4?)

```
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ontodecidedrc: <http://www.semanticweb.org/esaualencar/ontologies/2015/9/OntoDecideDRC#>

SELECT ?conduta4
WHERE { { ?subject ontodecidedrc:Guideline ?conduta4 .
FILTER(?subject=ontodecidedrc:Estagio4)}
}
```

conduta4
"- Manter os valores de fósforo normais. - Se o paciente for Hipertenso e Diabético e RAC for menor que 30 mg/g manter a Pre

Fonte: SPARQL query – Protégé

Figura 58. Retorno obtido após a execução da consulta SPARQL baseada da QC06 (Quem são integrantes da equipe multiprofissional de atenção especializada em DRC?)

```
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ontodecidedrc: <http://www.semanticweb.org/esaualencar/ontologies/2015/9/OntoDecideDRC#>

SELECT ?subject
WHERE {
{ ?subject rdfs:subClassOf ontodecidedrc:EquipeMultiprofissionalDeAtencaoEspecializadaEmDRC }
}
|
```

subject
Nefrologista
Enfermeiro
Nutricionista
Psicologo
AssistenteSocial

Fonte: SPARQL query – Protégé

Figura 59. Retorno obtido após a execução da consulta SPARQL baseada da QC07 (Quais as atribuições da equipe multiprofissional de atenção especializada em DRC?)

```
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ontodecidedrc: <http://www.semanticweb.org/esaualeucar/ontologies/2015/9/OntoDecideDRC#>

SELECT ?isDefinedBy
WHERE {
  { ?subject rdfs:isDefinedBy ?isDefinedBy }
  FILTER(?subject=ontodecidedrc:EquipeMultiprofissionalDeAtencaoEspecializadaEmDRC)
}

isDefinedBy
"Compete à Equipe Multiprofissional de Atenção Especializada em DRC: I - constituir preferencialmente referência regional par
```

Fonte: SPARQL query – Protégé

Figura 60. Retorno obtido após a execução da consulta SPARQL baseada da QC08 (Como conduzir o estágio 3 da DRC)

```
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ontodecidedrc: <http://www.semanticweb.org/esaualeucar/ontologies/2015/9/OntoDecideDRC#>

SELECT ?conduta3a ?conduta3b
WHERE {
  { ?subject ontodecidedrc:Guideline ?conduta3a .
  FILTER(?subject=ontodecidedrc:Estagio3A)}
  UNION
  { ?subject ontodecidedrc:Guideline ?conduta3b .
  FILTER(?subject=ontodecidedrc:Estagio3B)}
}

conduta3a
"- Se o paciente for Hipertenso e Diabético e rac for menor que 30 mg/g manter a Pressão Arterial menor q

conduta3b
"- Manter os valores de fósforo normais. - Se o paciente for Hipertenso e Diabético e RAC for menor que 30
```

Fonte: Adaptado do SPARQL query – Protégé