Uso de ontologias modulares para classificação de pacientes portadores da *Síndrome de Li-Fraumeni*: Estudo de caso no *A.C. Camargo Cancer Center*.

Ricardo Moura Sekeff Budaruiche

Tese apresentada
ao
Instituto de Matemática e Estatística
da
Universidade de São Paulo
para
obtenção do título
de
Doutor em Ciências

Programa: Ciência da Computação

Orientador: Prof[a]. Dr[a]. Renata Wassermann

São Paulo, 22 de Fevereiro 2016

Uso de ontologias modulares para classificação de pacientes portadores da *Síndrome de Li-Fraumeni*: Estudo de caso no *A.C. Camargo Cancer Center*

Esta é a versão original da dissertação/tese elaborada pelo candidato Ricardo Moura Sekeff Budaruiche, tal como submetida à Comissão Julgadora.

Agradecimentos

Qualquer caminho que se percorra, não importando a distância ou o tamanho da carga que se leve, parecerá menor e menos penoso quando estamos acompanhados.

Agradeço, antes de tudo, a Deus, por me proporcionar todas as condições necessárias a essa caminhada.

À minha família, que sempre me incentivou e me deu total apoio para que conseguíssemos chegar até aqui. Aos meus pais, Carlos Alberto e Lúcia de Fátima, por toda a força, pelo amor incondicional e por ser nosso porto-seguro em diversos momentos dessa caminhada. À minha amada esposa (minha metade da laranja), Juliana Ferreira, companheira de todas as horas. Obrigado por todo o seu amor, pela sua paciência e compreensão durante minhas ausências, frustrações e momentos de fraqueza. Nessas horas, você sempre esteve comigo, sofrendo e festejando, segurando a minha mão e me ajudando a caminhar. Saiba que essa conquista também é sua! Às minhas filhas, Ana Luísa e Lara, razão de tudo isso, que, tão pequenininhas, e já souberam enfrentar bem todas as dificuldades e asperezas que um doutorado impõe aos seus alunos. Obrigado por todas as palavras de carinho, por todas as declarações e por todos os sorrisos que ganhei de vocês quando eu estive triste. Aos meus irmãos Eduardo e Fernanda, pela força e pelos momentos leves de diversão. À minha cunhada Roberta Ferreira, pela sua compreensão quando não podia dar a atenção necessária. Vocês todos têm parte nessa conquista.

Agradeço à minha professora e orientadora Renata Wassermann pela forma generosa como ela me acolheu aqui no IME, e pelos valorosos conselhos e orientações. Ao Diogo Patrão, pelas oportunidades concedidas durante minha passagem pela Informática Médica do A.C. Camargo, por todas as críticas construtivas que fez ao meu trabalho, pelos momentos de descontração no CIPE e pelos conselhos que recebi. Aos amigos de trabalho da Informática Médica no CIPE, que tornaram essa caminhada menos dura e solitária, em especial ao Felipe Massicano, Marcelo Sagayama, Michel Oleynik, Aline Damascena, Guilherme de Oliveira, Fábio Rampazzo, Renato Puga e Leandro Lima.

Aqui no IME, fiz amizades para toda a vida e, por isso, não poderia deixar de agradecer a todos: Eduardo Cotrin, Paulo de Tarso, Viviane Menezes, Henrique Bustamante, Fábio Franco, Ricardo Augusto, Ignasi Franco, Ricardo Herrmann, Fabiano Luz, Thiago Bueno, Esdras Bispo, Robson Feitosa, Thiago Dias, Erika Guetti, Filipe Resina, Luciano Kelvin, Leticia Gindri, Bosco Pereira, Raphael Cobe, Yuri Santos, Viviane Bonadia, Ricardo Guimarães, José David e Rafael Lobato, além de todos aqueles que, direta ou indiretamente, contribuíram para deixar esses anos de doutorado mais divertidos.

Aos amigos da república Silvio (Punk) Fiorentin, Adalberto (Mirassol) Alício, Kamila Maguerroski, Bruno (Carioca) Ferreira e Mario Guilherme Pedroni, pelas churrascadas, pela acolhida e por terem sido tão amigos.

Por fim, aos meus coordenadores Francisco José de Araújo (Chiquinho) e Harilton Araújo, por

terem sido compreensivos e parceiros nos momentos de ausência para cumprir com as obrigações do doutorado, e aos meus alunos, que também foram compreensivos quando não pude render em sala de aula tudo aquilo que eles esperavam de mim.

Resumo

BUDARUICHE, R. M. S. **Uso de ontologias modulares para classificação de pacientes portadores da Síndrome de *Li-Fraumeni*: Estudo de caso no *A.C. Camargo Cancer Center***. 2016. 163f. Tese (Doutorado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2016.

O crescimento do poder computacional nas últimas décadas e o avanço da Inteligência Artificial para além dos limites da Ciência da Computação tem permitido que diversas outras áreas da ciência sejam beneficiadas com novas descobertas e avanços auxiliados por essas ferramentas computacionais. Uma dessas ferramentas, que é considerada multidisciplinar, apesar de ter sua origem na Filosofia, são as Ontologias, utilizadas para organizar conhecimentos adquiridos ao longo do tempo e deduzir novos conhecimentos por meio de técnicas dedutivas. Neste trabalho, apresentamos a *Li-Fraumeni Ontology*, uma ontologia modular de aplicação para a *Síndrome de Li-Fraumeni* desenvolvida para classificar pacientes de uma família segundo os quatro critérios da síndrome: *Classic*, Birch, Eeles e Chompret. Para isso, desenvolvemos uma metodologia ágil e colaborativa de modelagem de ontologias que permitiu a construção gradual e estável dos módulos da *Li-Fraumeni Ontology* sem que houvesse gasto de tempo com documentações extensas ou longas correções na ontologia. A *Li-Fraumeni Ontology* é formada por 3 módulos: GenOnto, CDOnto e LFOnto, este último podendo ser substituído por outra ontologia que modele o domínio de conhecimento para outra doença de caráter hereditário. Demonstramos essa capacidade modelando uma ontologia para a Síndrome de *Lynch* e mostrando que ela pode ser usada em conjunto com módulos GenOnto e CDOnto. Mapeamos a *Li-Fraumeni Ontology* à BioTopLite, uma ontologia *upper-level*, e que servirá de referência para futuras integrações com ontologias de outros domínios. Por fim, testamos o poder da Li-Fraumeni Ontology em dois grupos de famílias: um grupo formado por casos fictícios gerados aleatoriamente por uma ferramenta desenvolvida para para essa finalidade (OpenGLiFS) e outro grupo formado por casos reais do A.C. Camargo Cancer Center. Os resultados foram analisados e mostraram que a Li-Fraumeni Ontology pode ser utilizada de maneira satisfatória e confiável para classificação de pacientes e para a descoberta de novos fatores que venham a influenciar a Síndrome de Li-Fraumeni.

Palavras-chave: ontologia, Síndrome de Li-Fraumeni, modelagem de ontologias, Li-Fraumeni Ontology, aquisição de conhecimento, ontologia modular.

Abstract

BUDARUICHE, R. M. S. **Using modular ontologies for Li-Fraumeni Syndrome patients classification: A case study at A.C. Camargo Cancer Center.**. 2016. 163f. Thesis (Doctoral) - Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2016.

The growth of computational power in recent decades and the progress of Artificial Intelligence beyond the boundaries of Computer Sciences have allowed several other areas of science being benefited with new discoveries and improvements aided by these computational tools. One such tool, which is considered multidisciplinary despite having its origin Philosophy, are the ontologies, which are used to organize knowledge acquired over time and deduce new knowledge through deductive techniques. In this work, we present the Li-Fraumeni Ontology, a modular ontology application for Li-Fraumeni Syndrome developed to classify patients in a family according to the four syndrome criteria: Classic, Birch, Eeles and Chompret. To do so, we developed an agile and collaborative methodology of ontology modeling which allowed the gradual and stable construction of the modules of the Li-Fraumeni Ontology without spending time with extensive documentation and lengthy corrections in the ontology. The Li-Fraumeni Ontology consists of three modules: GenOnto, CDOnto and LFOnto, the latter being able to be replaced by another ontology that models the knowledge domain to another hereditary disease. We demonstrate this capability by modeling an ontology for Lynch Syndrome and showing that it can be used together with the GenOnto and CDOnto modules. We mapped the Li-Fraumeni Ontology to BioTopLite. an upper-level ontology, which will serve as reference for future integrations with ontologies from other domains. Finally, we tested the power of the Li-Fraumeni families Ontology with two family groups: one group of fictitious cases randomly generated by a tool developed for this purpose (OpenGLiFS) and another group of real cases of A.C. Camargo Cancer Center. The results were analyzed and showed that the Li-Fraumeni Ontology can be used satisfactorily and reliably to patient classification and discovery of new factors that may influence the Li-Fraumeni Syndrome.

Keywords: ontology, Li-Fraumeni Syndrome, ontology modeling, Li-Fraumeni Ontology, knowledge aquisition, modular ontology.

Sumário

Lista de Abreviaturas e Siglas

ABox	*Assertion Box*: Conjunto de assertivas verdadeiras sobre um determinado domínio.
CDOnto	Clinical Data Ontology (ontologia de dados clínicos)
CID-10	Classificação Internacional de Doenças versão 10
CID-O	Classificação Internacional de Doenças Oncológicas
CWA	Closed-World Assumption (Assunção de Mundo Fechado)
EHR	*Electronic Health Record - Sistema de Prontuário Eletrônico do Paciente*
GenOnto	Genealogy Ontology (ontologias de relações familiares)
IARC	*International Agency for Research on Cancer*
LFL	*Li-Fraumeni Like*
LFOnto	Li-Fraumeni Ontology (ontologia da Síndrome de Li-Fraumeni)
LFS	*Li-Fraumeni Syndrome*
OWA	Open-World Assumption (Assunção de Mundo Aberto)
OWL	*Web Ontology Language*
SNOMED-CT	*Systematized Nomenclature of Medicine–Clinical Terms*
SWRL	Linguagem utilizada para escrita de regras DL-*Safe* em arquivos OWL.
TBox	*Terminological Box*: Conjunto de regras e terminologias sobre um determinado domínio.

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 Motivação

A sociedade atual vem observando, ultimamente, um crescimento enorme no volume de dados gerados a cada ano. De acordo com a IBM [1], apenas em 2012, foram gerados aproximadamente 2.5 exabytes[2] de dados por dia em diversas áreas de atividade. Na medicina e na bioinformática, a computação tem se mostrado cada vez mais presente, quer seja assistindo aos profissionais da área, quer seja manipulando e armazenando dados e informações, auxiliando na geração e na gestão de conhecimento em hospitais e centros de pesquisa. Apesar dos benefícios trazidos por esse grande volume de dados (maior granularidade das informações, maior quantidade de fontes de dados, além de outros), manipular e extrair informações e conhecimentos dessas fontes têm se apresentado como desafios nas áreas da Computação e da Inteligência Artificial em razão das diferenças estruturais dos dados, da heterogeneidade semântica e da integração das diversas fontes de dados existentes [Suj01].

Segundo [Gru93], *ontologias são especificações explícitas de uma conceitualização*. Elas têm se difundido muito em áreas como a medicina, biomedicina e a computação. Segundo Smith (2004) *apud* [Gui05], o termo ontologia foi utilizado em ciência da computação pela primeira vez em 1967. A partir dessa época, seu uso aumentou sobremaneira, sendo responsáveis diretos por esse aumento, as áreas de banco de dados e sistemas de informação, engenharia de software e inteligência artificial (Smith & Welty (2001) *apud* [Gui05]). O uso concomitante das ontologias com outras tecnologias, como inferência lógica, armazenamento de dados e algoritmos de mineração de dados, podem potencializar os resultados positivos desejados, na medida em que contribuem com definições e relacionamentos entre os termos do domínio de estudo, resultando num processo conhecido como *descoberta do conhecimento*. Com esse processo, é possível extrair conceitos e conhecimentos implícitos no modelo conceitual e que não foram explicitamente descritos, através do uso dos **sistemas baseados em conhecimento**. Segundo [RN09], diversos domínios de estudo, como a bioinformática e a biomedicina, podem ser beneficiados pelo uso desses sistemas. Entretanto, construir e manter sistemas baseados em conhecimento acarreta em alto custo operacional devido ao uso de hardwares de alto desempenho (grande capacidade de memória, processadores de excelente desempenho, etc) e também pelo uso de mão de obra altamente especializada (especialistas em domínios específicos, em ontologias e em sistemas baseados em conhecimento).

[1] http://www-01.ibm.com/software/data/bigdata/what-is-big-data.html
[2] 1 exabyte = 1 milhão de terabytes.

Nesse cenário, o uso da computação com o objetivo de auxiliar na busca por conhecimentos na área médica tem se tornado cada vez mais frequente. O avanço científico em matérias como ontologia, inteligência artificial, armazenamento de dados e computação distribuída permitem que os pesquisadores se lancem em áreas antes deixadas em segundo plano por falta de capacidade computacional para o processamento dos dados. Na área biomédica, o uso de ontologias tem auxiliado muito na representação de conceitos que podem formar uma **rede semântica** ([HDG12]) para auxiliar, por exemplo, na descoberta de novos conhecimentos e na recuperação de dados para pesquisas médicas.

De fato, o uso das ontologias nestas áreas vem crescendo recentemente e, como resultado, têm surgido diversas aplicações que fazem uso de técnicas de inferência automatizada para classificar dados e extrair novos conhecimentos. Consequentemente, novas ontologias foram surgindo ao longo do tempo (Gene Ontology - GO[3], NCI Thesaurus - NCIT[4], SNOMED-CT[5]) e, com isso, o conhecimento extraído por meio dessas soluções foi ficando fragmentado. A integração dessas ontologias permite que esse conhecimento acumulado separadamente seja utilizado em conjunto formando uma base de conhecimentos mais ampla. Tornou-se necessário, então, a criação de ontologias que estabelecessem regras para essa integração, sendo chamadas de ***Upper-Level Ontologies*** (ULO) [Gua98, MCBV12]. Essas ontologias definem conceitos e propriedades mais amplos do que aqueles definidos nas ontologias de aplicação, estabelecendo um consenso entre as partes que desejam utilizar conceitos comuns à respeito de um determinado domínio.

Não obstante o crescente uso das ontologias como ferramenta de gerenciamento do conhecimento, muitos autores ainda divergem no tocante à avaliação da qualidade das ontologias. Alguns autores, como Guarino & Welty [GW02], Ensan & Weichang [ED13] e Sicilia *et al.* [SRGBSA12], apresentam metodologias baseadas em métricas coletadas diretamente das ontologias, como **compreensão** (quão fácil é a compreensão dessa ontologia por uma outra pessoa?), **usabilidade** (quão fácil é a compreensão de parte dessa ontologia de tal forma que ela possa ser reutilizada na modelagem de outro domínio do conhecimento?) e **performance na inferência** (quão eficiente é a resposta a consultas a essa ontologia?), dentre outros.

Este trabalho propõe a construção de uma **ontologia modular** de aplicação para o domínio de conhecimento da Síndrome de *Li-Fraumeni* que possa ser utilizada em uma base de dados integrada com dados de pacientes e famílias oriundos de vários bancos de dados do *A.C. Camargo Cancer Center*. Propõe-se, também, o uso de motores de inferência para a classificação dos pacientes segundo os quatro critérios clínicos da Síndrome de *Li-Fraumeni* (***Classic***, **Chompret**, **Eeles** e **Birch**), e para a aquisição de novos conhecimentos sobre a síndrome através da análise dos resultados dessa classificação. Durante a construção desta solução, notou-se a viabilidade de integração da mesma com uma ULO, como a OBO e a BioTopLite. Sugeriu-se, então, uma integração com uma dessas ontologias a fim de permitir seu reuso posterior em outro domínio do conhecimento. Por fim, propõe-se, também, a expansão do uso da **ontologia modular** desenvolvida para outras síndromes de câncer de caráter hereditário, como a Síndrome de *Lynch*[6].

[3] http://geneontology.org/
[4] https://ncit.nci.nih.gov/ncitbrowser/
[5] http://www.ihtsdo.org/
[6] A Síndrome de *Lynch* é decorrente de uma alteração genética que aumenta o risco de desenvolvimento de tumores no cólon e no reto. Fonte: http://www.accamargo.org.br/saude-prevencao/artigos/a-sindrome-de-lynch-e-sua-relacao-com-o-cancer-colorretal/158/

1.2 Objetivos

Esta tese de doutorado norteia-se na hipótese de que é possível proporcionar a classificação, de forma automática e mais precisa, de pacientes portadores da Síndrome de *Li-Fraumeni*, ou que se supõe ser portadores. Buscamos, com isso, um modelo conceitual, uma ontologia de aplicação para a Síndrome de *Li-Fraumeni*, que vai ajudar o médico/pesquisador na seleção de pacientes para estudos clínicos.

Para tanto, seguimos os objetivos parciais abaixo:

1. Propor uma arquitetura de **ontologia modular** capaz de extrair, por meio de um motor de inferência, conhecimento de uma base de dados `rdf`;
2. Construção de uma ontologia para armazenar conhecimento sobre graus de parentesco em árvores genealógicas;
3. Construção de uma ontologia para armazenar conhecimento sobre dados clínicos dos pacientes, como diagnóstico, documentos, idade do paciente no primeiro diagnóstico, etc;
4. Construção de uma ontologia para armazenar conhecimento sobre a Síndrome de *Li-Fraumeni*;
5. Validar a **ontologia modular** através da avaliação de critérios de qualidade que serão apresentados;
6. Propor uma integração das ontologias construídas com a ULO BioTopLite a fim de permitir seu reuso posterior em outro domínio do conhecimento;
7. Apresentar uma proposta de expansão para reuso em outro domínio de conhecimento para câncer de origem hereditária;

Cada um dos objetivos parciais acima sugerem uma descrição mais detalhada sobre como cada um será cumprido. Apresentamos, assim, uma proposta de atividades (*guidelines*) a serem executadas:

1. Mapear os dados específicos para a modelagem de graus de parentesco a partir de uma ontologia de domínio já existente;
2. Mapear os dados clínicos necessários para a construção de uma ontologia de aplicação para a Síndrome de *Li-Fraumeni*;
3. Formalizar as regras para cada um dos critérios *Li-Fraumeni* da ontologia e escreve-las em um formato padronizado (SWRL ou OWL);
4. Criar casos de testes **fictícios e reais** a fim de validar as regras de inferência e, consequentemente, os resultados da classificação de pacientes nesses casos base.
5. Identificar os conceitos na *Li-Fraumeni Ontology* que serão mapeados para a ULO;
6. Apresentar uma proposta de síndrome, diferente da Síndrome de *Li-Fraumeni*, que possa servir como prova da viabilidade de expansão destas ontologias.

Em seguida, vamos utilizar um estudo de caso do *A.C. Camargo Cancer Center* para validar as ontologias de aplicação e avaliar sua qualidade (oportunamente, este trabalho explicará os critérios de qualidade utilizados durante a avaliação das ontologias). Além disso, vamos utilizar uma solução desenvolvida no Departamento de Informática Médica do *A.C. Camargo Cancer Center* que mapeia os dados familiares e de pacientes, presentes nas diversas bases de dados, para uma *triplestore*, e que será posteriormente descrita. Pretendemos mostrar, também, que a ontologia proposta pode ser estendida para outros tipos de doenças cujas origens residem em fatores hereditários.

1.3 Justificativa e Contribuições

Esta tese de doutorado se justifica pelos seguintes aspectos:

- Não existe uma ontologia de aplicação específica para a Síndrome de *Li-Fraumeni*, muito menos que modele os conceitos a respeito dos critérios clínicos usados para classificar os pacientes portadores da síndrome em questão. As ontologias existentes nos repositórios pesquisados são apenas dicionários ou tesauros para os termos e conceitos da síndrome e não permitem que sejam realizadas inferências para a descoberta de novos conhecimentos, pois não foram modeladas para esse fim. Assim, este trabalho propõe a construção de uma **ontologia modular** para a Síndrome de *Li-Fraumeni* compatível com o uso de técnicas de inferência automáticas;

- O uso desta **ontologia modular** no estudo clínico da Síndrome de *Li-Fraumeni* pode contribuir para a descoberta de novos critérios clínicos, de novos classificadores para a síndrome ou até mesmo para a validação dos já existentes. Novos conhecimentos não explícitos podem ser descobertos, como por exemplo, se algum critério clínico (como o critério de *Eeles*) está contido em outro critério clínico (critério de *Chompret*, por exemplo);

- A possibilidade de modelagem de outras ontologias para outros domínios de síndromes de caráter hereditários é mais uma contribuição deste trabalho. Propomos, ao final desta tese, a expansão das ontologias aqui desenvolvidas para a inclusão de uma ontologia referente à Síndrome de *Lynch*, que é uma doença de caráter hereditário e que pode causar tumores malignos no cólon e no reto.

1.4 Organização deste Trabalho

Esta tese se estrutura da seguinte forma: no Capítulo 2, serão abordados todos os conceitos básicos essenciais ao entendimento da proposta deste trabalho. Será contextualizado o cenário do Câncer no Brasil e no mundo, bem como os mecanismos de formação e prevenção da doença. Ainda neste capítulo serão abordados temas centrais como conceito e classificação das ontologias; as ULOs na área biomédica, como a BioTopLite e a OBO; as ontologias familiares e os obstáculos na modelagem desse tipo de ontologia; a importância das linguagens usadas para escrever ontologias formais e a sua evolução; as ontologias usadas como dicionários para codificar doenças; e os critérios de avaliação de qualidade para ontologias. No Capítulo 3 será apresentado o projeto **Ontofamily**, projeto criado para a integração de dados clínicos sobre histórico familiar de pacientes através do uso de ontologias com o objetivo de inferir e extrair conhecimento sobre a Síndrome de *Li-Fraumeni* no *A.C. Camargo Cancer Center*. Um dos resultados do projeto foi a criação de uma base de dados

de pacientes e famílias, obtido através da integração das diversas fontes de dados do hospital, que foi usada como *ABox* para a **ontologia modular** proposta nesta pesquisa. No Capítulo 4, apresentamos uma análise de algumas metodologias para a construção de ontologias. Observamos que algumas características que julgamos importantes para a construção de uma ontologia de forma colaborativa não estão presentes nas metodologias avaliadas e, por essa razão, propomos uma nova metodologia para a construção da *Li-Fraumeni Ontology*. Em seguida, no Capítulo 5, será apresentada a construção das ontologias modulares GenOnto, CDOnto e LFOnto. Será apresentada uma proposta de integração com a ULO BioTopLite. Serão discutidos os aspectos dessa integração bem como os pontos positivos e negativos. No Capítulo 6 serão apresentados os resultados da classificação dos pacientes segundo os critérios *Li-Fraumeni*. Apresentamos as ferramentas OpenGLiFS e Directed-Extract-LiFraumeni, utilizadas para geração de casos de teste e classificação das famílias, respectivamente. No Capítulo 7 serão discutidos os resultados alcançados após a classificação das famílias de teste e reais. Abordaremos a importância da metodologia modular na construção da *Li-Fraumeni Ontology* bem como mostraremos que esses resultados foram satisfatórios à luz do consumo de recursos computacionais e de parâmetros como precisão, acurácia e sensibilidade. Por fim, o Capítulo 8 apresenta uma discussão geral do uso da ontologia modular para resolver o problema da classificação de pacientes segundo os critérios da Síndrome de *Li-Fraumeni*. Será discutido, também, uma proposta de expansão da ontologia para a classificação de pacientes portadores da Síndrome de *Lynch*.

Capítulo 2

Fundamentação Teórica

Neste capítulo trataremos dos fundamentos teóricos pertinentes a este trabalho. Na seção 2.1 serão abordados os mecanismos de aparecimento e disseminação do câncer. A seção abordará também os tipos de câncer, o mecanismo de controle de crescimento e de divisão celular, bem como as características inerentes ao câncer de natureza hereditária. Uma breve discussão sobre o cenário atual no combate ao câncer, no Brasil e no Mundo encerram esta seção. Na seção 2.2, serão apresentados os conceitos e classificações das ontologias, desde a sua abordagem filosófica até o seu uso na ciência da computação. A seção aborda, também, a importância das ontologias médicas/biomédicas e das ULO na construção das ontologias de aplicação, além das ontologias de histórico familiares, o estado da arte dessas ontologias e todos os obstáculos para a construção de uma taxonomia abrangente e, ao mesmo tempo, computável. Na seção 2.3, será tratado o problema da integração de bases de dados heterogêneas e dos obstáculos inerentes à sua implantação. Na seção 2.2.6, serão apresentados os esforços da comunidade médica e científica na criação de ontologias médicas e biomédicas que possam ser utilizadas na descoberta de novos conhecimentos científicos ou apenas na modelagem de novos domínios de conhecimento. A seção fará um breve resumo das ULOs BioTopLite (2.2.6) e OBO-Reference Ontology (2.2.6) e discutirá sua importância na criação de novas ontologias. A seção 2.4 trará os esforços empreendidos na direção de estabelecer um conjunto de métricas que possam avaliar a qualidade de uma ontologia. Apresentaremos como essas métricas foram utilizadas na construção da nossa solução. Por fim, a seção 2.2.6 abordará o sistema de classificação de doenças CID (e das suas subdivisões), sua origem, suas características e sua influencia na construção da nossa solução final.

2.1 Câncer

Em qualquer país do mundo, o câncer é uma doença considerada de difícil tratamento e encarada como um problema de saúde pública de alta prioridade. Segundo publicação do Ministério da Saúde[1], muito se tem investido em pesquisas no tratamento e na descoberta das causas dos diversos tipos de câncer. Apesar disso, ela ainda é uma das doenças que mais mata no mundo, resultando em mais de 8 milhões de óbitos por ano (13% do total de óbitos no mundo) e com uma tendência de crescimento de 70% nas próximas duas décadas [TBS+15]. No Brasil, segundo dados do INCA (Instituto Nacional

[1] A situação do câncer no Brasil / Ministério da Saúde, Secretaria de Atenção à Saúde, Instituto Nacional de Câncer, Coordenação de Prevenção e Vigilância. INCA, 2006. Fonte: http://bvsms.saude.gov.br/bvs/publicacoes/situacao_cancer_brasil.pdf

de Câncer José Alencar Gomes da Silva), a expectativa de incidência de câncer para o biênio 2014/2015 é de mais de 500.000 novos casos[2] (Figura 2.1). Essa estimativa foi realizada levando-se em conta dados dos RCBP[3], alimentados por meio de uma rede de 283 Registros Hospitalares de Câncer e, também, de dados do Sistema de Informações sobre Mortalidade. A **taxa bruta** é a razão entre a taxa bruta de incidência de câncer e a taxa bruta de mortalidade. A taxa bruta de incidência de câncer é o número total de novas ocorrências de determinado tipo de câncer, no caso de todas as neoplasias, soma-se o número de novos casos de todos os tipos de câncer, por cada cem mil habitantes[4] em um determinado espaço geográfico, no caso, o Brasil. A taxa bruta de mortalidade é o número total de óbitos por cada mil habitantes[5] em um determinado espaço geográfico, no caso, o Brasil. O mesmo conceito vale para a taxa bruta de incidência. Para maiores detalhes sobre a metodologia utilizada pelo estudo, fórmula de cálculos e outros conceitos, recomenda-se a leitura do estudo original[6].

Figura 2.1: *Número de casos de câncer por tipo, consolidado para todo o Brasil. Previsão para o biênio 2014/2015. Fonte: INCA http://www.inca.gov.br/estimativa/2014/*

Localização Primária da Neoplasia Maligna	Estimativa dos Casos Novos							
	Homens				Mulheres			
	Estados		Capitais		Estados		Capitais	
	Casos	Taxa Bruta	Casos	Taxa Bruta	Casos	Taxa Bruta	Casos	Taxa Bruta
Próstata	68.800	70,42	17.540	82,93	-	-	-	-
Mama Feminina	-	-	-	-	57.120	56,09	19.170	80,67
Colo do Útero	-	-	-	-	15.590	15,33	4.530	19,20
Traqueia, Brônquio e Pulmão	16.400	16,79	4.000	18,93	10.930	10,75	3.080	13,06
Cólon e Reto	15.070	15,44	4.860	22,91	17.530	17,24	5.650	23,82
Estômago	12.870	13,19	2.770	13,07	7.520	7,41	2.010	8,44
Cavidade Oral	11.280	11,54	2.220	10,40	4.010	3,92	1.050	4,32
Laringe	6.870	7,03	1.460	6,99	770	0,75	370	1,26
Bexiga	6.750	6,89	1.910	8,91	2.190	2,15	730	2,97
Esôfago	8.010	8,18	1.460	6,76	2.770	2,70	540	0,00
Ovário	-	-	-	-	5.680	5,58	2.270	9,62
Linfoma de Hodgkin	1.300	1,28	410	5,72	880	0,83	420	8,64
Linfoma não Hodgkin	4.940	5,04	1.490	6,87	4.850	4,77	1.680	7,06
Glândula Tireoide	1.150	1,15	470	1,76	8.050	7,91	2.160	9,08
Sistema Nervoso Central	4.960	5,07	1.240	5,81	4.130	4,05	1.370	5,81
Leucemias	5.050	5,20	1.250	5,78	4.320	4,24	1.250	5,15
Corpo do Útero	-	-	-	-	5.900	5,79	2.690	11,24
Pele Melanoma	2.960	3,03	950	4,33	2.930	2,85	1.150	4,57
Outras Localizações	37.520	38,40	9.070	42,86	35.350	34,73	8.590	36,49
Subtotal	**203.930**	**208,77**	**51.100**	**241,30**	**190.520**	**187,13**	**58.710**	**248,46**
Pele não Melanoma	98.420	100,75	19.650	92,72	83.710	82,24	22.540	95,26
Todas as Neoplasias	**302.350**	**309,53**	**70.750**	**334,08**	**274.230**	**269,35**	**81.250**	**343,85**

O aumento na incidência de novos casos se deve em muito ao envelhecimento da população brasileira e mundial. Esse envelhecimento faz com que as células fiquem mais susceptíveis a mutações que levam ao aparecimento de tumores. A adoção de um estilo de vida saudável (alimentação saudável, atividades físicas regulares, não consumo de tabaco, etc) é um dos fatores preponderantes para evitar o surgimento do câncer. Entretanto, alguns tipos de câncer têm sua origem ligada a fatores exclusivamente genéticos (Síndrome de *Li-Fraumeni*, Síndrome de *Lynch*) e a adoção

[2] Estimativa 2014: Incidência de Câncer no Brasil / Instituto Nacional de Câncer José Alencar Gomes da Silva. Coordenação de Prevenção e Vigilância 2014. Fonte: http://www.inca.gov.br/estimativa/2014/estimativa-24042014.pdf

[3] Registro de Câncer de Base Populacional - Segundo o INCA, este tipo de registro de câncer coleta dados de uma população com diagnóstico de câncer em uma área geográfica delimitada. Fonte: http://www.inca.gov.br/conteudo_view.asp?id=353

[4] http://tabnet.datasus.gov.br/tabdata/LivroIDB/2edrev/d05.pdf

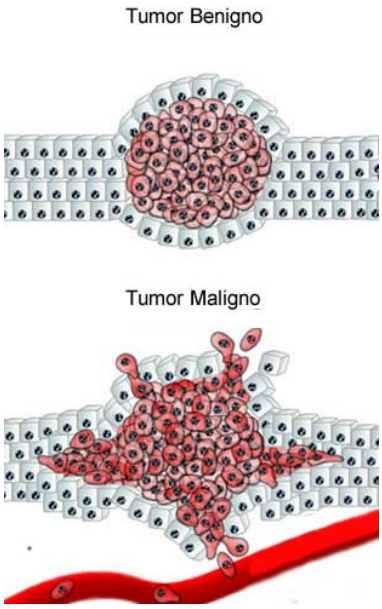
[5] http://tabnet.datasus.gov.br/tabdata/LivroIDB/2edrev/a10.pdf

[6] Estimativa 2014: Incidência de Câncer no Brasil / Instituto Nacional de Câncer José Alencar Gomes da Silva. Coordenação de Prevenção e Vigilância 2014. Fonte: http://www.inca.gov.br/estimativa/2014/estimativa-24042014.pdf

de hábitos saudáveis apenas contribuem para aumentar a qualidade de vida ou para retardar o aparecimento de tumores ligados a elas.

O Câncer é considerado como um conjunto de diversas doenças que causam o crescimento e a divisão desordenada de células de um determinado tecido do corpo humano. Segundo [Jor04], ela é uma coleção de doenças que compartilham a característica comum de crescimento celular incontrolado. Esse processo resulta no aparecimento de uma massa de células denominada neoplasia ou **tumor** e que, se não forem tratadas, podem levar o indivíduo à morte. De maneira geral e abrangente, existem dois tipos de tumores: os **malignos** e os **benignos**. Os tumores malignos têm como características o crescimento desordenado das células cancerosas e sua disseminação pelo corpo por meio do sistema linfático ou do sistema circulatório, podendo causar a morte do indivíduo acometido. Já os tumores benignos não possuem a característica de se disseminar pelo corpo, pois ficam encapsulados e restritos ao tecido de origem, apesar de poderem crescer e assumir grandes dimensões (Figura **??**). Não é comum que tumores benignos causem a morte do indivíduo, apesar de determinados tipos terem essa capacidade (tumor benigno no cérebro, por exemplo). [Jor04]

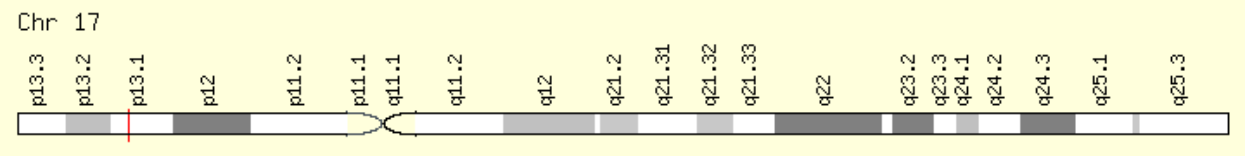
Figura 2.2: *Diferença entre tumores benignos (acima) e tumores malignos (abaixo). Fonte: Stop Cancer Portugal (http://www.stopcancerportugal.com/wp-content/uploads/2013/04/Diferenca-Tumor-Maligno-e-Benigno_Vertical.jpg)*



A causa principal para o aparecimento de tumores em um tecido é o mal-funcionamento do mecanismo de controle de crescimento e divisão celular, que se encontra no DNA da célula. Existem três tipos de genes responsáveis por realizar essas tarefas: os proto-oncogenes (*proto-oncogenes*), os genes supressores de tumor (*tumor supressor gene*) e os genes reparadores do DNA (*DNA repair genes*). Os proto-oncogenes são responsáveis pelo controle do crescimento das células e, quando apresentam mau funcionamento, permitem que as células cresçam desordenadamente, formando um tumor. Os genes supressores de tumores também são responsáveis pelo crescimento e divisão celulares. Quando alterados, estes genes permitem que as células se dividam de maneira descontrolada, formando tumores. A Síndrome de *Li-Fraumeni* é causada por uma mutação no gene $TP53$, que é do tipo supressor de tumores (Figura 2.3). Já os genes responsáveis pela reparação do DNA atuam na correção de trechos do DNA que apresentarem erros, que podem ter sido causados por uma

divisão celular mal sucedida ou por agentes externos, como radiação ou substâncias cancerígenas (tabaco e álcool, por exemplo) ([BCK03] *apud* [DSdC+09]).

Figura 2.3: *Localização do gene TP53 (linha vertical próxima da localização p13.1), no braço curto (p) do cromossomo 17, na localização 13.1. Extraído de http://www.genecards.org/cgi-bin/carddisp.pl?gene=TP53#localization*



Um tumor pode ocorrer dentro do próprio tecido em que ele foi originado ou então pode migrar para outro tecido distante no corpo humano por meio da corrente sanguínea ou então pelo sistema linfático. Ele também pode "invadir" os tecidos adjacentes ao local onde ele se originou. No primeiro caso, o tumor é chamado de **primário**. No segundo caso, o tumor é chamado de **tumor metastático** e o seu processo de espalhamento em outros tecidos é conhecido como **metástase**. Durante o processo de metástase, o tumor passa por alguns estágios até a sua instalação no tecido afetado. São elas **Metástase local**, quando o tumor invade um tecido adjacente ao seu, **Intravasamento**, quando o tumor extrapola o tecido e adentra aos vasos sanguíneos ou cai no sistema linfático, **Circulação**, quando as células tumorais circulam pelo corpo usando o sistema circulatório sanguíneo, **Extravasamento**, quando as células se projetam para os tecidos adjacentes aos vasos sanguíneos ou capilares onde ele se fixou, **Proliferação** quando as células tumorais começam a se multiplicar e a crescer no tecido "hospedeiro" e **Angiogênese**, quando começam a se formar vasos sanguíneos adjacentes ao tumor para que as células tumorais sejam alimentadas de oxigênio e tenham suas toxinas excretadas. A Síndrome de *Li-Fraumeni*, pela própria constituição da mutação presente nas células tumorais, provoca o crescimento de vários tumores primários em vários tecidos do corpo do indivíduo. [Jor04]

O site do NCI[7] (*National Cancer Institute* - principal órgão do governo norte americano responsável pelas pesquisas sobre Câncer) relata a catalogação de mais de 100 tipos diferentes de câncer, que podem ser descritos pelos tecidos que atacam ou pelo tipo de célula tumoral que formam. Alguns tipos são:

- Carcinoma: Tipo de câncer comum que se forma no tecido epitelial, tecido que reveste a parte interna e externa dos órgãos;
- Sarcoma: Tipo de câncer que se forma nos chamados tecidos moles (músculos, ligamentos e vasos linfáticos, por exemplo) e nos ossos;
- Leucemia: Tipo de câncer que se forma na medula óssea, responsável por produzir as células sanguíneas. Não resulta em aparecimento de nenhum tumor, mas modifica a estrutura dos glóbulos vermelhos[8] e dos linfócitos[9] de forma que as primeiras percam a capacidade de transportar oxigênio para as outras células e que as segundas percam a capacidade de combater infecções pelo corpo;

[7] http://www.cancer.gov/about-cancer/what-is-cancer#how-cancer-arises
[8] Células responsáveis por transportar o oxigênio dos pulmões para as células do nosso corpo.
[9] São as células responsáveis pelo nosso sistema imunológico.

- Melanoma: Tipo de câncer que ataca as células produtoras de melanina.

O diagnóstico do tipo de célula tumoral encontrada e do tipo de câncer que está acometendo um paciente é feito por meio de um exame chamado **biópsia**, em que um pedaço do tecido suspeito é retirado do indivíduo e examinado em um microscópio. O diagnóstico oriundo de uma biópsia pode revelar o tipo de câncer, qual o estágio de avanço da doença e qual a origem do tumor analisado. O uso de um sistema de classificação de tumores malignos TNM[10] auxilia na definição do **estadiamento clínico** da doença, que indica o quão avançado, invasivo e espalhado se encontra o tumor. Ainda segundo o site do NCI ([NCI]) os tumores podem ser agrupados em 5 categorias distintas baseado no resultado da biópsia:

- *In situ*: as células tumorais ocorrem, exclusivamente, na camada do tecido onde se desenvolveram;
- *Localizado*: as células tumorais se espalharam para outros tecidos, órgãos e linfonodos próximos do local de origem;
- *Distante*: as células tumorais se espalharam para locais distantes do seu tecido de origem, por meio do sistema circulatório sanguíneo e do sistema linfático;
- *Desconhecido*: não foi possível definir o estadiamento do tumor;

Sabendo que o câncer depende, exclusivamente, de alterações genéticas para sua proliferação [Jor04] é possível dizer que os fatores hereditários exercem um papel fundamental na descoberta e no controle da doença. Por essa razão, no caso da Síndrome de *Li-Fraumeni*, o correto aconselhamento genético também contribui para a melhoria na qualidade de vida do paciente.

2.2 Ontologias

O termo Ontologia está presente nas sociedades desde a Grécia antiga. Aristóteles, por meio de um de seus mais famosos tratados, *Metaphysics*, foi um dos primeiros filósofos gregos a escrever sobre a necessidade de se descrever e classificar as coisas e os seres (reais ou não) em categorias, além de descrever as relações existentes entre eles [Gui05]. Não obstante o fato de que as ontologias, em sua maioria, descrevem coisas de um domínio material/real, a não existência material de algo, *per se*, não impede que ele reúna atributos e características que convirjam para um conceito comum de si mesmo. Em suma, para que haja uma ontologia, faz-se necessário, tão somente, o conceito, subjetivo ou não, daquilo que se deseja descrever. Nicola Guarino, em [GOS09], exemplifica este fato ao citar que um unicórnio, apesar de não existir em um mundo real, reúne conceitos e características que fazem com que as pessoas formem um conceito sobre este ser. Assim, o estudo da ontologia é, em contraponto a outras ciências, descrever e classificar, de maneira genérica, as coisas do mundo. Por essa razão, ela pode estar presente nas mais diversas áreas de estudo, como ponto de apoio ou de convergência, para cientistas, teorias e definições de termos. É possível, por meio dela, encontrar respostas a questionamentos filosóficos como "quais elementos compõem determinado objeto?" ou

[10] Sistema de classificação de tumores malignos inventado na França entre os anos de 1943 e 1952 por Pierre Denoix e que leva em consideração o grau de avanço do **T**umor, o grau de invasibilidade do Linfo**Nodos** e o grau de **M**etástase em que o tumor se encontra.

"quais as aplicações de determinado procedimento?", aproximando-se, consequentemente, de outras ciências, como a Mereologia[11].

Entretanto, apenas recentemente (século XX) o termo **Ontologia** ganhou seu espaço na computação através da área da **Inteligência Artificial**. Aqui, segundo Guarino [GOS09], uma ontologia descreve conceitos (objetos) e as relações entre eles, se comportando como um artefato computacional. A origem do uso das ontologias na Inteligência Artificial reside na necessidade de representação de conhecimentos diversos, como estruturas de dados, relações lógicas, matemática e hierarquia de conceitos (taxonomia), consistindo como ferramenta importante na formação das bases de conhecimento. Em outras áreas, por exemplo, as ontologias estabelecem bases conceituais como meio formal de representação de sistemas. Um exemplo desse uso são os **Modelos de Entidade e Relacionamento (MER)**[12] que descrevem as relações entre grupos de conceitos que possuem características comuns e que representam a **conceitualização** de um problema. É, na realidade, a maneira formal de representar estruturas de dados que armazenam, agrupam e recuperam informações sobre um determinado domínio do conhecimento. De fato, essa ideia corrobora com o exposto por Guarino: "[...] cada sistema de informação possui sua própria ontologia, seja ela implícita ou explícita [...]" [Gua98].

A **Conceitualização**, por sua vez, é uma visão abstrata, uma representação de determinado objeto ou conceito, junto com seus atributos e relações, do mundo o qual se quer representar [GN87]. Segundo a definição de Gruber [Gru95], uma ontologia **é uma "especificação explícita de uma conceitualização"**. Guarino, entretanto, discordou desse conceito por avaliar que o termo conceitualização apenas define o sentido filosófico das coisas, sem levar em consideração a sua representação [Gua98]. O problema, segundo ele, estava na própria noção de "conceitualização" usada por Gruber, e descrita em [GN87]. Como resultado dessa análise, Guarino [Gua97] propôs, então, uma nova definição para o termo "ontologia", levando em consideração um acordo entre as diferentes partes envolvidas no entendimento sobre determinado conceito e descrito em uma certa **linguagem**: *"An ontology is an explicit, partial account of the intended models of a logical language"*. Esse novo entendimento a respeito do termo conceitualização veio de nova tentativa de definir o conceito de ontologia feita por Gruber e mencionada em [**?**]: "*[...] Ontologies are agreements about shared conceptualizations [...]*". Percebe-se, segundo Guarino, que a linguagem é parte essencial no processo de definição de uma ontologia, distanciando-se, assim, do campo exclusivamente filosófico e se aproximando mais do sentido computacional.

Já mais recentemente, em 2001, Noy e McGuiness [NM01] definiram ontologia como uma **descrição explicita, em um determinado domínio de conhecimento, de todos os conceitos, relações (propriedades) e restrições sobre esse modelo**. Sua definição adentra um pouco mais a área da Inteligência Artificial, descrevendo elementos que estão presentes no mundo real e estabelecendo as regras de como esses elementos interagem, entre si, e com outros elementos de outros domínios. Além desses, existem diversos outros trabalhos que discutem o termo ontologia e seu conceito sob os aspectos filosóficos e computacionais ([Gui05, GOS09, Bor97]). Para este trabalho, utilizaremos a noção de ontologia dada por Guizzardi em [Gui05], levando em consideração a linguagem como aspecto fundamental na construção de um modelo conceitual adequado à sistemas

[11] Segundo o dicionário Merriam-Webster, Mereologia é a ciência que estudo as relações do tipo parte-todo entre os indivíduos.

[12] Modelo de Entidade e Relacionamento é um modelo de representação de alto nível que representa a semântica dos dados em um domínio do conhecimento.

computacionais.

O grande volume de dados gerado hoje por sistemas, pessoas e maquinários diversos, juntamente com a necessidade crescente das organizações de estruturar informações, em busca de conhecimento que as levem a obter vantagens competitivas, têm justificado, cada vez mais, a construção e o uso de ontologias em sistemas ditos "inteligentes". Cada organização estrutura as informações e seus dados segundo seus próprios critérios e interesses, gerando, assim, diferentes pontos de vista sobre os mesmos domínios (conhecimentos diferentes do mesmo domínio). Não que esses pontos de vista sejam necessariamente antagônicos, pois podem, inclusive, ser complementares entre si, contendo diferentes **conceitualizações** do mesmo domínio e que, quando vistos sob uma ótica comum, acabam por compartilhar esse conhecimento gerado, permitindo diferentes tomadas de decisão por parte dos responsáveis pelas organizações.

Ainda nesse cenário, as ontologias podem ser usadas para integrar sistemas diferentes, dentro e fora de uma mesma organização, por meio da uma camada intermediária conceitual que mapeia/traduz conceitos diferentes sobre coisas e objetos de um mesmo domínio. Uma instituição de ensino privado, por exemplo, possui diversos sistemas de informação: controle acadêmico, controle financeiro, sistema de protocolo, sistema de acesso à biblioteca, dentre outros possíveis. Um aluno dessa instituição pode ser visto sob diversos pontos de vista diferentes, dependendo do sistema de informação usado. Para o sistema financeiro, um aluno pode ser visto como um **cliente**, que gera receita e, portanto, deve ser tratado como um **cliente**. Já para o sistema acadêmico, o aluno é visto como um **discente**, interagindo com professores, cursos e disciplinas. Suas propriedades, vistas sob cada um dos contextos descritos, são diferentes e representam características distintas de cada um. Não obstante esse fato, cliente e discente estão sob o mesmo domínio (acadêmico) e suas propriedades, quando vistas conjuntamente, só contribuem para a extração de conhecimentos pertinentes aos dois contextos e que não são claras quando vistas separadamente (por exemplo: o índice de reprovação em determinada disciplina de um determinado curso está ou não relacionado à inadimplência do aluno?).

Dentro do contexto computacional, as ontologias podem servir a diversos propósitos possíveis, mas, segundo [NM01], todas elas são formadas por elementos comuns: **classes**, **relações**, **atributos**, **axiomas** e **indivíduos**. As classes são um conjunto de características comuns reunidas, geralmente, em um termo. O termo ÓRGÃO, no domínio da anatomia humana, é usado para caracterizar uma estrutura formada por um conjunto de tecidos presente em uma pessoa e que desempenha uma função sistêmica específica. Não é preciso que haja fisicamente um órgão para que o conceito ÓRGÃO exista na natureza. Entretanto, o fígado só será considerado um ÓRGÃO se reunir as características comuns a um órgão e que formam a **conceitualização** da classe ÓRGÃO. As relações representam as restrições semânticas existentes entre duas classes. As restrições estabelecem (mas não de maneira isolada) as **bases semânticas** de uma ontologia, na medida em que apresentam **axiomas** que limitam a classificação de indivíduos dentro das classes, de acordo com o modelo conceitual. É possível dizer, por exemplo, que a classe TECIDO se relaciona com a classe ÓRGÃO por meio da relação *compoeOrgao*. Esta representa uma restrição na relação existente entre tecidos e órgãos indicando que um órgão, por exemplo, não é composto por outro órgão. Também é possível que uma classe seja subclasse de outra, como, por exemplo, TECIDO_CONJUNTIVO seja subclasse de TECIDO e que TECIDO_ADIPOSO seja subclasse de TECIDO_CONJUNTIVO. Os atributos apresentam os dados específicos sobre determinado indivíduo que pertence a determinada classe. Esses atributos

são, na realidade, as características que identificam esse indivíduo como sendo parte da classe em que ele se encontra. É possível, por exemplo, que o indivíduo `fígado` possua o atributo **`cor`**, cujo valor seja **"vermelho"**, ou **`peso`**, cujo valor seja **"1,9 kg"**. O indivíduo, por sua vez, representa uma instanciação de um representante de uma determinada classe (como o indivíduo `fígado` que pertence à classe `ORGÃO`). Essa complexa interação entre os cinco elementos que compõem uma ontologia formam uma **base de conhecimento**. A diferença entre ontologia e base de conhecimento é, entretanto, muito tênue. Uma ontologia descreve uma maneira estruturada de organizar o que se sabe sobre algum domínio do conhecimento e é utilizada na construção das bases de conhecimento [NM01].

As bases de conhecimento, por sua vez, podem ser modificadas/ampliadas/refinadas, dentre outras maneiras, por meio da:

- *Criação de novas regras por meio da adição de axiomas lógicos:* através da observação do mundo real e das relações existentes entre os objetos, é possível mudar a maneira como percebe-se a realidade, e com isso, mudar a sua descrição formal em uma ontologia por meio da modificação (adição ou exclusão) de novas regras que representem essa nova percepção.

- *Inclusão de novos indivíduos à base de instâncias:* por meio da mudança de percepção do mundo real, é possível que novos indivíduos passem a fazer parte de outras classes, ampliando o conhecimento existente sobre aqueles indivíduos. O que houve, nesse caso, foi uma mudança na forma como se percebe o domínio de conhecimento (conceitualização) e não uma mudança, de fato, no mundo que percebemos.

- *Uso de inferência sobre a já existente base de conhecimento:* a inferência é o processo pelo qual é possível deduzir novos axiomas a partir de outros que estão explicitamente declarados na base de conhecimentos. O uso de motores de inferência automatizados para deduzir novos axiomas ou regras vêm ganhando mais espaço ultimamente. O aumento no poder de processamento dos computadores é um dos fatores principais para esse crescimento. Nesse caso, o motor de inferência deduz novos axiomas que, anteriormente, não estavam explícitos no modelo. Dependendo da importância dessas novas regras, elas podem ser adicionadas à base de conhecimentos existente, ampliando, assim, o que se conhece de fato sobre essa realidade.

2.2.1 Classificação das ontologias

As ontologias, em uma abordagem mais computacional, podem desempenhar papéis distintos na modelagem do conhecimento. Podem servir, por exemplo, como referência na modelagem de novos conceitos ou, então, modelar um domínio de problema específico que reutiliza conceitos já estabelecidos por ontologias mais abrangentes. Podem apenas descrever um domínio genérico e abrangente ou então descrever processos e tarefas que manipulam instâncias já classificadas segundo alguma taxonomia. Guarino descreve essas ontologias por meio do seu uso nas aplicações de software, evidenciando a importância que elas têm para o desenvolvimento de novos sistemas de informação que delas fazem uso [Gua98]. Ele propõe que as ontologias sejam classificadas em 4 tipos, dependendo da função que elas desempenham no domínio em questão:

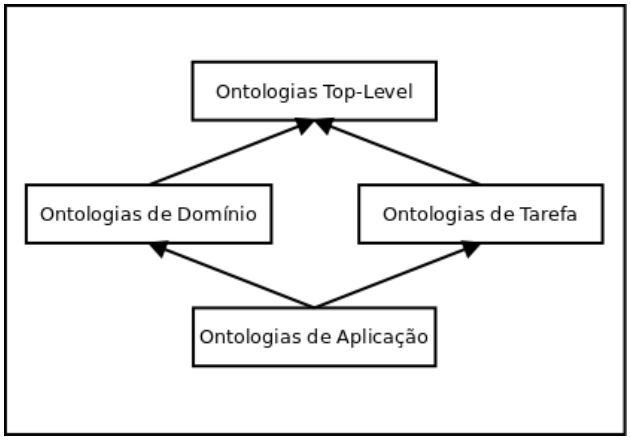
- *Ontologias Top-Level*: que descrevem, de maneira genérica, conceitos que são independentes do domínio do problema. Aconselha-se que elas sejam usadas para estabelecer uma semântica

de conceitos comum a ser utilizada por outras ontologias. Pode-se dizer que essas ontologias modelam o senso comum;

- *Ontologias de Domínio*: descrevem "mini-mundos"específicos do domínio do problema, como conceitos sobre biomedicina ou vinhos. É comum as ontologias de domínio fazerem uso de conceitos e relações definidas nas ontologias *top-level*;
- *Ontologias de Tarefa*: utilizadas para descrever tarefas, processos ou atividades independentemente da situação-problema, como, por exemplo, o diagnóstico de determinada doença, o processo de fabricação de vinhos ou etapas da construção de um veículo. As ontologias de tarefa, assim como as de domínio, também fazem uso dos conceitos e relações estabelecidos pelas ontologias *top-level*;
- *Ontologias de Aplicação*: descreve conceitos específicos de uma situação problema e que dependem das ontologias de domínio e das ontologias de tarefa.

A classificação descrita sugere uma dependência funcional entre as ontologias, formando um conjunto de "camadas" na qual aquelas, nos níveis mais inferiores, dependem dos conceitos e relações estabelecidos por aquelas de níveis superiores (Figura 2.4).

Figura 2.4: *Hierarquia dos tipos de ontologias segundo a função desempenhada por ela. Adaptado de [Gua98].*

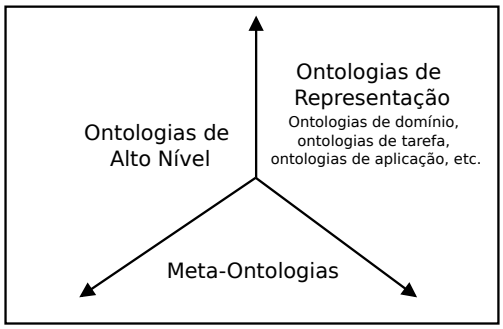


Diversos autores descrevem outras classificações de ontologias segundo critérios distintos. Guizzardi, por exemplo, diferencia entre dois tipos diferentes de ontologias, segundo o nível de abstração: ontologias de referência (*reference ontology*) e ontologias leves (*lightweight ontology*) [Gui05]. Uma **ontologia de referência** é aquela construída e utilizada para estabelecer um consenso a respeito do mundo real, unificando a conceitualização do domínio modelado. Uma **ontologia leve** é usada depois que os envolvidos entram em acordo quanto ao uso dos conceitos e decidem construir ontologias que irão ajudar os sistemas e os usuários a resolver problemas específicos de um dado domínio. Por essa razão, a preocupação maior dessas últimas ontologias é ser o mais computável possível.

Mais recentemente, Martinez-Cruz *et al.* [MCBV12] citam dois critérios básicos para a classificação das ontologias: um baseado na **linguagem e na valorização semântica dos dados** (apresentado em [UG96]) e outro critério baseado, exclusivamente, na **semântica das informações** (apresentado em [Gua98]). No primeiro critério, as ontologias são classificadas como: *proper ontologies*, ontologias que descrevem toda a semântica sobre um determinado domínio do conhecimento e; as **ontologias leves** (*lightweight ontologies*), como por exemplo as taxonomias, esquemas

conceituais de banco de dados, XML, etc. No segundo critério (Figura 2.5), as ontologias são classificadas em 3 tipos: **meta-ontologias** (*metaontologies*), que estabelecem inclusive o vocabulário conceitual usado na construção de ontologias, como classe, propriedade, axioma, etc.; as **ontologias de alto nível** (*high level ontologies*), que representam conceitos gerais, de alto nível, sobre determinado domínio do conhecimento; e **ontologias de representação**, que englobam todos os outros tipos (de domínio, de aplicação, de tarefa, etc.).

Figura 2.5: *Classificação das ontologias segundo a semântica das informações. Adaptado de [MCBV12].*



É possível observar que a classificação descrita em [Gua98] difere, pouco ou nada, daquela apresentada mais recentemente por [MCBV12], quanto à semântica das informações. As ontologias *top-level* ([Gua98]) são agora descritas como ontologias *high-level* (entretanto, mantêm o mesmo objetivo já descrito anteriormente) enquanto que as ontologias de domínio, de tarefas e de aplicação [Gua98] são colocadas em um mesmo critério de classificação, chamada de **ontologias de representação**.

2.2.2 Metodologias para engenharia de ontologias

Assim como no desenvolvimento de software, a construção de ontologias também necessita de formalismos e de etapas claras capazes de definir seu domínio do conhecimento, suas fontes de informação e sua finalidade enquanto fonte estruturada de conhecimento. Ao longo dos anos, diversas metodologias para construção de ontologias foram propostas e testadas. Entretanto, este trabalho irá concentrar-se apenas naquelas mais largamente utilizadas e discutidas, por entendermos que estas reúnem o que há de mais moderno no estado da arte.

Depois do surgimento da Web Semântica, a área da construção de ontologias ganhou cada vez mais espaço na solução de problemas computacionais que envolviam Inteligência Artificial e Processamento de Linguagem Natural, resultando em um grande crescimento no número de novas ontologias. Juntamente com esse crescimento, vieram, também, diversas propostas de metodologias para a sua construção, manutenção e testes. Como não havia um consenso (e ainda não há) sobre recomendações de metodologias para construção das ontologias, cada grupo de pesquisadores aplicava seus próprios procedimentos e métodos na resolução de seus problemas, resultando, assim, em uma nova metodologia. Em tempos atuais, mesmo com a expansão na disseminação e uso das ontologias, o que existe em termos de metodologias é um apanhado de técnicas, desenvolvidas de maneira individualizada, que apresentam técnicas e fornecem caminhos para a estruturação de conhecimentos, testadas em áreas específicas, mas que não foram colocadas à prova de maneira extensiva. Fernández-López [FL99] afirma que áreas do conhecimento que já atingiram a maturidade possuem metodologias formalizadas largamente aceitas e utilizadas, enquanto as demais ainda não.

Podemos afirmar, então, que o desenvolvimento de ontologias não pode ser considerada como uma área madura pelas razões expostas acima.

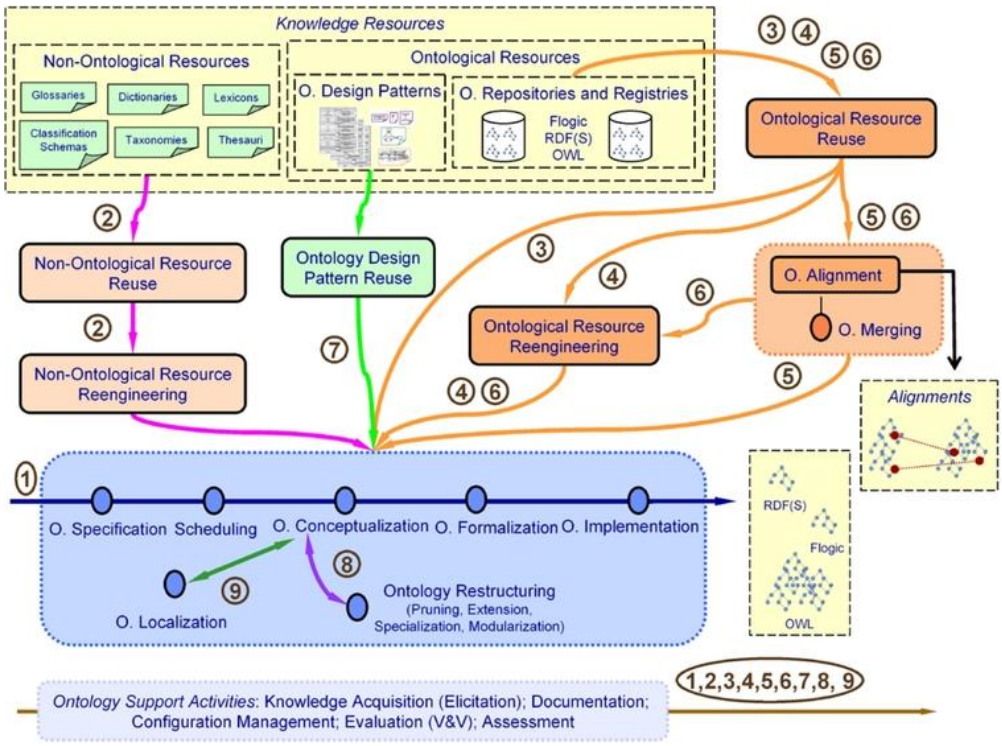
As primeiras metodologias tiveram origem na década de 90, com uma proposta de passo a passo para construção e gestão de ontologias Cyc [LG90]. Surgiram, ainda nessa década, a Enterprise Ontology [UK95], a TOVE [GF95] e a Methontology [FLGPJ97]. As duas primeiras baseadas em experiências na área empresarial e a última, baseada nas experiências dos autores no domínio da Química. Na década seguinte, surgiram as metodologias On-To-Knowledge [SSSS01], o guia de desenvolvimento de ontologias 101 [NM01], a UPON [DMN05], a RapidOWL [AH06], COLM - *Corporate Ontology Lifecycle Methodology* [LRH09] e a NeOn [CGPFL+12]. Sugerimos a divisão das metodologias em dois grupos distintos para melhor compreensão: o primeiro grupo, formado pelas metodologias TOVE, Enterprise, Methontology, On-To-Knowledge e o guia 101, e; o segundo grupo, formado pelas metodologias UPON, RapidOWL e NeOn. Essa divisão leva em consideração um marco importante para a construção de ontologias: o surgimento formal da Web Semântica, em 2001 [BLHL01]. Ainda segundo os autores, a Web Semântica veio suprir uma deficiência da linguagem HTML à época: a incapacidade de lidar com termos ambíguos em documentos diferentes. Naquele ano, ainda não havia uma recomendação oficial quanto à linguagem utilizada para escrever ontologias (atualmente, a OWL ocupa esse papel, sendo recomendada pela W3C).

Ainda hoje, não existe uma metodologia para construção de ontologias que seja formalmente recomendada pela W3C, mas sim, um conjunto de metodologias que foram construídas separadamente e que se propõem a fornecer um conjunto de passos e técnicas que foram aplicadas em domínios do conhecimento específicos. As metodologias do primeiro grupo não levavam em consideração a presença da Web Semântica, pois a mesma ainda não existia. Tampouco detalhavam os formalismos usados na linguagem de construção dessa ontologia, pois a linguagem OWL e RDF ainda não tinham sido lançadas (a linguagem XML só veio a ser formalizada em 2001). Nota-se, então, que as metodologias do primeiro grupo estabeleciam, de maneira mais genérica, os passos a serem seguidos durante a conceitualização, a aquisição dos dados e do conhecimento e da manutenção, definindo, assim, uma proposta de ciclo de vida para a ontologia sem, portanto, levar em conta aspectos da Web Semântica, como formalismos na representação dos dados, referências únicas para termos (URI - *Uniform Resource Identifier*) e criação de uma plataforma comum para disseminação e reuso dos conhecimentos. Dentre essas metodologias, Fernández-López [FL99] afirma que a mais recomendada é a Methontology, por estar mais alinhada aos padrões da Engenharia de Software e da Engenharia de Conhecimentos, apesar de reconhecer, também, que nenhuma delas está tão madura quanto as metodologias utilizadas na Engenharia de Software. Fernández-López também afirma que as propostas desse primeiro grupo não são unificadas, o que requer um grande esforço para que se chegue a um consenso quanto a um método padrão que englobe todas as características essenciais em cada um dos métodos.

Quanto às metodologias do segundo grupo, as mesmas referenciam a OWL como linguagem padrão para construção das ontologias. A NeOn, desenvolvida com o objetivo de facilitar a construção de ontologias baseada em cenários, fundamenta, explicitamente, um desses cenários (cenário 7: Reuso de Padrões de *Design* para Ontologias) na Web Semântica. Entretanto, podemos encontrar a presença dela em toda a metodologia, a partir do momento em que é definida uma camada de recursos de conhecimento (*Knowledge Resources*) e que, a mesma, é utilizada, direta ou indiretamente, por todos os demais cenários do modelo (Figura 2.6). A UPON investe no processo

unificado de modelagem (UML - *Unified Modeling Language*) para a construção de ontologias com a justificativa de que a curva de aprendizado para aqueles profissionais que já são habituados ao uso da UML será baixa. Além disso, a UPON aposta "no poder dos diagramas" [DMN05] como ferramenta de documentação e de condução do processo de construção das ontologias. Esse fato permite que a UPON se integre mais às tarefas de conceitualização e desenvolvimento, tal qual a Methontology. A RapidOWL propõe técnicas de desenvolvimento colaborativo de ontologias e de bases de conhecimento inspiradas nas metodologias ágeis de desenvolvimento de softwares. Ela é pautada no paradigma de que, toda metodologia ágil de engenharia do conhecimento deve focar nos padrões de representação de conhecimento da Web Semântica, como RDF e OWL.

Figura 2.6: *Cenários definidos na metodologia NeOn. [CGPFL+12]*



Um ponto comum entre essas metodologias é a ausência ou uma insuficiente especificação de como deve ocorrer o desenvolvimento colaborativo de ontologias. A RapidOWL, por exemplo, menciona que deve existir esse tipo de colaboração sem, entretanto, especificar como ele deve ser alcançado quando as equipes de desenvolvimento estão geograficamente separadas. Nenhuma das demais metodologias mencionadas anteriormente especificam detalhadamente como esse desenvolvimento distribuído deve ocorrer. Nesse sentido, outras metodologias foram propostas, como a CO4 [Euz96] e a OntoMaven [Pas13], com o objetivo de ocupar essa lacuna deixada pelas demais. A metodologia CO4 aborda três problemas do desenvolvimento colaborativo de ontologias para as organizações: formalização e checagem de consistência, ligações com documentação informal e o consenso sobre sua validade e aplicação. A solução encontrada pelos autores para resolver o problema do desenvolvimento distribuído foi a criação de repositórios (bases) de conhecimento, que deverão ser validados previamente. Já a metodologia OntoMaven utiliza a técnica de gestão de projetos e versionamento aliada ao software Apache Maven. Em ambas as metodologias, faz-se necessário que exista um sistema central (ou um servidor) que armazene as ontologias e que todos os colaboradores sejam capazes de se conectar a ele. Ademais, o desenvolvimento de ontologias por equipes cujos colabora-

dores encontram-se geograficamente distantes exigem, em ambos os métodos, rígidos protocolos de comunicação e políticas para consenso da base de conhecimento.

Assim, dentre as metodologias observadas, nenhuma delas apresenta uma proposta simplificada de desenvolvimento colaborativo de ontologias, requerendo, dessa forma, um grande esforço da equipe para manter um ambiente propício para a interação entre os colaboradores. Entenda-se por ambiente propício, regras rígidas e políticas de comunicação e de acordo sobre validação do conhecimento, servidores de arquivos e de ontologias e linguagens específicas para manipulação do conhecimento validado. Com isso, cria-se a exigência de um ambiente "burocratizado", com um excesso de regras e procedimentos que podem deixar rígidos os processos de modelagem do conhecimento. Um exemplo que ilustra essa rigidez poderia ser observado em uma equipe com mais de 20 colaboradores que constroem uma determinada base de conhecimentos. Para validar uma ontologia, por exemplo, além da grande quantidade de troca de mensagens e das etapas de conceitualização, faz-se necessária a aceitação unânime de todos os colaboradores, aumentando o tempo de liberação de novas versões ou até inviabilizando o processo de desenvolvimento, caso os integrantes não cheguem a um consenso (principalmente em equipes culturalmente heterogêneas e que tenham diferentes entendimentos sobre o mesmo domínio).

Assim, na Seção 4, abordaremos com mais detalhes a metodologia empregada na construção das ontologias utilizadas neste trabalho, bem como apresentaremos uma solução simplificada para o problema do desenvolvimento colaborativo de ontologias que foi desenvolvida para o cenário deste trabalho, mas que pode ser adaptado a outras realidades sem maiores esforços.

2.2.3 Linguagens para construção de ontologias

Como descrito anteriormente, as ontologias são representações **formais** do conhecimento sobre determinado domínio. Esse formalismo se traduz em representar esse conhecimento de maneira estruturada e de fácil entendimento, tanto para humanos como para computadores. Há um consenso entre alguns autores ([DCHW08] *apud* [MCBV12]) de que, se uma ontologia não puder compartilhar o conhecimento modelado e nem formar uma acepção, comum entre os especialistas da área, no tocante ao seu entendimento, então ela não pode ser considerada como tal. Por essa razão, ser compreensível aos humanos é um requisito essencial na formalização de uma ontologia. Por outro lado, cada vez mais sistemas de informação estão fazendo uso de ontologias para inferir novos conhecimentos e compartilhar novas descobertas, por meio de técnicas de dedução. O aumento no uso da Web Semântica e o surgimento, cada vez mais frequente, de projetos para gestão do conhecimento na área da biologia, da bioinformática e da medicina são exemplos da importância do papel que a ontologia exerce na descoberta de novos conhecimentos. Dessa forma, torna-se imprescindível que as ontologias façam uso de uma linguagem de fácil processamento pelos computadores

Na ciência da computação, assim como em outras áreas, o crescente emprego das ontologias levou os especialistas a desenvolver, ao longo dos tempos, diferentes formas de representar o conhecimento, que, consequentemente, se traduziu em diferentes linguagens para escrita de ontologias. Assim como sistemas de computador podem ser escritos usando diferentes linguagens de programação, as ontologias também podem ser escritas usando linguagens diferentes. O que determina, então, o uso de uma ou outra linguagem é, além de outros fatores, a aplicação que se deseja atribuir à ontologia.

As linguagens foram sendo desenvolvidas à medida em que os obstáculos e as necessidades de representar conhecimento em determinadas áreas foram surgindo. Existem diversos fatores que

diferenciam uma linguagem das outras: **expressividade** e **complexidade** são exemplos de alguns desses fatores. Na realidade, juntamente com o domínio de conhecimento da ontologia, esses são os fatores que mais influenciam na escolha de uma linguagem, no momento de escrever uma ontologia. De maneira simples, pode-se definir expressividade e complexidade da seguinte forma:

- **Expressividade**: segundo [RN09], expressividade é tudo aquilo que se deseja representar por meio de uma linguagem. Ou seja, a expressividade é um fator que indica o espectro de ideias que se pode representar, de forma estruturada, por meio de uma linguagem. Quanto maior a quantidade de ideias possíveis de serem modeladas por meio de uma linguagem específica, maior é a sua expressividade. Por exemplo, a linguagem OWL2 possui, dentre outras, duas sub-linguagens que servem a propósitos diferentes: OWL2-EL e OWL2-RL (serão explicadas com maiores detalhes mais adiante). Segundo a [W3Ca], existem ideias e informações que não podem ser expressas usando o conjunto de símbolos da OWL2-EL, mais especificamente aquelas que fazem uso da negação (complemento) e da união. Entretanto, esses operadores estão presentes na OWL2-RL, permitindo que ela possa ser usada para modelar certos tipos de conhecimento que só podem ser expressos por meios dos operadores citados, fazendo desta, uma linguagem mais expressiva que aquela.

- **Complexidade**: esta é uma área de estudos da teoria da computação. Complexidade representa o quão "difícil" é, para um algoritmo, "resolver" um determinado problema de forma eficiente. Um problema tem complexidade baixa se existe um algoritmo que o resolva usando poucos recursos computacionais, como memória, tempo de uso da CPU ou qualquer outro que se faça necessário. Quanto menos esforço computacional um algoritmo demandar, menos complexo ele será. Quando um problema computacional não possui nenhuma solução que possa ser implementável, diz-se que ele é **não computável**. Alguns problemas computáveis possuem algoritmos que demandam tanto recurso computacional que acabam por ser considerados inviáveis para uso (tempo de processamento da ordem de séculos, etc.), sendo chamados, assim, de problemas **intratáveis**.

A escolha da linguagem a ser utilizada na modelagem de uma ontologia também deve levar em conta aspectos da sua expressividade e da complexidade do problema a que ela (ontologia) se refere. Seria óbvio, portanto, que fossem sempre utilizadas linguagens muito expressivas na construção de ontologias, pois, assim, seria possível modelar todo tipo de conhecimento e realizar deduções complexas sobre ele. Entretanto, expressividade e complexidade são fatores "antagônicos" no que se refere ao processo de dedução de novos conhecimentos. Vários autores ([BCM+03, GHA07, SS09a]) descrevem o problema da **expressividade *versus* complexidade** como um grande obstáculo para a escolha das linguagens e dos mecanismos de dedução usados nas ontologias. Assim, uma linguagem só precisa conter os elementos necessários para a modelagem do domínio de conhecimento pretendido por uma ontologia para manter uma boa expressividade e garantir, talvez, a computabilidade dos algoritmos de inferência na hora de extrair novos conhecimentos. Apesar disso e da recomendação formal da W3C quanto ao uso da linguagem OWL como padrão para escrita de ontologias, não existe uma uniformização quanto ao uso dessas linguagens (OWL, OBO *file format*, KIF, etc.), muito embora haja um esforço da comunidade científica no sentido de padronizar o uso da OWL.

Historicamente, as linguagens usadas para escrever ontologias evoluíram e foram incorporando diversas características que, atualmente, as diferem umas das outras, como grau de expressividade,

sintaxe e aplicação (web semântica, aplicação específica, etc.). Durante essa evolução, algumas linguagens foram construídas fundamentando-se em outras, o que resultou em linguagens de características híbridas (como a linguagem *Ontolingua*, que foi escrita baseada na linguagem KIF e, mais recentemente, a OWL que foi escrita baseada na RDF-S). As primeiras linguagens lógicas usadas para construir ontologias foram a CLASSIC, LOOM e KIF (Knowledge Interchange Format), esta última considerada a mais expressiva das linguagens usadas para representar conhecimento. Segundo [CFLGP03], essa elevada expressividade resultou em uma linguagem de complexidade alta, fato esse que acabou impedindo que fossem desenvolvidas ferramentas dedutivas para extração de conhecimento. Até meados dos anos 90, as linguagens não ofereciam suporte à web semântica, visto que a Internet acabava de ser liberada para o uso comercial e, as linguagens existentes, ainda estavam se estabelecendo enquanto padrão. Nesse cenário, três linguagens se destacam exercendo um papel fundamental no aparecimento das atuais linguagens de ontologias: HTML, XML e a RDF-S.

A linguagem de marcação HTML, criada por Tim Berners-Lee em meados dos anos 90, permitiu que conteúdos fossem compartilhados usando a recém criada Internet [BL96]. Nessa época, também foi criada a linguagem SHOE como uma extensão da linguagem HTML e baseada em *frames* e regras, cujo objetivo era o de permitir que ontologias fossem escritas (anotadas) dentro de páginas HTML. Com isso, as taxonomias e relações modeladas no interior de documentos HTML serviam de base para que motores de inferência pudessem deduzir novos conhecimentos, conforme descrito no trabalho de Luke *et al.* 1996: *"Instead of trying to glean knowledge from existing HTML, another approach is to give HTML authors the ability to embed knowledge directly into HTML pages, making it simple for user-agents and robots to retrieve and store this knowledge."* [LSR96].

Ainda no mesmo ano, a linguagem XML foi proposta pela W3C com o objetivo de permitir a troca de informações por meio da recém-criada *World Wide Web*. Ela representa informações estruturadas que podem ser facilmente tratadas pelos computadores e, ao mesmo tempo, facilmente compreensível aos seres humanos. Além disto, possui flexibilidade para a definição de novos marcadores semânticos que podem ser usados na criação de outras linguagens. A padronização da linguagem XML pela W3C, em 2001, permitiu uma adaptação da linguagem SHOE para usar a sintaxe da linguagem XML.

Por fim, a linguagem RDF-S surgiu de uma fusão das linguagens RDF, desenvolvida inicialmente para descrever recursos utilizados na *Web*, e a RDF Schema, criada como uma extensão da RDF para definição de hierarquias de conceitos e relações. Apesar da baixa expressividade da linguagem RDF-S [CFLGP03], alguns motores de inferência foram desenvolvidos para essa linguagem, em sua maioria para checagem de consistência.

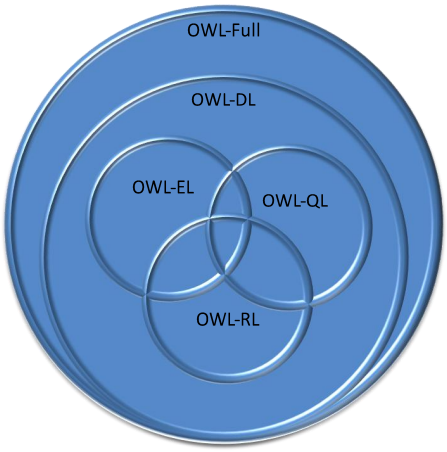
Sobre a linguagem RDF-S, foram propostas 3 linguagens específicas para escrita de ontologias: a OIL, DAML+OIL e OWL. A linguagem OIL [FVH+01] foi proposta por pesquisadores europeus, é compatível com as linguagens RDF-S e XML, possui sintaxe simples, é baseada em *frames* e possui motor de inferência para extração de conhecimento. A linguagem DAML+OIL, resultado da fusão dos esforços das comunidades europeia (OIL) e americana (DAML), implementou uma estrutura baseada em lógica descritiva (DL) e teve um motor de inferência desenvolvido para ela (BOR [SJ02]). Posteriormente, a linguagem DAML+OIL deu origem à linguagem OWL em meados de 2004 [W3Cb].

Em 2004, a W3C recomendou como padrão, a linguagem *Web Ontology Language* (OWL), baseada em lógica de descrição e *frames*, que viria a substituir a linguagem DAML+OIL. Segundo

[GHA07], um dos maiores desafios enfrentados pelo comitê que estabeleceu os padrões da OWL foi encontrar um equilíbrio para o problema da expressividade e da escalabilidade. Uma das soluções encontradas foi dividi-la, inicialmente, em três sub-linguagens: uma pouco expressiva e com baixa complexidade computacional (*OWL-Lite*), uma segunda, com expressividade intermediária (*OWL-DL*) e outra mais expressiva e, consequentemente, com maior complexidade computacional (*OWL-Full*) [GHM+08, W3Cb]. Em 2009, a W3C publicou uma nova versão da linguagem OWL, chamada de OWL 2. A principal motivação dessa mudança residia em alguns problemas estruturais da OWL 1, dentre eles problemas de expressividade, problemas de sintaxe, problemas com importações de outras ontologias e versionamento e problemas de natureza semântica. Uma descrição mais detalhada desses e de outros problemas que motivaram essa evolução pode ser encontrado em [GHM+08].

Atualmente, a versão OWL 2 conta com cinco sub-linguagens (descritas pela W3C como perfis), *OWL-EL*, *OWL-QL*, *OWL-RL*, *OWL-DL* e *OWL-Full*, cada uma possuindo um conjunto de características que permite seu emprego em diferentes cenários, em ordem crescente de complexidade e expressividade. A sub-linguagem *OWL-DL* foi dividida em três novas sub-linguagens: *OWL-EL*, *OWL-QL* e *OWL-RL* (Figura 2.7). Cada uma dessas sub-linguagens é mais restritiva que a outra e apresenta diferentes relações expressividade *versus* complexidade. A *OWL-EL* é a mais simples, menos expressiva e, consequentemente, a que apresenta melhor performance de dedução. Possui construtores simples que permitem seu uso em ontologias grandes. A *OWL-QL* é mais expressiva e também mais complexa. É recomendada para uso em integração com bancos de dados relacionais ou quando há a necessidade de organizar grande volumes de dados usando uma estrutura de ontologia simples. Por fim, a *OWL-RL*, mais expressiva dentre as três, permite o uso de axiomas baseados em regras e é recomendada para manipular dados armazenados em forma de triplas RDF em uma *triplestore* [W3Ca].

Figura 2.7: *Hierarquia dos perfis da linguagem OWL 2.*



2.2.4 *Upper-Level Ontologies* (ULO)

A pressão para integrar bancos de dados heterogêneos levou a uma tamanha expansão do uso das ontologias de domínio e de aplicação que estas acabaram por extrapolar os limites organizacionais na qual estavam inseridas e passaram a incorporar um sentido mais filosófico ao seu uso na ciência

da computação, sendo consideradas uma ponte para o alinhamento[13] de conceitos entre ontologias diferentes. Entretanto, um dos problemas que ainda persistem é a diferença na definição dos conceitos em ontologias de diferentes domínios do conhecimento, obrigando os especialistas a construir uma ”camada conceitual superior“ que unifique essas definições (ou ao menos o entendimento sobre as diversas definições).

As *Upper-Level Ontologies* (ULO), também conhecidas por *top-level ontologies* (Guarino) ou *foundational ontologies* (Guizzardi), são tipos de ontologias que formalizam, de maneira mais abstrata que as ontologias de domínio, os conceitos de um determinado domínio do conhecimento. Ou seja, as ULO são uma tentativa de estabelecer os fundamentos, um consenso entre as partes envolvidas, à respeito dos conceitos descritos por elas [KSD01]. Enquanto uma ontologia de domínio classifica conceitos e coisas de um domínio específico do conhecimento por meio de taxonomias (relações hierárquicas entre categorias), uma ULO agrupa essas categorias da forma mais genérica e abstrata possível e atribui a elas propriedades que as distinguem umas das outras. Um Gene pode ser conceitualizado como sendo parte de um DNA. Em uma ontologia de domínio, talvez essa taxonomia descrita seja suficiente para descrever o domínio do conhecimento de **expressões gênicas**[14], doenças hereditárias ou evolução das espécies. Entretanto, se for preciso descrever mais genericamente um gene, torna-se necessária a criação de um conceito mais abrangente que seja capaz de descrever os conceitos do Gene, todas as relações/propriedades que envolvem um Gene, tudo isso independente de um domínio de conhecimento específico. Pode-se, assim, classificar, em um nível de abstração maior, Gene como uma parte do DNA e sendo composto por Ácidos Nucleicos, Ácidos Nucleicos como sendo Molécula, e assim por diante (Figura 2.8).

O uso das ULO permite verificar se duas ou mais ontologias de domínio são semanticamente equivalentes no que diz respeito aos conceitos, propriedades e axiomatizações. Aliás, esse é um dos maiores benefícios do seu uso durante a construção de ontologias de domínio: prover "interoperabilidade semântica"para as ontologias que possuem domínios cruzados. Guizzardi [Gui05] corrobora com esse conceito quando afirma que as ULO (*foundational ontologies*) tentam descrever, da forma mais precisa possível, o mundo que elas representam, independentemente da linguagem utilizada e sem se preocupar com o custo computacional. Além disto, o fato dessas ontologias estarem organizadas em taxonomias permite estabelecer uma relação explicita entre as categorias de uma ULO e apenas algumas categorias (as mais superiores na hierarquia) das ontologias de domínio [Hoe10].

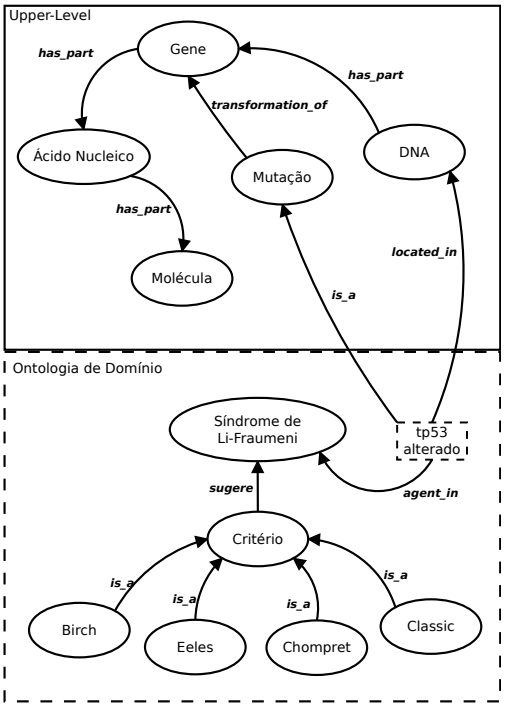
Alguns termos, mesmo que, semanticamente, representem o mesmo conceito, são utilizados de forma distinta, dependendo do domínio do conhecimento. O conceito de **Mutação**, por exemplo, pode ser representado de diferentes maneiras, dependendo do domínio do qual faz parte. Ela pode descrever, em uma ontologia de domínio sobre **Espécies**, um ser vivo de uma determinada espécie que possua características muito diferentes da espécie da qual ela faz parte (uma segunda cabeça, tronco e membros atrofiados, etc.). Ou, então, pode descrever o processo de degeneração molecular do ácido nucleico que forma um gene, no DNA de um ser vivo e que pode, ou não, ocasionar uma expressão fenotípica no ser vivo. Por conseguinte, uma ULO deve ser capaz de resolver esses conflitos de forma explícita, provendo uma base conceitual e semântica bem sólida na descrição dos seus conceitos.

Com o objetivo de criar uma representação única do mundo, os especialistas em diferentes áreas

[13] Entende-se por alinhamento de ontologias o mapeamento de conceitos pertencentes a diferentes ontologias.

[14] É o processo pelo qual uma informação codificada em um gene é usada para montar diretamente uma molécula de proteína. Fonte: http://www.genome.gov/glossary/index.cfm?id=73. Acesso: 12/10/2015

Figura 2.8: *Esboço de uso de uma Upper-Level Ontology e de uma Ontologia de Domínio representando a Síndrome de Li-Fraumeni.*



É possível notar que os conceitos de Mutação, Gene, etc. independem do domínio que está sendo modelado na ontologia de domínio. O quadrado tracejado (tp53 alterado) indica um gene específico tp53 (indivíduo) que sofreu uma mutação.

do conhecimento acabaram criando várias ULO's que descrevem, separadamente, seus universos (às vezes coincidentes), cada uma à sua maneira. A falta de processos formais de construção das ontologias ainda se constitui como um grande obstáculo na área. Alguns autores consideram o processo de construção de ontologias como uma "arte", e não uma ciência [GPFLD96]. O que existe, então, é um conjunto de princípios e critérios que são criados e seguidos por cada equipe durante seu processo de construção. Dessa forma, torna-se necessário distinguir as diferentes ontologias segundo alguns critérios [Hoe10]:

- *Categoria vs Indivíduos*: categorias podem ser instanciadas enquanto indivíduos não podem, pois estes já são instâncias de alguma(s) categoria(s). Algumas ULO fundamentam seu desenvolvimento nos Indivíduos, enquanto outras, nas Categorias. Decidir qual abordagem utilizar durante a construção de uma ULO é uma decisão que deve levar em conta a natureza do conhecimento a ser modelado.

- *Tempo vs Espaço*: nesse critério, torna-se importante decidir sobre o nível de granularidade do tempo ou do espaço usado para definir uma categoria. São exemplos de questões a serem resolvidas "O tempo será medido em intervalos (manhã, tarde, noite) ou de forma contínua (15h33m12s)?", "a localização espacial de um órgão é feita de forma discreta (acima do rim esquerdo) ou contínua (23cm à esquerda da veia cava)?" , ou "usando o plano tridimensional largura, comprimento, altura?".

- *Objetos vs Processos*: nesse nível, difere-se os conceitos entre coisas que existem e permanecem com a mesma definição, independente do tempo ou da sua existência (um fígado continua

sendo um fígado, independentemente da sua existência), ou processos que possuem começo e fim (como a cirrose, que é um processo de degeneração das células hepáticas).

Outro conceito importante para as ULO é a teoria das partes e do todo, conhecido por **Mereologia**. Ela exerce um papel fundamental na construção de ontologias *upper-level*, na medida em que permite a definição de outros termos por meio da relação de composição (um conceito pode ser formado pela composição de outros conceitos, e vários conceitos diferentes podem formar um único conceito). Apesar da importância, não existe um consenso na comunidade científica quanto à definição das relações de composição que devem fazer parte de uma ULO. Muitas das justificativas para essa não concordância reside em questões filosóficas e cognitivas da própria Mereologia. Uma discussão mais profunda e formal pode ser vista do trabalho de Guizzardi [Gui05].

2.2.5 Ontologias para relações familiares

Genealogia é uma área que estuda a origem, as linhagens e as relações entre os indivíduos de uma mesma família; segundo o dicionário Merriam-Webster[15], representa o estudo do histórico familiar. Desde o século XIX, a Genealogia desperta o interesse tanto dos profissionais quanto de curiosos e amadores. Nos estudiosos, o interesse maior está na condução de pesquisas, publicação de métodos de representação de informações genealógicas, etc. Entre os amadores, o interesse é remontar sua linhagem, conhecer sua história familiar e a de seus parentes. Em sociedades mais antigas, o estudo das origens familiares e a descoberta de linhagens tinham o objetivo de justificar heranças e títulos de nobreza recebidos. Atualmente, as pessoas pesquisam seu histórico familiar por vários motivos, como por exemplo, descobrir seus antepassados e as origens que ligam sua família a outras ou então para ligar suas origens a fatos históricos ocorridos na humanidade [Wei13].

Atualmente, observa-se um crescente interesse da ciência na Genealogia, talvez resultado do aparecimento da Web Semântica, no final do século XX, ou do crescente uso de sistemas de informações na Internet, criando uma base de dados distribuída pela web, ou simplesmente por causa da expansão, mais recentemente, no uso de redes sociais. Apesar desse crescente interesse, não existe ainda um padrão estruturado para troca de informações genealógicas, entre os sistemas de informação, que seja universalmente adotado. Como resultado, é possível encontrar esforços que levam a diferentes soluções de compartilhamento de dados referentes ao histórico familiar, baseadas em formato de arquivos para troca de dados ou baseadas em ontologias. A primeira delas apresentou um modelo de troca de dados, baseado em formato de arquivo, chamado GEDCOM[16], ainda em 1984. Diversos obstáculos (formato rígido, padrão proprietário, poucas referências, etc.) ao uso do GEDCON impulsionaram a criação de novas versões desse formato. Atualmente, ele se encontra na versão 5.5, publicada em 1996 e nenhuma das versões publicadas é capaz de armazenar dados no padrão XML, apesar de uma tentativa de construir uma com essa capacidade ter sido empreendida em 2002 (mas não foi recomendada).

O GENTECH foi uma tentativa da *National Genealogical Society*[17] (NGS) em criar um modelo padrão de troca de dados genealógicos, também baseada em arquivo. Atualmente na versão 1.1, o padrão foi criado em 1996 e, desde 2002 (versão 1.1), não sofre nenhuma evolução. Segundo

[15] "Genealogy."Merriam-Webster.com. Merriam-Webster, Acessado em 20/10/2015.
[16] https://familysearch.org/learn/wiki/en/GEDCOM
[17] http://www.ngsgenealogy.org/

[ABB+00], este não foi um modelo genealógico criado para substituir o formato GEDCOM. O GENTECH se apresenta como uma versão que possui uma visão diferente de como os dados genealógicos se relacionam e devem ser armazenados.

Mais recentemente, as iniciativas para troca de dados genealógicos foram voltadas para o uso de ontologias e da Web Semântica. [Zan05] apresenta uma primeira tentativa de construção de ontologia para informações genealógicas usando Web Semântica e armazenando os dados em arquivos de texto através da linguagem XML e OWL. Ele divide a ontologia em três módulos distintos: o primeiro (**genont**) armazena dados e informações pessoais, como relações de parentesco (primo, pai, esposa, etc.), nomes, datas de nascimento, nomes de pais, etc.; o segundo módulo (**srcont**) armazena informações sobre as fontes de dados utilizadas para buscar informações, como nome dos repositórios, localização, etc.; e o terceiro módulo, que contém os dados propriamente ditos, resultado da extração dos dados referenciados pela **srcont** e pela **genont**.

Outras implementações baseadas em ontologias foram propostas, como [Woo10], que propõe uma ferramenta de extração de dados genealógicos baseada em análise textual e regras de inferência SWRL, além de uma interface para realizar consultas no resultado das inferências. Já em [PB07], uma ontologia simples com informações familiares foi construída com o objetivo de representar informações diversas além das informações biomédicas. Nessa ontologia, não foram modeladas regras de inferência para descoberta de informações quanto a relações que não foram explicitamente afirmadas no seu ABox. Por fim, [SS09b] demonstra a construção de uma ontologia genealógica (**FHKB**) usando a linguagem OWL 2. Alguns obstáculos na modelagem de algumas relações familiares são discutidos nesse trabalho, como a construção dos axiomas para as relações familiares de irmão (`hasSibling`) e de primo (`hasCousin`), que necessitam do uso de regras SWRL e não permitem o uso do classificador textttirreflexive. A FHKB deixa de fora algumas relações familiares (como casamentos) e também não responde sobre graus de parentesco (indivíduo x é parente de 1º grau do indivíduo y).

Iniciativas baseadas em ontologias possuem uma grande vantagem em relação àquelas baseadas em formato de arquivo para troca de informações: a capacidade de descoberta de conhecimento implícito por meio de técnicas automáticas de inferência usando axiomas e regras complexos. As técnicas dedutivas não são capazes de inferir conhecimento sobre arquivos proprietários de genealogia (GENTECH ou GEDCOM, por exemplo) devido à ausência, nesses, de axiomas que definem conceitos e relações entre conceitos. Por outro lado, ontologias descrevem conhecimento através de axiomas que estabelecem um conjunto de fatos sobre uma realidade (ABox) e definem conceitos e relações entre eles (TBox). Com isso, os mecanismos automáticos de inferência são capazes, por exemplo, de descobrir novas relações familiares implícitas no conjunto de dados ou então responder a consultas complexas sobre graus de parentescos, fazendo desse tipo de implementação (ontologias) a mais adequada ao atual contexto em que vivemos (Web Semântica, grande quantidade de sistemas de informação médicos e expansão do uso da Inteligência Artificial em sistemas especialistas).

No contexto das ontologias, alguns obstáculos técnicos permeiam a construção de ontologias genealógicas, a saber:

1. Segundo [ACM12], a maioria das ontologias que descrevem genealogias assumem o modelo *CWA*[18], para que seja mantida a computabilidade e a decidibilidade. Consequentemente, o

[18] No modelo *Closed-World Assumption*, apenas as assertivas declaradas são consideradas verdadeiras e todo o mais (assertivas não declaradas ou declaradas como falsas) são consideradas falsas.

poder de dedução lógica dos motores de inferência ficam mais limitados, em razão da redução da expressividade, comparando-se a uma ontologia que assuma o modelo *OWA*[19]. O desafio, então, é, mais uma vez, encontrar um ponto de equilíbrio entre a expressividade e a computabilidade.

2. As diferenças culturais entre as sociedades são exemplos de barreiras explícitas que dificultam a construção de um modelo conceitual único de genealogia. Por exemplo, considere uma relação `a hasPartner b` que descreve o casamento de uma pessoa a com outra pessoa b (a tem cônjuge b). Essa relação poderia ser descrita como sendo uma relação **simétrica** (se a é cônjuge de b, então b também é cônjuge de a), **irreflexiva** (a não pode ser cônjuge de si mesmo) e **funcional** (se a é cônjuge de b, então a não pode ter mais nenhum outro cônjuge definido). Seja, então, uma ontologia $\mathcal{O}$ que tenha sido modelada conforme a propriedade `hasPartner` acima. Em uma sociedade poligâmica, esta propriedade causaria inconsistências a partir do momento em que existisse um indivíduo que possuísse mais de um cônjuge. A solução, então, para esse problema exigiria um "relaxamento" do axioma que descreve `hasPartner` como funcional (exclusão do axioma). Sem ele, no entanto, a ontologia $\mathcal{O}$ poderia permitir a ocorrência de inconsistências caso fosse aplicada para representar as relações familiares de uma sociedade monogâmica. Um outro problema cultural suscetível a figurar como obstáculo à modelagem de ontologias diz respeito a sociedades que criminalizam relações incestuosas (casamento entre primos de primeiro grau).

3. Outro problema reside no propósito destinado a uma ontologia genealógica. Modelar relações familiares que não sejam necessariamente consanguíneas (irmãos adotivos, por exemplo) pode resultar em uma ontologia bem diferente daquelas que tenham o propósito de modelar relações consanguíneas entre indivíduos (ontologias para representar informações sobre herança genética, por exemplo). Novamente, podemos citar a computabilidade e a decidibilidade como elementos decisivos que podem comprometer a usabilidade da ontologia. Ontologias de árvores genealógicas com muitos graus de parentesco podem demorar um tempo impraticável para que o mecanismo de inferência consiga deduzir todas as relações entre os indivíduos. É por essa razão que a maioria das ontologias que descrevem as relações familiares são do tipo *CWA*. Entretanto, mais uma vez, dependendo do propósito, uma ontologia que assuma *CWA* pode não ser adequada a uma determinada finalidade (se o propósito for descobrir relações familiares não explícitas no modelo conceitual, uma ontologia *CWA* poderá limitar o alcance do motor de inferência deixando de fora resultados que poderiam ser considerados válidos).

As ontologias citadas anteriormente possuem características muito distintas, quanto ao propósito e a granularidade das informações modeladas, daquelas almejadas para a conclusão deste projeto. Além da decidibilidade, precisamos de uma ontologia com expressividade suficiente para modelar não só relações familiares simples como também deduzir graus de parentesco entre membros de uma mesma família ou de famílias diferentes que possuem um ancestral comum. Por essa razão, o reuso das referidas ontologias não se apresentou como a melhor opção para esta solução. Não obstante esse fato, propomos um mapeamento de conceitos e relações entre a ontologia proposta neste trabalho e a ontologia de referência publicada no portal de ontologias BioPortal[20] [PB07]. A tabela 2.1 apre-

[19] No modelo *Open-World Assumption*, qualquer assertiva que não esteja declarada ou não possa ser deduzida pelos motores de inferência não pode ser concluída como verdadeira nem como falsa.

[20] http://bioportal.bioontology.org/ontologies/FHHO

senta uma comparação entre as ontologias e arquivos de troca de dados genealógicos e a GenOnto, justificando a escolha desta para mapeamento. A coluna **Inferência** informa se o arquivo/ontologia é capaz de proporcionar descoberta de conhecimento por meio de algum mecanismo de inferência existente; a coluna **Regras SWRL** informa se a ontologia possui regras SWRL implementadas em algum conceito da ontologia; a coluna **OWL 2** informa se a ontologia se encontra na versão 2, que permite escolher perfis adequados à modelagem de ontologias que usam regras SWRL (ver seção 2.2.3); a coluna **Grau Parentesco** informa se a ontologia possui algum conceito que informe as relações de grau de parentesco baseadas na consanguinidade, conforme estabelecido na Seção 5.1; e, finalmente, a coluna **Publicada** informa se a ontologia está disponível em algum portal ou repositório de ontologia para que possa ser publicamente utilizado. O capítulo 5.1 relacionará, com mais detalhes, cada um desses aspectos com a construção da GenOnto.

Ontologias	**Inferência**	**Regras SWRL**	**OWL 2**	**Grau Parentesco**	Publicada
GEDCOM					
GENTECH					
GENONT	✓				
FHHO	✓		✓		✓
FHKB	✓	✓	✓		
GENONTO	✓	✓	✓	✓	

Tabela 2.1: *Características das ontologias e arquivos de troca de informações genealógicas disponíveis.*

2.2.6 Ontologias Biomédicas

O crescente uso da tecnologia da informação na área médica, a partir do final dos anos 90, fez aumentar o volume de informações produzidas e armazenadas pelos sistemas EHR. Atualmente, são milhares de imagens geradas por aparelhos de tomografia, ressonância magnética e raio-x, informações oriundas de laudos médicos, resultados de pesquisas científicas, todos armazenados em bases de dados diversas espalhadas pelo mundo. Outro grande fator que contribuiu significativamente para o crescimento desse fluxo de dados e informações foi o mapeamento do genoma humano, também no final dos anos 90, e os consequentes desdobramentos dessa conquista (pesquisas sobre detecção de doenças tardias como Parkinson e câncer de pulmão e a transmissão de doenças hereditárias, como a Síndrome de *Li-Fraumeni*). Com tanta informação armazenada, tornou-se necessária a criação de técnicas automatizadas que minimizassem possíveis erros causados pela incorreta manipulação humana dos dados, agilizando, assim, o seu processamento. Não obstante o uso de métodos computadorizados para o processamento desse grande volume de dados, alguns problemas de natureza técnica e filosófica podem ser encontrados, como a grande quantidade de termos médicos diferentes utilizados para definir um mesmo conceito (**infarto**, **ataque do coração** e **infarto agudo do miocárdio** são exemplos de termos que definem o mesmo conceito); a redundância na definição de termos (diferentes grupos de pesquisadores em diferentes locais criam artefatos diferentes que modelam conceitos sobre o mesmo domínio do conhecimento); e diferenças semânticas causadas pelas diferentes interpretações médicas (o termo diagnóstico pode representar o estado atual em que se encontra um paciente ou então o resultado de um processo de investigação iniciado pelo médico).

Ontologias biomédicas são aquelas cujos conceitos representam entidades do domínio biomédico, como termos, processos, substâncias, elementos e partes do corpo, bem como a interação entre elas.

Segundo [HDG12], ontologias biomédicas despontam em uma área emergente na qual se aplicam teorias e métodos de diversos domínios do conhecimento, como a filosofia, a ciência cognitiva, a linguística e a lógica formal, na execução ou melhoria de aplicações biomédicas. Konopka [Kon15] afirma que o principal objetivo de uma ontologia biomédica é permitir a recuperação efetiva e o reuso de informações estruturadas legíveis por computador e armazenadas em banco de dados. Pode-se dizer que tem como objetivo não apenas a estruturação hierárquica de termos oriundos da biologia ou da biomedicina, mas fornecer um conjunto de artefatos capaz de relacionar termos de mesmo domínio, permitindo, assim, a integração de dados de diferentes bases de dados e a descoberta de conhecimento por meio do uso de linguagens de consulta especializada e de ferramentas de inferência lógica. Um hospital, por exemplo, pode ter diversos sistemas de informação médica desenvolvidos por empresas diferentes (diferentes metodologias de engenharia e modelos conceituais de dados) e em épocas diferentes (diferentes versões e terminologias para termos médicos, doenças, etc). Essa situação ilustra a necessidade de uniformizar o processamento e o armazenamento de todos esses dados para uma melhor gestão desse hospital, o que pode ser conseguido através do mapeamento entre os dados dos diferentes sistemas de informação médica e um sistema de terminologias e conceitos (uma ontologia biomédica, por exemplo) de referência. Outro cenário que acentua a importância das ontologias biomédicas é o dos desenvolvedores de sistemas de informação médica. Uma ontologia de referência nesse domínio permitiria a criação de ferramentas padronizadas para a manipulação de dados médicos mais consistentes, bem como proporcionar a integração e o cruzamento entre dados de sistemas de diferentes fabricantes. Segundo [SCC97], um sistema de terminologia de referência permite a portabilidade de regras e lembretes entre *web sites* além do refinamento dos dados sem que haja a necessidade de reescrita de código de sistema a partir do início.

GO - Gene Ontology

É possível acompanhar, atualmente, o aparecimento de diversas ontologias biomédicas que, por sua vez, são utilizadas em diversos projetos para compartilhamento de dados médicos ou descoberta do conhecimento na área biomédica. Talvez o mais utilizado de todos os projetos[21], atualmente, seja o *Gene Ontology* (GO) [Con01], desenvolvido no início dos anos 2000 e cujo objetivo principal, dentre outros, era criar um artefato para anotações biológicas para genes de seres vivos. A GO é formada por três outras ontologias que contêm um conjunto de informações básicas presentes em todos os seres vivos e representam três aspectos dos genes e das proteínas: **processos biológicos** (BP) que representam os processos de transformações que envolvem genes e proteínas; **componente celular** (CC), que descreve os elementos envolvidos para localização do gene ou da proteína no ser vivo; e **função molecular** (MF), que descreve os processos de interação e resultados da ação biológica do gene ou da proteína.

GALEN - Generalised Architecture for Languages, Encyclopaedias, and Nomenclatures

A ontologia GALEN (*Generalised Architecture for Languages, Encyclopaedias, and Nomenclatures* em medicina) [RRP96], ontologia biomédica de alto nível (*top-level*), foi criada em meados dos anos 90 com o objetivo de fornecer um vocabulário controlado que permitisse reuso de conceitos sobre processos, substâncias, estruturas e modificadores biomédicos. Alguns obstáculos no

[21] Segundo o portal de ontologias BioPortal (http://bioportal.bioontology.org/ontologies/GO), foram informados 48 projetos que fazem uso da GO. Acessado em 15-11-2015.)

uso dessa ontologia, descritos em [Yu06], são a incapacidade de expressar comportamento padrão ou de exceção e de manipulação de incertezas, o que seriam consequências da própria natureza da linguagem utilizada para sua construção (GRAIL), baseada em lógica de descrição.

UMLS - Unified Medical Language System

A UMLS (*Unified Medical Language System*), desenvolvida e mantida desde meados dos anos 80, é considerada um grande dicionário de termos e conceitos. Segundo [Yu06], ela foi desenvolvida para facilitar o desenvolvimento de sistemas de computador que simulam o conhecimento de termos médicos e de suas relações. A UMLS é formada por diversos sistemas de classificação e de vocabulários controlados, como SNOMED-CT, MESH, Gene Ontology e a ICD-10, que, juntos, formam a sua base de conhecimentos (*MetaThesauri* e uma rede semântica). O *MetaTheusauri* é uma base de terminologias construída a partir de diversos sistemas de codificação, dicionários e tesauros. A rede semântica modela relações entre os conceitos presentes no *MetaThesauri*, além de categorizar os seus termos. Apesar do grande número de pessoas envolvidas na manutenção e validação da UMLS, erros semânticos, inconsistências, redundâncias e ambiguidades na sua estrutura ainda são presentes nas versões mais recentes citeSchulze2004, Geller2009.

SNOMED-CT - Systematized Nomenclature of Medicine - Clinical Terms

A ontologia SNOMED-CT (*Systematized Nomenclature of Medicine - Clinical Terms*) é uma coleção de terminologias médicas usadas em laudos, relatórios clínicos e sistemas EHR. Ela é utilizada, principalmente, na codificação de termos médicos cujo objetivo é ajudar no armazenamento de dados e na padronização de laudos, diagnósticos e relatórios. Possui mapeamento cruzado (*cross map*) de conceitos com outras ontologias médicas, como é o caso da ICD-9-CM, ICD-10 e ICD-O-3, além de estar disponível em diversos idiomas. É descrita por diversos autores como uma ontologia de referência para termos clínicos, classificação de códigos, sistemas EHR e descrição de conceitos médicos [SCC97, HSV08, Yu06] e possui, atualmente, mais de 300.000 conceitos [Kon15].

A origem da SNOMED-CT está ligada a dois projetos anteriores à sua criação: o SNOMED-RT (*Systematized Nomenclature of Medicine - Reference Terms*) e o CTV3 (*Clinical Terms Version 3*). O primeiro teve seu desenvolvimento iniciado em 1965 (*SNOP - Systematized Nomenclature of Pathology*), e, depois de sofrer várias expansões (1975 - SNOMED, 1979 - SNOMED II, 1993 - SNOMED 3.0), recebeu a denominação de SNOMED-RT em 2000. Já o CTV3 teve suas origens no projeto *Read Codes*, criado em 1980 no Reino Unido. Em 2002, a SNOMED-RT e a CTV3 tiveram suas estruturas, hierarquias e conceitos mescladas através de um projeto iniciado em 1998 pelo *College of American Pathologists - CAP*. O resultado dessa junção foi a criação da SNOMED-CT, que se tornou o dicionário padrão para termos médicos a partir de então [IHT].

Apesar da construção e da manutenção da SNOMED-CT envolver uma grande quantidade de especialistas de domínio, médicos e organizações de saúde, assim como a UMLS, ela ainda possui erros de inconsistências e ambiguidades que podem resultar em mal funcionamento de sistemas que dela dependem, alguns oriundos da etapa de junção entre a SNOMED-RT e a CTV3, outros resultantes de erros humanos [Yu06].

International Classification of Diseases - ICD

Há alguns séculos, a humanidade empreende esforços para sistematizar a classificação de doenças no mundo. Os primeiros trabalhos com essa iniciativa datam do Século XVIII, com *François Bossier de Lacroix*, *Linnaeus* e *William Cullen* (cujos trabalhos foram *Nosologia methodica*, *Genera morborum* e *Synopsis nosologiae methodicae*, respectivamente). Entretanto, o primeiro estudo, considerado estatístico, das doenças foi realizado por *John Graunt* no *London Bills of Mortality* (no Século XVII), uma espécie de publicação com dados de mortalidade da sociedade Londrina, e tinha como objetivo estimar a quantidade de crianças nascidas vivas e que vieram a óbito antes dos 6 anos. [WHO92]. O trabalho de John Graunt inspirou diversos estudos que se seguiram ao longo dos séculos, como *Bertillon Classification of Causes of Death* (1893) e *International Classification of Causes of Death* (1900). Durante a *International Health Conference*, em 1946, foi solicitada uma revisão (sexta revisão) dos sistemas de classificação de causas de mortalidade existentes à época, que resultou na criação do *International Classification of Diseases, Injuries, and Causes of Death*, posteriormente chamado de ICD (em português, chamado de CID - Classificação Internacional de Doenças). As revisões que se seguiram trataram de corrigir erros e inconsistências (sétima e oitava e nona revisões) e mudanças estruturais com aumento de granularidade, como uma melhoria no sistema de codificação de 4 dígitos e criação de um outro com 5 dígitos (nona - CID9 e décima revisões) e com 6 e 7 dígitos (décima revisão - CID10) [WHO92]. Atualmente, a WHO tem trabalhado em uma revisão da décima edição da CID, que foi finalizada finalizada em abril de 2015 e está prevista para ser publicada em 2018 [WHO15a, WHO15b].

O surgimento da CID enquanto ontologia computacional dependeu da transcrição da sua hierarquia de códigos e capítulos para uma das linguagens utilizadas na criação de ontologias computacionais. A primeira versão da CID 10, segundo o portal BioPortal[22], foi disponibilizada para uso computacional em abril de 2013. Entretanto, Cardillo *et al.* [CEST08] e Möller&Mukherjee [MM09] apresentaram formalizações da CID 10 anos antes como parte de um projeto de mapeamento entre a CID 10 e a ICPC2. Ainda em 2013, o trabalho desenvolvido por [MSE13] descreveu as dificuldades em criar uma ontologia, usando a linguagem OWL, que representasse a CID10 e propôs uma metodologia complementar aos trabalhos mencionados anteriormente para atingir esse objetivo.

Algumas classificações paralelas surgiram de estudos mais a aprofundados sobre capítulos específicos da CID, como é o caso da CIDO, uma classificação de doenças específicas da área de Oncologia, lançada inicialmente em 1976 e que se encontra em sua terceira edição (2000). Desde o início, é considerada a classificação referência para a codificação de diversos tipos de cânceres. Seu sistema de códigos leva em consideração não apenas a topografia (local ou órgão de origem) do tumor (como é o caso da CID9 e CID10) mas também a sua morfologia (tipo de célula, grau de malignidade).

Apesar da tentativa de padronização da estrutura de códigos e da lógica utilizada para o agrupamento das doenças ou problemas de saúde semelhantes em categorias/capítulos, o mapeamento entre os códigos das diferentes classificações não é feita sempre de forma direta em razão dos diferentes graus de granularidade ou mesmo pela falta de código em uma das classificações. O CID9, por exemplo, não é capaz de identificar se uma queimadura, em um paciente, está situada no braço esquerdo ou no braço direito. Outro problema se refere à quantidade de dígitos disponíveis para codificação de doenças. Alguns capítulos possuem uma grande quantidade de doenças classificadas

[22] https://bioportal.bioontology.org/ontologies/ICD10

e, com isso, não permitem mais alocar nenhuma outra dentro dela. Como consequência, as futuras doenças acabam sendo disponibilizadas em outras categorias diferentes daquela onde ela deveria estar, dificultando seu uso pelos especialistas. Por essas e outras razões, não é possível existir um mapeamento total entre códigos CID9 e CID10 que consiga representar todas as mudanças de um sistema para o outro sem que haja perda de informações. Na Tabela 2.2, é possível observar as diferenças entre as classificações CID9 e CID10 e que justificam essa afirmação.

ICD-9	**ICD-10**
3-5 dígitos de tamanho	3-7 dígitos de tamanho
Comporta aproximadamente 13.000 códigos	Comporta aproximadamente 68.000 códigos
Primeiro dígito é caractere (E ou V) ou numérico; dígitos de 2-5 são números	Primeiro dígito é caractere; dígitos 2 e 3 são numéricos; dígitos 4-7 são alfanuméricos
Espaço limitado para adicionar novos códigos	Espaço flexível para adição de novos códigos
Baixa granularidade	Alta granularidade
Não modela "lateralidade" (por exemplo, braço esquerdo *versus* braço direito)	Modela "lateralidade"

Tabela 2.2: *Características que diferem os sistemas de codificação de doenças CID9 e CID10. Adaptado de [AMA14]*

Apesar dessas ontologias representarem apenas um sistema de classificação de doenças baseado em uma hierarquia de códigos, elas são consideradas ontologias importantes para a construção de outras mais complexas que envolvem extração de conhecimento sobre doenças, causa/consequência de sintomas médicos e para sistemas baseados em diagnósticos.

OBO - Open Biomedical Ontologies

O crescente uso de ontologias pela comunidade biomédica trouxe consigo um problema que ainda está longe de ser resolvido: o grande número de ontologias criadas por comunidades diferentes que modelam domínios semelhantes sem reuso de conceitos. Esse fato resulta em ontologias que possuem definições de termos ambíguas ou redundantes, e, até mesmo, lacunas, em que determinados termos ou conceitos são ignorados durante a modelagem, como os projetos **caBIG** [FSM+05, vEB06] e o **HL7** [HL7]. O projeto caBIG é uma iniciativa do NCI que propõe integrar dados de todas as pesquisas de câncer em uma enorme infraestrutura de armazenamento de dados. O HL7 propõe uma padronização para troca, gestão e integração de informações relevantes para a área de saúde. Entretanto, segundo [SAR+07], ambos os projetos deixam de fora informações sobre genes, seres, proteínas e doenças, considerados de extrema relevância para os pesquisadores da área biomédica.

O Projeto OBO nasceu como um esforço conjunto de pesquisadores e médicos para a criação de um vocabulário biomédico controlado que pudesse ser utilizado de maneira compartilhada pela área médica. Ainda segundo [SAR+07], ele nasceu a partir dos princípios que nortearam a criação da *Gene Ontology*:

- Ter boa padronização de sua estrutura que permitisse o processamento automático através de algoritmos e consequentes expansões das definições de novos termos;

- Ser uma ontologia aberta para uso e manutenção pela comunidade científica, permitindo sua evolução incremental, e;
- Permitir cruzamento de termos com o de outros repositórios formando, assim, uma "rede semântica" de definições que pudesse ser utilizada abertamente pela comunidade científica ao redor do mundo;

Posteriormente, baseado na crença de que *"o valor dos dados é maior quando ele está representado por meio de uma estrutura que lhe permita ser integrado com outros dados"* [SAR+07], desenvolvedores que contribuíram com parte do projeto OBO deram início à *OBO Foundry*, uma iniciativa para a criação e melhoria de um conjunto de princípios que contribuem para o alinhamento de ontologias desenvolvidas por comunidades diferentes e para a modelagem de novas ontologias. Alguns dos princípios que guiam a construção de ontologias OBO[23] são:

1. As ontologias devem ser criadas através de um esforço colaborativo;
2. As ontologias devem usar relações bem definidas e não ambíguas entre classes;
3. As ontologias devem prover suporte à identificação e uso de diferentes versões de ontologias;
4. As ontologias devem estabelecer clara diferenciação entre os subdomínios modelados.

Alguns projetos foram iniciados dentro da própria *OBO Foundry* com o objetivo de alinhar seus princípios a algumas ontologias já existentes. Um exemplo dessa realidade é o projeto CARO (*Common Anatomy Reference Ontology*) [HNOS+07], que disponibiliza novas recomendações para a modelagem de ontologias sobre seres vivos ou então para comunidades que desejam integrar modelos ontológicos legados alinhando-as aos novos princípios da *OBO Foundry*. Outro exemplo de ontologia nascida dentro do projeto *OBO Foundry* foi a OBI (*Ontology for Biomedical Investigations*), que oferece um vocabulário controlado de termos e conceitos para a integração de dados sobre experimentos biológicos. Ela nasceu por meio de uma expansão da FuGO (*Functional Genomics Investigation Ontology*), em 2006. Atualmente, a OBI inclui o domínio de estudos clínicos, imagens biomédicas, pesquisas epidemiológicas, protocolos, análise de dados, etc [SAR+07].

As ontologias curadas dentro da *OBO Foundry* são formadas, principalmente, por conceitos e termos que se relacionam através de relações do tipo *is_a* ou *part_of*. Entretanto, alguns problemas foram encontrados em ontologias (*Gene Ontology*, por exemplo) que faziam uso dessas relações, levando, consequentemente, a interpretações incorretas e ocasionando incoerências nos mecanismos de inferência. Um exemplo dessa situação é a ausência de relações que determinam a localização espacial de determinado conceito (`a is_located b`) que seja capaz de distinguir **espaço** e **região**. Esse fato leva a construções complexas (e por vezes ambíguas!) de semânticas muito semelhantes, mas usando relações *is_a* e *part_of* [SCK+05]. Além desse fato, o problema, segundo [SCK+05], no uso da OBO como ULO é que ela acaba sendo incorporada de maneira informal a outras ontologias e, com isso, os conceitos e o uso das relações não ficam claros o suficiente para prover consistência semântica a outras ontologias.

Com o objetivo de quebrar esses e outros obstáculos, a ontologia *OBO - Relation Ontology* (OBO-RO) foi concebida para dar suporte à padronização de relações (binárias) em ontologias biomédicas por meio do uso de um vocabulário controlado. O seu escopo é apenas estabelecer um

[23] Para uma lista completa dos princípios, recomenda-se [OBO].

TBox correto para as ontologias biomédicas. Portanto, asserções do tipo `individuo_a` *`has_Mass`* `333` pertencem ao *ABox* e, portanto, não fazem parte do escopo de modelagem da OBO.

Resumidamente, a OBO-RO possui três tipos de relações [SCK+05]:

1. <*classe,classe*>: são relações entre classes, do tipo `is_a`, como a presente na GO `'citoplasmic part'` **`is_a`** `'intracellular part'`;

2. <*classe,individuo*>: são relações entre indivíduos e classes, do tipo `instance_of`, como, por exemplo, um gene específico de controle da divisão celular (TP53) é instância (*`instance_of`*) da classe `'positive regulation of transcription, DNA-templated'`;

3. <*individuo,individuo*>: são relações envolvendo dois indivíduos, do tipo `part_of`, em que um indivíduo em particular compõe ou é composto por outro indivíduo em particular (a membrana celular de uma célula específica compõe (*`part_of`*) uma célula específica).

Com relação às classes, ainda segundo [SCK+05], estas são divididas em 2 categorias disjuntas: *`continuants`* e *`processes`*[24]. Ambos os termos representam generalizações de duas outras classes presentes na GO: `'cellular component'` e `'biological process'`. Assim, a classe `continuants` foi criada para representar coisas, como objetos em geral que perduram ao longo do tempo ou que não dependam dele para existir. Por exemplo, um gene continua sendo gene independente das transformações sofridas ao longo do tempo. O TP53, por exemplo, **é gene** e não **está gene**. A segunda classe, `processes`, representa mudanças sofridas pelos `continuants` ao longo do tempo e que, normalmente, possuem começo e fim. A apoptose celular[25] é um exemplo de mudança sofrida pela célula.

A classe `continuants` compreende as subclasse `material`, que representa tudo aquilo que possui matéria, como célula, osso, pessoa, etc.; e a subclasse `immaterial`, que compreende tudo aquilo que não é feito de matéria ou que não é tangível, como região, buraco, etc. Essas duas classes são claramente disjuntas, pois não seria possível a existência de algo que fosse tangível e intangível simultaneamente. Por sua vez, a classe `processes` pode ser dividida em 4 subclasses: *`Foundational Relation`*, que estabelece as relações básicas para os processos biológicos; *`Temporal Relation`*, que estabelece as relações em que a existência dos continuantes depende da passagem de tempo; *`Spatial Relation`*, que estabelece as relações de localização espacial dos continuantes; e `Participation Relation`, que estabelece as relações entre processos e continuantes. A Tabela 2.3 sumariza as relações (relações inversas foram deixadas de fora, como por exemplo `agent_in`, que é inversa à `has_agent`) e suas respectivas classificações.

Foundational relations	**Spatial relations**	**Temporal relations**	**Participation relations**
is_a (oriundo da OBO)	located_in	transformation_of	has_participant
part_of (oriundo da OBO)	contained_in	derives_from	has_agent
	adjacent_to	preceded_by	

Tabela 2.3: *Agrupamento dos tipos de relações da OBO-RO, segundo [SCK+05].*

[24] Aqui, daremos preferência pelos termos originais em inglês pelo fato dos mesmos serem largamente usados na literatura especializada.

[25] Segundo o Dicionário Priberam, apoptose é um tipo de morte celular programada.

BioTopLite

O projeto BioTop [BSSH08] teve suas bases inspiradas no projeto GENIA [Tsu03, OTK02], um modelo formal para representar conhecimento acerca do comportamento e das reações das células humanas e cujo objetivo era servir de base para aplicações de processamento de linguagem natural. Entretanto, a BioTop não se limitava apenas aos conceitos do GENIA, mas também a um espectro mais amplo de categorias que pudessem ser utilizadas em todas as áreas da biologia.

Entretanto, segundo [SB13], os experimentos realizados na BioTop foram desanimadores no que diz respeito à sua performance. O seu uso, alinhado às ULO's (UMLS por exemplo), para extração de conhecimento por meio de motores de inferência mostrou problemas de performance, que foram relacionadas à quantidade classes e regras presentes nas ULO's. Como resultado, uma nova versão da BioTop, chamada de BioTopLite, foi lançada, contendo apenas as classes e relações necessárias à formalização de conhecimentos sobre biologia e biomedicina [SB13].

A base estrutural da hierarquia de classes da BioTopLite foi herdada da BioTop após alguns ajustes para contemplar problemas de ambiguidade em alguns termos (como por exemplo, o termo **fratura**, que pode expressar uma ação ou algo material [SB13]). Ela também teve suas classes alinhadas às da BFO - *Basic Formal Ontology* [BFO] (característica herdada da BioTop), que distingue *Continuants* e *Occurents* (o mesmo que *Processes*, na OBO). Atualmente, a BioTopLite está sendo usada como ULO do projeto *SemanticHealthNet* [SHN], que visa integrar bases semânticas heterogêneas. Seu foco é voltado para a relação entre entidades clínicas e de informação (SNOMED-CT e CID). O uso da BioTopLite como ULO permitiu a descoberta de algumas situações que até o momento ainda não tinham sido notadas, como o fato de que os termos clínicos para doenças e seus conceitos estão mais ligados a uma situação momentânea da vida do paciente, que preenche alguns pré- requisitos clínicos para aquela doença, em um determinado momento da sua vida do que à própria condição clínica em si.

O processo de desenvolvimento da BioTop e da BioTopLite revelou outras dificuldades diferentes daquelas já elencadas anteriormente, principalmente com relação à avaliação de qualidade de ontologias. De fato, até o momento, não existem ainda critérios nem métodos formais de avaliação de qualidade de ontologias. O ideal é que uma ontologia seja avaliada levando em consideração seu reuso e sua interoperabilidade, além da extensibilidade. A Seção 2.4 abordará o tema apresentando o estado da arte sobre a qualidade de ontologias e quais critérios serão utilizados como balizadores da qualidade das ontologias produzidas neste trabalho de pesquisa.

2.2.7 Ontologias Modulares

O aumento da quantidade de ontologias descrevendo, de maneira redundante, domínios do conhecimento semelhantes se deve, em parte, ao crescimento, nas últimas décadas, no número de profissionais dedicados à modelagem destas. Já a expansão da Web Semântica permitiu que um volume cada vez crescente de dados fosse colocado à disposição dos usuários na Internet, permitindo a criação de ontologias de grande porte que tivessem a capacidade de expressar cada vez mais conceitos. Assim, o reuso de ontologias apresenta-se como uma importante técnica na tentativa de reduzir o grande número de ontologias repetidas e uniformizar a descrição de conceitos. Dividi-las em ontologias menores (modularização) facilita o reuso dos seus conceitos, permite a manutenção mais adequada das ontologias, além de minimizar o custo computacional no caso do uso de motores

de inferência, pois apenas as partes (conceitos, propriedades, indivíduos) envolvidas no processo de descoberta de conhecimento são carregadas para a memória e processadas [LZTJ10]. Para tanto, Alan Rector [Rec03] estabelece algumas condições essenciais para que a modularização de ontologias traga os efeitos esperados: (i) deve ser possível identificar e separar do todo, os módulos a serem reutilizados; (ii) os autores devem ser capazes de prover manutenção a cada um dos módulos, independentemente; (iii) os módulos devem ser capazes de evoluir, independentemente, uns dos outros e os efeitos da adição de novos módulos devem ser mínimos e; (iv) as diferenças entre categorias distintas de informações devem estar explícitas, tanto em linguagem humana como de forma estruturada para os computadores.

Um módulo é uma ontologia que representa um subdomínio do conhecimento modelado pela ontologia modular [WBHQ07, LZTJ10]. Formalmente, [WBHQ07] define ontologias modulares sob dois aspectos: **sintático** e **semântico**. Sob o aspecto **sintático**, ontologias modulares são uma coleção de módulos independentes que podem utilizar diferentes formalismos (linguagens distintas) e que estão relacionadas através de um conjunto de regras, ou seja, $\Sigma = \langle \{L_i\}, \{M_{i,j}\}_{i \neq j} \rangle$, em que Σ é uma ontologia modular, L_i é o i-ésimo módulo que compõe a ontologia modular e $M_{i,j}$ é o conjunto de regras relacionam o módulo i com o módulo j. Observa-se nesta definição dois pontos importantes: o primeiro é a independência dos módulos quanto à linguagem na qual cada um foi escrito. Esse fato permite que uma ontologia possa ser posteriormente estendida por meio de módulos que venham a fazer uso de linguagens mais poderosas ou mais adequadas ao propósito em questão. Uma situação que ilustra esse fato, por exemplo, são ontologias modulares modeladas usando a linguagem OWL1.1 e que, atualmente, podem ser estendidas por meio de novas ontologias escritas usando a linguagem OWL2. O segundo ponto é que um módulo nunca pode ser relacionado consigo mesmo $\{M_{i,j}\}_{i \neq j}$. Se considerarmos, sem perda de generalidade, que o conjunto de relações $\{M_{i,j}\}$ entre os módulos i e j tem a função de ligar conceitos locais (na ontologia local i) a conceitos externos (oriundos de outro módulo j), então $\{M_{i,i}\}$ expressaria um conjunto de relações entre um conceito local e outro conceito local, devendo ser interpretado como uma propriedade, apenas, dentro do próprio módulo i, e não uma ligação entre diferentes módulos. Sob o aspecto **semântico**, a interpretação de uma ontologia modular Σ pode ser descrita como $\mathcal{I} = \langle \{\mathcal{I}_i\}, \{r_{i,j}\}_{i \neq j}\ rangle$, em que $\mathcal{I}_i$ é a interpretação do módulo L_i e $r_{i,j}$ é a interpretação de $M_{i,j}$, que relaciona os módulos L_i e L_j.

Alguns formalismos foram desenvolvidos com o intuito de estabelecer um padrão para a partição de ontologias em módulos menores. O formalismo mais utilizado para esse fim é a linguagem OWL [BCH06a] . Ela utiliza a diretiva `owl:imports` para reutilizar conceitos do módulo externo. Apesar de simples, algumas desvantagens emergem no uso dessa abordagem, como a importação cíclica (A importa B e B importa A), a ausência de semântica localizada (a semântica de uma ontologia sobre Genealogia é importada para dentro da ontologia de dados clínicos, gerando a interpretação de que uma `Pessoa` é um dado clínico!) e a ausência de reuso de conceito parcial. Nesse último caso, considerando que uma ontologia Li-Fraumeni que importe a ontologia de dados clínicos, ou toda a ontologia de dados clínicos é importada para dentro da ontologia Li-Fraumeni ou nada é importado. Outros trabalhos descrevem outros formalismos para construção de ontologias modulares, como DDL (*Distributed Description Logics*) [BS02], P-DL (*Package-Based Description Logics*) [BCH06b] e ε-*connection* [KLWZ03]. No contexto deste trabalho, optamos por utilizar o formalismo implementado pela linguagem OWL (`owl:imports`) para representar os diferentes domínios de

conhecimento da Síndrome de *Li-Fraumeni*.

2.3 Integração de Bases de Dados usando Ontologias

Atualmente, a grande quantidade e diversidade de informações disponíveis vem exigindo cada vez mais que as soluções na área de banco de dados sejam capazes de tratar a natureza heterogênea dessas estruturas de armazenamento ([Len02] e [CG05]). Localizar e acessar dados e informações que estão espalhados entre diferentes bases de dados requer do usuário técnicas para desenvolver consultas complexas que levem em consideração esses aspectos heterogêneos mencionados acima. Dentro desse contexto, a área de integração de dados aparece como uma proposta interessante para o acesso unificado dos dados de forma transparente ao usuário final.

De maneira geral, a ideia subjacente às soluções de integração de dados existentes é permitir que o usuário submeta consultas a uma "entidade mediadora", que poderá reescrevê-las em outra linguagem, a fim de buscar as informações separadamente nas bases de dados e devolvê-las ao usuário de uma única vez. Destacam-se, dentre outras, duas abordagens da integração de bases de dados heterogêneas, conforme descrito em [Len02]: a *Global-as-view (GAV)* e a *Local-as-view (LAV)*, e que serão tratadas oportunamente na Seção 2.3.3.

Não obstante os métodos existentes conseguirem contornar alguns dos obstáculos técnicos de uma integração de dados heterogêneos, as barreiras ligadas aos aspectos semânticos dessa integração ainda se apresentam como desafios a serem vencidos, conforme veremos a seguir.

2.3.1 Desafios da integração de bases de dados heterogêneas

A crescente demanda por informações mais complexas e mais bem estruturadas tem transformado, em um grande desafio, o esforço da comunidade científica para padronizar o armazenamento e a estruturação dos dados. Nas áreas afins da biomedicina, [VSD+04] justifica que essa dificuldade em integrar dados heterogêneos é oriunda não somente da grande diversidade de bancos de dados especializados, mas também da quantidade de sistemas especialistas desenvolvidos com o objetivo de facilitar o acesso a informações que foram previamente processadas por outros sistemas. Em um banco de tumores, por exemplo, é possível encontrar uma infinidade de dados com estruturas distintas e que podem ter sido obtidos de diversas fontes de dados. Os dados oriundos de um processo de sequenciamento genético, na qual o pesquisador deseja encontrar determinada mutação em um certo tipo de gene, não necessariamente estão estruturados da mesma forma em que serão armazenados no sistema de controle do banco de dados de tumores, sendo necessário manipular sua estrutura previamente.

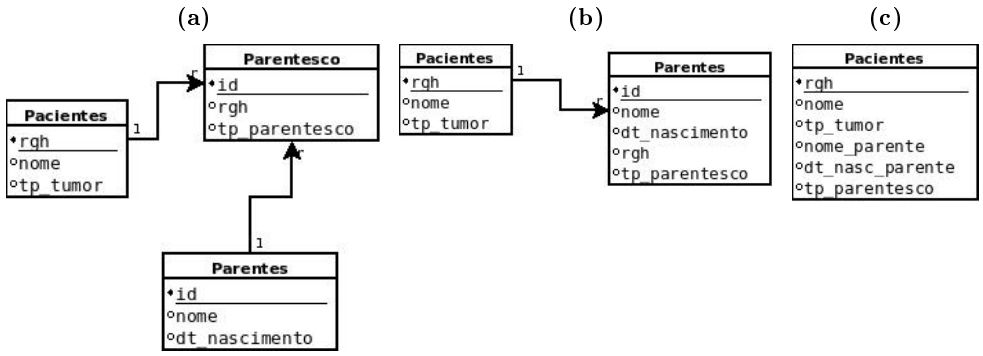
A heterogeneidade dos dados pode ser vista sob dois aspectos: **sintática**, no qual a diferença reside no modelo usado para representar o dado e na sua linguagem de definição (como o exemplo acima) e; **semântica**, no qual a diferença se apresenta nos diversos significados que um mesmo termo pode apresentar. Em [Suj01], discute-se os pontos relevantes em um processo de integração de dados heterogêneos para casos biomédicos.

- *Diferenças estruturais*

  Essas diferenças estão intimamente relacionadas em como os dados são tabulados ou armazenados nas tabelas. Os esquemas relacionais de dados, fundamentalmente, são incapazes de

representar determinados tipos de elementos estruturados, como hierarquias de tipos. A representação de uma informação pode ser feita usando vários desenhos de tabelas diferentes (Figura 2.9), dependendo do tipo de normalização[26] utilizada no esquema do banco de dados (2.9a ou 2.9b ou 2.9c) ou da granularização da informação. Por essa razão, integrar dados modelados usando níveis de granularidades diferentes ou representação/codificação de dados diferentes pode, ou não, resultar em perda de detalhes na informação.

Figura 2.9: *Diagrama Entidade-Relacionamento mostrando 3 formas distintas de representar a mesma situação.*



- *Diferenças semânticas*

  As diferenças semânticas são resultantes dos diferentes esquemas de dados cujos elementos não possuem correspondência unívoca (*um-para-um*). Nesse caso, não existe uma maneira única de relacionar dois conceitos, que representam a mesma informação, mas sim, duas ou mais formas. A título de exemplo, seja a Tabela 2.4a, que armazena os valores de exposição de pacientes a raios ultravioletas (UV) em uma data específica. Os valores medidos seguem proposta de escala da Agência Ambiental Canadense[27], em que o valor 0 representa "sem luz do sol" e o valor 11, "exposição extrema". É possível que haja medição de raios UV acima do valor 11. A Tabela 2.4b mostra uma segunda representação da mesma informação presente na Tabela 2.4a, em que os valores de exposição de cada paciente naquela tabela (*uv_index*) foram discretizados (*uv_categories*) segundo a Tabela 2.5, o que resultou em duas categorias de exposição, *low*, para indicar baixa exposição aos raios UV, e *extreme*, para indicar exposição máxima do paciente aos raios UV. A discretização (categorização) da Tabela 2.4b segue indicação dos órgãos governamentais canadenses[28], conforme Tabela 2.5.

  Não é difícil de notar que, ao representarmos os índices de raios UV por meio da Tabela 2.4b, existe uma perda significativa de detalhes quanto ao índice de exposição de cada paciente. Não é possível saber, por exemplo, qual o índice de exposição sofrido pelo paciente cujo RGH = 430. Sabe-se, apenas, que ele sofreu exposição "extrema". Se, em algum momento futuro, a categorização dos índices sofrer alguma alteração (criação de uma nova categoria *"highly extreme"* para índices acima de 13, deixando *extreme* para valores entre 11 e 13), não será mais possível saber, pela Tabela 2.4b, se esse paciente ainda deve ser classificado como *"extreme"* ou mudar para a categoria *"highly extreme"*.

[26] Segundo [EN10], a normalização é o processo de refinamento do modelo físico de dados (DER) a fim de encontrar, sucessivamente, formas melhores de agrupar os atributos das tabelas.

[27] *UV Index and Ozone, Environment Canada.* Fonte: http://www.ec.gc.ca/uv/

[28] *UV Index and Ozone, Environment Canada.* Fonte: http://www.ec.gc.ca/uv/

RGH	uv_index
334	+2
338	+1
430	+12

(a) *Valores contínuos de exposição aos raios UV para 3 pacientes.*

RGH	uv_categories
334	*low*
338	*low*
430	*extreme*

(b) *Valores discretos de exposição aos raios UV para 3 pacientes.*

Tabela 2.4: *Nível de exposição de pacientes à raios UV.*

UV Index	Description
0 - 2	Low
3 - 5	Moderate
6 - 7	High
8 - 10	Very High
11+	Extreme

Tabela 2.5: *Categorização dos índices de exposição a raios UV. Adaptado de UV Index and Ozone, Environment Canada. Fonte: http://www.ec.gc.ca/uv/*

- *Diferenças de nomes*

  Em situações relacionadas à biomedicina e à bioinformática, diferenças entre nomes de termos que representam semanticamente a mesma coisa são problemas frequentes que podem subtrair um substancial tempo de modelagem da solução. Existem para isso dicionários que trazem equivalências de nomes para alguns termos e conceitos médicos que podem auxiliar nesse processo de integração das bases de dados, como o *ICD*, *NCI Thesaurus* e o *SNOMED-CT*. Entretanto, algumas diferenças de nomes não são mapeadas nesses dicionários ou pertencem especificamente ao problema a ser resolvido. Uma situação que ilustra esse fato ocorre em um hospital que já teve diversos sistemas de informação de prontuário eletrônico (EHR - *Electronic Health Record*). É possível que o identificador de cada paciente (Patient_id) tenha assumido nomes diferentes ao longo das diversas migrações de sistemas (RGH, Patient_id, etc.) e até regras de validação diferentes, apesar de possuírem a mesma semântica. Com isso, é preciso mapear todos os nomes distintos a fim de que a integração das diferentes bases de dados consiga refletir a sua heterogeneidade.

Além dos aspectos mostrados acima, é importante citar a dificuldade em se realizar consultas em bases de dados heterogêneas usando as linguagens convencionais, como a SQL[29]. Apesar de ser poderosa na recuperação de dados e possuir grande expressividade ([Lib03]), essas linguagens levam em conta apenas as representações estruturais dos dados e carecem de elementos adicionais, ainda não implementados, que manipulem certos conjuntos de dados, como o problema da recursividade, só presente a partir da SQL3[30] ([Lib03]).

Em seguida, será mostrada uma proposta para contornar, dentre outros, os problemas acima citados. Essa proposta representa um campo de estudos recente, sinalizada pelas atividades da

[29] *Structured Query Language*: Linguagem estruturada de consulta comumente usada para manipular dados e elementos estruturais de um sistema de banco de dados relacional.

[30] A linguagem SQL passou por algumas revisões, ao longo dos anos. A versão de 1992 passou por uma revisão em 1999, ficando conhecida como SQL3, e em 2003, conhecida por SQL2003.

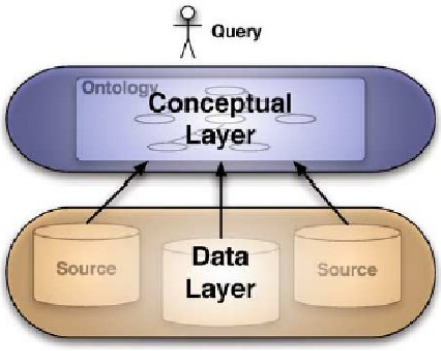
W3C, na última década, na área da Web Semântica[31] e no desenvolvimento de linguagens para ontologias.

2.3.2 Uso de ontologias no acesso a dados (ODBA)

Um dos desafios a serem vencidos no tocante à representação de conhecimento por meio da Web é o grande volume de dados armazenados nos bancos de dados e a estrutura na qual eles são concebidos, que não favorece a expressividade necessária à representação de conhecimento [LeP06]. Atualmente, um conjunto de conceitos e ferramentas vem sendo estudado e desenvolvido com o objetivo de permitir que consultas a fontes de dados relacionais passem a considerar os aspectos semânticos dos dados, por meio do desenvolvimento de ontologias que descrevem os conhecimentos existentes por detrás das estruturas relacionais. A OBDA (*ontology-based data access*) é uma área de estudo recente que tem como objetivo fornecer o acesso a uma ou mais bases de dados, desde que seja mediado por meio de uma ontologia (camada conceitual) [WVV+01]. Uma ontologia usada para mediar o acesso a dados é considerada uma camada de visão de alto nível ([CGL11, PL08]) escrita em linguagem OWL e representa conceitos, regras, objetos e propriedades (Figura 2.10). A utilidade dessa abordagem fica mais aparente quando existe a necessidade de acesso a bancos de dados que foram construídos usando metodologias diferentes e/ou que foram evoluindo ao longo dos anos de tal maneira que não existe uma forma única para recuperar essas informações sem que haja perda de granularidade.

Como exemplo, suponha um (ou um conjunto de) sistema(s), cada um com um conjunto de bases de dados, distribuídos ou não, modelados segundo critérios e especificidades distintas. Juntas, essas bases de dados formam uma camada que será acessada por outra de serviço abstrata, conhecida como uma camada de visão conceitual do domínio de interesse. Ela será considerada a mediadora do processo de integração das bases de dados e será representada por meio de uma ontologia ([PL08]). Ela deverá armazenar, de maneira incremental, todo o conhecimento adquirido ao longo do tempo sobre um determinado domínio de interesse.

Figura 2.10: *Extraído de [CGL11].*



Uma base de conhecimentos pode ser descrita em função de um conjunto de terminologias $\mathcal{T}$, conhecido por TBox, e de um conjunto de fatos sobre um determinado domínio $\mathcal{A}$, conhecido por ABox. No TBox estão todas as regras, conceitos e propriedades desses conceitos. No ABox estão

[31] Segundo Berners-Lee *et al.* ("*The Semantic Web*". Scientific American. Retrieved March 13, 2008.), a Web Semântica é uma extensão da web atual na qual é dada à informação um significado bem definido, permitindo melhor cooperação no trabalho entre computadores e pessoas.

descritos todos os fatos conhecidos sobre determinado domínio, geralmente relacionados às regras, conceitos e propriedades descritos no TBox.

Assim, considerando uma ontologia $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ a ser usada na camada mediadora em um sistema de acesso OBDA, o TBox $\mathcal{T}$ é escrito em linguagem OWL e representa os conceitos de domínio e as relações entre eles. A família da linguagem OWL utilizada no TBox influencia diretamente na complexidade computacional do sistema de inferência [CGL11].

Já o *ABox* $\mathcal{A}$ representa o conhecimento específico sobre o domínio de interesse. Ele contém os indivíduos e suas relações, atributos e valores e é representada pelas bases de dados a serem integradas, podendo assumir um volume excessivamente grande. Ou seja, o *ABox* se refere aos indivíduos e a tudo o que se sabe sobre eles.

Apesar de poder ser utilizada para acessar dados de bancos de dados relacionais, a grande vantagem dos OBDA é poder integrar bases de dados heterogêneas, não somente na estrutura dos dados como também na estrutura da própria fonte de dados (arquivos tabulados, bancos relacionais, textos, etc) . Na próxima seção, apresentaremos como ocorre o processo de integração dos dados usando ontologias.

2.3.3 Integração usando ontologias

Nas últimas décadas, várias propostas de representação do conhecimento surgiram, como CyC[32], KL-ONE [BS85], KIF[GFB+92], OBO[33], RDF, e OWL [W3Ca]. Entretanto, esta última tem tido um papel mais relevante nas áreas de inteligência artificial e de representação do conhecimento e, mais recentemente, como linguagem para viabilizar a integração de bases de dados heterogêneas.

Apesar dos expressivos progressos nessa área, alguns problemas relacionados à integração de dados ainda persistem como desafios a serem vencidos. O cenário atual inclui como desafios: (*i*) a relação entre a expressividade e a complexidade computacional; (*ii*) o grande volume de dados armazenados nas bases de dados (que forma a camada de dados do modelo de integração) e que representa a quantidade de indivíduos em uma ontologia, aumentando consideravelmente o tamanho do ABox e, consequentemente, a complexidade do motor de inferência sobre essa ontologia; (*iii*) a distância existente entre o modelo conceitual (ontologia), que representa a visão de alto nível do mundo real e o modelo relacional de dados (bases de dados), que representa a visão granulada do mundo real, voltada apenas aos dados, e; (*iv*) diferença semântica (*semantic mismatch*), relacionada à diferença entre a forma como um dado é representado em uma ontologia e em uma base de dados.

Basicamente, a integração entre bases de dados é feita usando duas metodologias de mapeamento, conforme mencionado no início desta seção: GAV e LAV. Na primeira abordagem, as consultas submetidas a uma entidade mediadora (*global schema*) são reescritas em termos de novas consultas por meio da técnica de *unfolding* e submetidas às bases de dados locais (*local schema*), ou seja, o esquema global de dados é definido por meio de visões sobre as fontes de dados locais. Já na segunda abordagem, as fontes de dados locais são definidas como visões sobre o modelo de dados global. O objetivo desta última abordagem é permitir que fontes de dados sejam adicionadas ou retiradas sem que haja prejuízo na definição do conhecimento global, apesar do processamento das consultas ser mais complexo e fazer uso de inferência [Len01]. De maneira inversa, na abordagem GAV, sempre que uma nova fonte de dados for adicionada ou retirada, o esquema global terá que

[32] http://www.cyc.com/
[33] https://oboformat.googlecode.com/svn/trunk/doc/GO.format.obo-1_4.html

ser reescrito, sob pena da consulta não retornar corretamente o conjunto de dados pretendidos. Seja a definição de um sistema de integração de dados $\mathcal{I}$ como sendo uma tripla $\langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$, onde $\mathcal{G}$ representa o esquema de dados global, $\mathcal{S}$ representa o conjunto das fontes locais de dados e $\mathcal{M}$, um mapeamento de consultas responsável por traduzir consultas entre os dois esquemas $\mathcal{G}$ e $\mathcal{S}$.

A abordagem GAV (*global-as-view*) define um modelo global mediador $\mathcal{G}$ construído sobre visões das diversas bases de dados locais $\mathcal{S}$. Nesta abordagem, o mapeamento $\mathcal{M}$ relaciona cada elemento do modelo global $\mathcal{G}$ como uma visão individual sobre as bases de dados locais $\mathcal{S}$ através da reescrita de consultas. Já na abordagem LAV (*local-as-view*), as bases de dados locais $\mathcal{S}$ são escritas baseadas em visões sobre o esquema global $\mathcal{G}$. Nesta abordagem, o mapeamento $\mathcal{M}$ relaciona cada elemento do esquema local $\mathcal{S}$ como uma consulta sobre o esquema global $\mathcal{G}$, que representa o conhecimento específico sobre um domínio geral.

Alguns critérios são relevantes para a escolha dos dois modelos (apesar de ser possível uma implementação híbrida usando os dois métodos), conforme abaixo:

1. *Global-as-view*

   - Vantagens
     - Todo o esforço de processamento está direcionado para o módulo de reescrita da consulta. Se este módulo estiver bem codificado, então as consultas resultantes serão simples o suficientes para o sistema de inferência;
     - Preferível quando o modelo de integração é mais **estático**, ou seja, existe pouca ou nenhuma mudança estrutural nos modelos de mapeamento $\mathcal{M}$ ou nas fontes locais de dados $\mathcal{S}$. Isso é decorrente do fato de que, ao adicionar uma nova fonte de dados $\mathcal{S}$ ao sistema de integração $\mathcal{I}$, então todo o mapeamento $\mathcal{M}$ tem que ser reescrito para poder recuperar os dados da nova base $\mathcal{S}$.

   - Desvantagens
     - Adicionar uma nova base de dados local $\mathcal{S}$ ao modelo significa ter que reescrever o mapeamento $\mathcal{M}$ para que os dados da nova fonte de dados sejam recuperáveis. Como essa nova base de dados não necessariamente terá a mesma estrutura das bases já existentes, então torna-se necessário ajustar a ontologia que define o mapeamento $\mathcal{M}$.

2. *Local-as-view*

   - Vantagens
     - O mapeamento não precisa ser reescrito sempre que uma nova base de dados $\mathcal{S}$ for adicionada ao sistema de integração $\mathcal{I}$. Como as bases de dados locais $\mathcal{S}$ são mapeadas como visões individuais sobre o esquema $\mathcal{G}$, então essas fontes já devem ser adicionadas considerando a estrutura do esquema global, necessitando pouco ou nenhum ajuste para isso. Além disso, um mapeamento de uma base local sobre o esquema global não interfere no mapeamento de outra base de dados local.

   - Desvantagens

- O motor de inferência é o responsável por executar as consultas submetidas e, dessa forma, mapeamentos muito complexos não conseguem ser escritos usando linguagens mais simples, como SQL. Como consequência, a complexidade do motor de inferência em resolver as consultas pode aumentar muito, encarecendo o custo de consulta ou até mesmo inviabilizando o mapeamento[34].

Além dos pontos acima citados, um fator importante que irá determinar o modelo de mapeamento utilizado na solução a ser apresentada será a complexidade de resolução de consultas. A figura 2.11 ilustra a complexidade de resolução de consultas em sistemas de integração com mapeamento LAV quanto à linguagem utilizada na base local e no esquema global (linguagens *conjunctive query* - CQ, *conjunctive query with inequalities* - CQ$^{\neq}$, *positive conjunctive query* - PQ, Datalog e Lógica de Primeira Ordem - FOL)[35]. Visões do tipo *sound* são aquelas que retornam um subconjunto das possíveis respostas sobre o modelo global $\mathcal{G}$. Em outras palavras,

$$s \subseteq q_{\mathcal{G}}$$

onde $q_{\mathcal{G}}$ representa o conjunto de tuplas resultantes de uma consulta q submetida ao modelo global $\mathcal{G}$ e s os elementos do banco de dados local $\mathcal{S}$. Visões do tipo *exact* são aquelas que retornam exatamente o mesmo conjunto das possíveis respostas sobre o modelo global $\mathcal{G}$. Em outras palavras,

$$s = q_{\mathcal{G}}$$

Nos sistemas de integração com mapeamento GAV não há a necessidade de realizar inferência nem reescrita de consultas, pois o mapeamento entre as bases de dados origem $\mathcal{S}$ e o esquema global $\mathcal{G}$ ocorre de maneira direta.

Figura 2.11: *Complexidade do processo de resolução de consultas. Extraído de [Len02].*

Sound	CQ	CQ$^{\neq}$	PQ	Datalog	FOL
CQ	*PTIME*	*coNP*	*PTIME*	*PTIME*	*undec.*
CQ$^{\neq}$	*PTIME*	*coNP*	*PTIME*	*PTIME*	*undec.*
PQ	*coNP*	*coNP*	*coNP*	*coNP*	*undec.*
Datalog	*coNP*	*undec.*	*coNP*	*undec.*	*undec.*
FOL	*undec.*	*undec.*	*undec.*	*undec.*	*undec.*
Exact	**CQ**	**CQ$^{\neq}$**	**PQ**	**Datalog**	**FOL**
CQ	*coNP*	*coNP*	*coNP*	*coNP*	*undec.*
CQ$^{\neq}$	*coNP*	*coNP*	*coNP*	*coNP*	*undec.*
PQ	*coNP*	*coNP*	*coNP*	*coNP*	*undec.*
Datalog	*undec.*	*undec.*	*undec.*	*undec.*	*undec.*
FOL	*undec.*	*undec.*	*undec.*	*undec.*	*undec.*

2.4 Métricas de avaliação de qualidade em ontologias

Qualquer que seja o artefato computacional produzido (software, ontologia ou modelo computacional), é importante que haja indicadores e metodologias adequadas para que seja possível aferir a qualidade do produto. Segundo o dicionário Merrian-Webster, **qualidade** representa *"o quão bom ou ruim uma determinada coisa é"*. Esta definição, apesar de não estar distante daquela utilizada pelos profissionais da tecnologia da informação, se apresenta muito abstrata e, consequentemente,

[34] Segundo [GHM+08] e [Kaz08], a complexidade de inferência em ontologias escritas em OWL-*Full* é da ordem de 2NEXPTIME-Complete.

[35] Utilizaremos o termo *conjunctive query* e seus derivativos em inglês por se tratar de um termo já consolidado na literatura

não adequada à descrição de qualidade de softwares, ontologias e de outras mais. Não é possível, por exemplo, saber se um software A tem qualidade baseado apenas em suas funcionalidades, pois o mesmo pode não ser "bom o suficiente" para os usuários que irão manuseá-lo. Nesse caso, Pressman [Pre11] define qualidade de software como *"uma gestão de qualidade efetiva aplicada de modo a criar um produto útil que forneça valor mensurável para aqueles que o produzem e para aqueles que o utilizam"*. É possível notar a preocupação em atribuir uma medida, um valor mensurável para o produto em questão, tanto pela ótica de quem o produz como de quem o utiliza. E essa preocupação já existe a, pelo menos, mais de 30 anos. Diversos autores já apresentaram propostas, ainda na década de 70 e 80, visando quantificar, definir ou apenas estabelecer critérios práticos que auxiliem na avaliação de qualidade de produtos de software [MRW77, Gar87].

No que diz respeito às ontologias, existe uma quantidade considerável de trabalhos dedicados à criação de métricas voltadas à avaliação de qualidade destas [ED13, BJSSA05, TAM+05, RCCG14]. Alguns trabalhos [BJSSA05, LTGP04] consideram que as ontologias são repositórios únicos, tantos de axiomas da base de terminologias (TBox) quanto da base de asserções (ABox), chamados nesse caso de ontologias **monolíticas**. Outros [WBHQ07, ED13] apresentam metodologias de avaliação baseados em métricas específicas para ontologias modulares, em que os subdomínios do conhecimento envolvidos na modelagem de um domínio mais abrangente estão separados em ontologias distintas, permitindo a manipulação e o reuso apenas das partes do conhecimento que interessam naquele momento.

Apesar de já existirem alguns estudos sobre a avaliação de ontologias, esta ainda permanece como uma área de pesquisa relativamente nova e com poucos resultados definitivos. Algumas razões contribuem para o fato das ontologias ainda não contarem com um conjunto formal e bem estabelecido de métricas de qualidade. Uma delas foi a expansão da Web Semântica, que popularizou e, com isso, aumentou a quantidade de técnicas e metodologias para a modelagem de ontologias [FLGPJ97, GF95, UK95, UG96, AH06]. Assim, durante algum tempo, o foco das pesquisas em ontologias ficou direcionado em "como fazer" uma ontologia, algo citado por [BJSSA05] como YAMA (*yet another modelling approach*). Outro fator preponderante para a falta de metodologias formais de avaliação de ontologias é a dificuldade de definição dos elementos que irão compor suas métricas. Segundo [Gua98], cada ontologia é modelada de acordo com um domínio de conhecimento específico e, dessa forma, atende a diferentes tipos de necessidades dos seus engenheiros do conhecimento. Isso faz com que cada ontologia possua seu próprio conjunto de métricas, dificultando a comparação entre essas metodologias. Neste trabalho, utilizamos um processo de avaliação das ontologias modeladas baseado no cumprimento dos propósitos para os quais elas foram criadas (conforme defendido em [NM01, FLGPJ97]) e no tempo de inferência gasto para extrair novos conhecimentos a respeito da Síndrome de *Li-Fraumeni*. No Capítulo 4, serão discutidas e apresentadas as metodologias utilizadas na construção da *Li-Fraumeni Ontology* e no Capítulo 6 serão descritas com mais detalhes as métricas utilizadas na avaliação das ontologias modeladas.

Capítulo 3

Projeto Ontofamily

O Projeto *Ontofamily* é um projeto de integração de dados clínicos sobre histórico familiar de pacientes através do uso de ontologias pra inferir e extrair conhecimento sobre a Síndrome de *Li-Fraumeni* no *A.C. Camargo Cancer Center*. Desde a sua fundação, em 1953, mais de 800.000 pacientes já foram tratados no *A.C. Camargo Cancer Center*. Em 2000, o hospital ganhou um Departamento de Oncogenética, aonde mais de 600 famílias foram assistidas no tratamento de câncer, 130 delas diagnosticadas clinicamente como portadora da Síndrome de *Li-Fraumeni* e 33 com mutação carcinogênica no gene $TP53$ (para maiores detalhes sobre o mecanismo de aparecimento do câncer, ver a Seção 2.1). Cada uma dessas famílias possui um histórico médico digital detalhado do probando[1], contendo informações da sua genealogia e da genealogia de sua família. Adicionalmente, existem muitos dados armazenados em diversas outras fontes de dados, como sistemas legados e de banco de dados do *A.C. Camargo Cancer Center*.

O principal objetivo do projeto *Ontofamily* é organizar e distribuir todo esse conhecimento aos pesquisadores, usando um sistema de integração de dados baseado em ontologias, permitindo a atualização desses dados e a adição futura de novas fontes de dados. O projeto também prevê a construção de uma ferramenta de extração e navegação de dados com a finalidade de garantir fácil acesso a estes pelos pesquisadores. Por meio dessa ferramenta e usando métodos estatísticos, será possível, futuramente, validar os critérios clínicos da Síndrome de *Li-Fraumeni* ou ajudar na descoberta de novos critérios clínicos.

O próximo tópico será dedicado à apresentação da Síndrome de *Li-Fraumeni* e dos critérios clínicos envolvidos no seu diagnóstico. O estudo desses critérios é fundamental para a compreensão da análise clínica da síndrome bem como para a formalização dos axiomas usados na classificação dos pacientes como portador da síndrome. Para uma melhor compreensão dos conceitos relativos ao câncer e ao mecanismo de formação do tumor, bem como os tipos de tumor descritos nesta seção, recomendamos a leitura prévia da Seção 2.1.

3.1 A Síndrome de Li-Fraumeni

A Síndrome de *Li-Fraumeni* é uma síndrome rara de predisposição hereditária ao câncer que aumenta consideravelmente o risco do paciente desenvolver múltiplos tumores, como o câncer de mama em mulheres jovens, câncer no cérebro, na supra-renal e sarcomas em idade jovem [AOC+07].

[1] Indivíduo com o qual se inicia o estudo familiar de uma doença ou característica genética.

Ela é causada por uma mutação no gene $TP53$, que é considerado um gene supressor de tumores [Jor04], onde sua principal função é preservar a integridade do genoma, controlar o crescimento celular e evitar o aparecimento de tumores. Indivíduos portadores de mutações germinativas no gene $TP53$ o recebem alterado de seus progenitores e podem passá-lo a 50% dos seus descendentes.

O aparecimento da Síndrome de *Li-Fraumeni* começou a ser estudado entre os anos de 1960 e 1964 por meio dos pesquisadores Li e Fraumeni, que, após a analisar de mais de 200 prontuários médicos e 418 obituários de crianças que faleceram de rabdomiossarcoma, identificaram, dentre esses dados, 5 famílias que apresentaram um comportamento incomum quanto ao aparecimento de sarcomas de tecido mole em um segundo filho de uma mesma família. Eles também identificaram nesse mesmo grupo de estudo, a ocorrência de diversas outras formas de câncer em indivíduos que mantinham grau de parentesco com o probando em primeiro e segundo grau [Mal11]. Surgiu, a partir daí, a definição do primeiro conjunto de critérios clínicos, chamado de Critério Clássico, da Síndrome de *Li-Fraumeni* (LFS), e que serão abordados nas próximas subseções.

À medida que novas famílias foram sendo estudadas, novos tipos de tumores foram sendo relacionados com a Síndrome de *Li-Fraumeni*. Assim, com o objetivo de deixar o conceito da Síndrome de *Li-Fraumeni* mais amplo, novos critérios foram surgindo ao longo do tempo. Primeiramente, em 1994, Birch [BHTP94] avaliou 24 famílias e estabeleceu novos critérios clínicos para a Síndrome de *Li-Fraumeni*, levando em consideração a ausência do fenótipo completo e a presença de tumores típicos da síndrome (sarcoma, câncer de mama, tumor no cérebro, leucemia, tumor adrenocortical), sendo chamados de critérios de Birch. No ano seguinte, Eeles [Eel95] expandiu os critérios estabelecidos por Birch, englobando, também, aquelas famílias com histórico de mais de um tumor pertencente ao espectro *Li-Fraumeni*, que também foi ampliado (sarcoma, câncer de mama, tumor no cérebro, leucemia, tumor adrenocortical, melanoma, câncer de próstata, câncer no pâncreas), ou com múltiplos tumores, independentemente do seu histórico familiar. Os critérios de Birch e Eeles também são conhecidos como critérios *Li-Fraumeni Like* (LFL). Como o conjunto de critérios Clássicos incluía apenas pacientes que apresentaram sarcoma e incluía poucas informações sobre o seu histórico familiar, Chompret [FABP+01] apresentou um conjunto de novos critérios mais abrangentes, que incluía novos tipos de tumores e mais informações sobre o seu histórico familiar, que foram chamados de critérios de Chompret. Mais tarde, esses critérios foram revistos e atualizados por Tinat *et al.* e chamados de Chompret revisado [TBBD+09]. Os quatro critérios serão melhor descritos posteriormente, na Seção 3.2.

No Brasil, uma variação incomum da mutação do gene $TP53$ é descrita em [AOC+07]. A hipótese da origem da Síndrome de *Li-Fraumeni* está associada à ocorrência de uma mutação específica que ocorre em até 0,3% da população na região Sul e Sudeste. Por meio de estudos genéticos, foi identificado um ancestral comum que originou essa mutação, possivelmente de origem portuguesa [Ach08]. Atualmente, a Agência Internacional para Pesquisas do Câncer (IARC[2]) relatou aproximadamente 767 famílias em todo o mundo como portadoras da mutação no gene $TP53$ (*IARC* $TP53$ *Germline dataset statistics, release R17*[3]) desde sua descoberta, em 1967, o que a faz uma síndrome rara.

Apesar da existência de quatro conjuntos de critérios de classificação (*Classic*, Birch, Eeles e Chompret) para pacientes suspeitos de serem portadores da mutação do gene $TP53$, apenas o

[2] http://www.iarc.fr/
[3] http://p53.iarc.fr/GermlineGrowthStats.aspx

teste genético pode emitir um laudo cujo diagnóstico atesta, com 100% de certeza, a presença dessa mutação. A decisão de realizar o teste genético é de cunho pessoal e deve ser feito após aconselhamento genético, pois seus resultados, caso sejam positivos, podem afetar a qualidade de vida do paciente e dos seus parentes mais próximos. Os critérios de classificação *Classic*, Chompret, Birch e Eeles servem como um primeiro diagnóstico clínico realizado pelo profissional especializado, apesar do fato de que nenhum dos critérios descritos possuem precisão de 100% nos diagnósticos (Tabela 3.1). Um estudo mais recente ([GNB+09]) sugere alta prevalência da mutação do gene $TP53$ em pacientes que atendem a determinados critérios: 100% dos pacientes afetados por tumores pediátricos com um membro da família afetado por pelo menos um tumor do tipo câncer de mama, sarcoma, câncer de cérebro ou carcinoma adrenocortical possuíam mutação no gene $TP53$. Outro percentual indica que 88% dos pacientes com dois ou mais tumores de mama, sarcoma, câncer de cérebro ou carcinoma adrenocortical, com pelo menos um deles diagnosticado antes dos 40 anos e dois ou mais familiares com algum dos tumores supracitados também possuem mutação no gene $TP53$. Apesar da alta prevalência, Gonzalez ([GNB+09]) realizou uma análise dos critérios quanto a capacidade de predição da mutação do gene $TP53$. Esse estudo concluiu que o critério Clássico (LFS) é mais preditivo que os outros critérios (apesar desse valor ser de apenas 56%), com alta especificidade[4] (91%) e baixa sensibilidade[5] (40%) (Ver Tabela 3.1). A seguir, serão apresentados detalhadamente cada um dos conjuntos de critérios da Síndrome de *Li-Fraumeni*.

Critério	Especificidade	Sensibilidade
Classic	91%	56%
Birch	38%	16%
Eeles	16%	14%
Chompret	52%	35%

Tabela 3.1: *Sumário da Especificidade e da Sensibilidade de cada um dos critérios Li-Fraumeni. Adaptado de [GNB+09].*

3.2 Critérios Clínicos para classificação da Síndrome de Li-Fraumeni

No passado, o diagnóstico da Síndrome de *Li-Fraumeni* era feito exclusivamente a partir de avaliação dos tumores ocorridos no paciente e em seus familiares. Esse diagnóstico não era fácil em razão da dificuldade em analisar todos os critérios clínicos aplicados ao paciente juntamente com informações sobre a sua genealogia. Além disso, nem sempre o paciente ou seus familiares que o acompanhavam possuíam informações detalhadas sobre os antepassados.

3.2.1 Critério Clássico

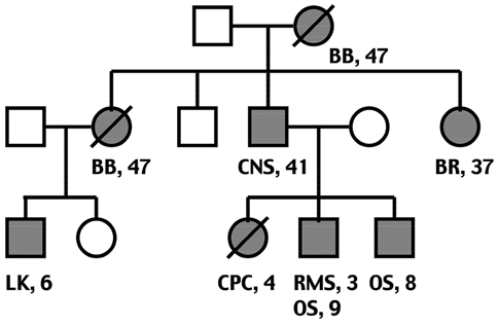
Com o objetivo de identificar características comuns a pacientes portadores da mutação no gene $TP53$, Li e Fraumeni iniciaram, em 1969, um estudo com quatro famílias [LJ69], aparentemente

[4]Especificidade é a capacidade que um teste possui de confirmar a negativa de uma determinada característica (uma doença, por exemplo) para uma amostra de suspeitos confirmadamente negativos.

[5]Sensibilidade é a capacidade que um teste possui de confirmar a positiva de uma determinada característica (uma doença, por exemplo) para uma amostra de suspeitos confirmadamente positivos.

portadoras de algum tipo de síndrome familiar, aonde alguns membros dessa família desenvolviam sarcomas de tecido mole e câncer de mama, além de outros tipos de tumores. O estudo constatou a presença de crianças com graus de parentesco de primeiro grau (irmãos) e de terceiro grau (primos) que apresentavam sarcoma de tecido mole e, cujas mães, apresentavam câncer de mama abaixo dos 30 anos (três mães), e leucemia mieloide aguda e câncer de pele (2 pais). Li e Fraumeni sugeriram, à época, que existia algum mecanismo oncogenético capaz de ser transmitido verticalmente entre gerações.

Figura 3.1: *Árvore genealógica de uma família com Síndrome de Li-Fraumeni.*



Círculos e quadrados preenchidos representam indivíduos afetados; barras representam indivíduos falecidos. Números representam a idade do paciente no diagnóstico. BB = câncer de mama bilateral; CNS = tumor de cérebro; BR = câncer de mama unilateral; LK = leucemia; CPC = carcinoma de plexo coroide; RMS = rabdomiossarcoma; OS = osteosarcoma. Extraído de [Mal11]

Posteriormente [LFM+88], eles conduziram estudos em 24 famílias que apresentavam características semelhantes àquelas das famílias descritas no primeiro estudo, com um espectro maior de tumores, expandindo, assim, a literatura existente até o momento sobre síndromes familiares de câncer e estabelecendo um conjunto de tumores cujo aparecimento poderia indicar a presença de dessa síndrome (a Figura 3.1 mostra um exemplo de uma família com a Síndrome de *Li-Fraumeni*). Eles estabeleceram, assim, um conjunto de critérios para a avaliação clínica de pacientes portadores da Síndrome de *Li-Fraumeni*: (1) paciente que tenha apresentado algum tipo de sarcoma antes dos 45 anos, (2) um parente de primeiro grau com qualquer tipo de câncer antes dos 45 antes e (3) um segundo parente próximo (primeiro ou segundo grau) que também tenha apresentado ou um sarcoma, não importando a idade, ou qualquer tipo de câncer antes dos 45 anos. Esses critérios são conhecidos como critérios Clássicos da Síndrome de *Li-Fraumeni* (LFS) e estão descritos conforme Tabela 3.2.

Critério	Situação
1	Paciente diagnosticado com sarcoma antes dos 45 anos e;
2	Um parente de primeiro grau ser diagnosticado com qualquer tipo de câncer antes dos 45 anos e;
3	Um outro parente de primeiro grau ou segundo grau diagnosticado com qualquer tipo de câncer antes dos 45 anos ou com um sarcoma em qualquer idade

Tabela 3.2: *Conjunto de critérios Clássicos para o diagnóstico da LFS.*

3.2.2 Critério Chompret

Observando o conjunto de critérios Clássico, nota-se que os eles são inclusivos apenas para pacientes que desenvolvem sarcomas, deixando de fora outros tipos de tumores. Além disso, o histórico familiar do paciente é restritivo, excluindo aqueles paciente que não possuem muitas informações sobre seus familiares. Para contornar essas especificidades do critério *Classic*, Chompret [FABP+01] apresentou um conjunto de critérios mais abrangentes, que incluíam pacientes que possuem outros tipos de tumores, não somente sarcomas. O espectro de tumores considerados pelo critério de Chompret nesse estudo são sarcoma, tumor de cérebro, câncer de mama e carcinoma adrenocortical. A idade do probando também foi reduzida em relação ao critério *Classic*, permitindo incluir mais pacientes como possíveis portadores da Síndrome de *Li-Fraumeni*. A Tabela 3.3 apresenta essa versão do critério de Chompret.

Critério	Situação
1	Um probando com um tumor do espectro LFS (sarcoma, tumor de cérebro, câncer de mama e carcinoma adrenocortical) antes dos 36 anos e que tenha pelo menos um parente de primeiro ou segundo grau com um tumor característico do espectro LFS (exceto câncer de mama caso o probando tenha tido câncer de mama) antes dos 46 anos ou;
2	Um probando com múltiplos tumores, 2 deles pertencentes ao espectro LFS aonde o primeiro deles antes dos 36 anos ou
3	Um probando com carcinoma adrenocortical, não importando a idade em que ocorreu nem o histórico familiares.

Tabela 3.3: *Critérios de Chompret para o diagnóstico da LFS*

Em 2009, Tinat *et al.* [TBBD+09], revisaram o critério de Chompret, ampliando o conjunto de tumores característicos da Síndrome de *Li-Fraumeni*, flexibilizando os critérios referentes ao histórico familiar do probando, incluindo tumores do plexo coroide (devido ao seu valor preditivo) e excluindo parentes com múltiplos tumores mamários. Não que este último critério não seja importante, mas foi concluído que o percentual de pacientes que atendiam a esse critério específico era muito baixo ($\leq 5\%$) se comparado ao elevado nível de ansiedade e o desconforto causados pela espera do resultado de um exame de sequenciamento genético. Na Tabela 3.4 é possível conferir o critério de Chompret revisado.

3.2.3 Critério Birch

Apesar da maioria das famílias analisadas por Li e Fraumeni apresentarem pacientes com as mesmas características descritas nos seus trabalhos, Birch ([BHTP94]) observou a existência de algumas famílias que não apresentavam todas as características descritas por Li `et al.` ([LJ69],[LFM+88]) mas que sugeriam a presença da Síndrome de *Li-Fraumeni*. Birch, então, procedeu um estudo com 21 famílias, 12 delas foram selecionadas conforme os critérios LFS e 9 delas segundo um critério mais abrangente, que foi chamado de Li-Fraumeni-*like* (LFL). Segundo o autor, esses critérios foram definidos segundo as características de famílias que atendiam aos critérios Clássicos, juntamente com o aumento da idade máxima do primeiro diagnóstico de 45 para 60 anos e uma ampliação do espectro de tumores característicos da síndrome, principalmente aqueles tumores pediátricos (rabdomiossarcoma, por exemplo).

Um probando com	
1	Tumor pertencente ao espectro de tumores LFS - sarcoma de tecido mole, osteosarcoma, câncer da mama pré-menopausa, tumor cerebral, carcinoma adrenocortical, leucemia ou câncer bronco-pulmonar - antes de 46 anos de idade
E	
	Pelo menos um parente de primeiro ou segundo grau com um tumor do espectro de tumores LFS (exceto câncer de mama se o probando tem câncer de mama) antes de 56 anos de idade **OU** qualquer parente do probando com múltiplos tumores primários em qualquer idade;
OU	
2	Um probando com vários tumores (exceto múltiplos tumores mamários), dois deles pertencentes ao espectro de tumores LFS e o primeiro ocorrido antes dos 46 anos de idade;
OU	
3	Um probando que é diagnosticado com carcinoma adrenocortical ou tumor do plexo coroide, independente da história familiar;

Tabela 3.4: *Critérios de Chompret revisado para o diagnóstico da Síndrome de Li-Fraumeni*

O estudo confirmou que o critério *Classic* da Síndrome de *Li-Fraumeni* definiram, de maneira mais precisa, as famílias que possuíam mutações no gene $TP53$ (6 confirmadas das 12 famílias que atendiam aos critérios Clássicos) se comparadas às famílias que atendiam aos critérios definidos por Birch (1 confirmada das 9 famílias). O estudo também identificou que 5 das 6 famílias selecionadas pelos critérios Clássicos e que foram confirmadas com mutação no gene $TP53$ tinham crianças que desenvolveram rabdomiossarcoma. Também dessas 6 famílias, foram identificadas 3 delas com crianças que desenvolveram carcinoma adrenocortical. A única família selecionada usando os critérios de Birch e identificada com mutação no gene $TP53$ também apresentou uma criança que desenvolveu carcinoma adrenocortical. Birch concluiu, então, que rabdomiossarcomas e carcinomas adrenocortical, apesar de não serem descritos nos critérios Clássicos, representam fatores na presença de mutações no gene $TP53$. A Tabela 3.5 apresenta os critérios descritos por Birch nos seus estudos.

Critério	Situação
1	Um probando com qualquer tipo de câncer na infância ou sarcoma, tumor cerebral ou carcinoma adrenocortical diagnosticados antes dos 45 anos;
2	Um parente de primeiro grau ou segundo grau com um tumor maligno típico de LFS (sarcoma, leucemia ou câncer de mama, cérebro ou córtex adrenal), independente da idade no momento do diagnóstico
3	Um parente de primeiro grau ou segundo grau com qualquer câncer diagnosticado antes dos 60 anos;

Tabela 3.5: *Critérios de Birch para o diagnóstico da LFL*

3.2.4 Critério Eeles

O critério de Eeles [Eel95] também é conhecido por ser um critério mais relaxado e incluir pacientes que não apresentam todas as características dos critérios Clássicos mas que podem sugerir

a presença da mutação no $TP53$. Juntamente com os critérios Clássicos e de Birch, são largamente usados em estudos clínicos. O critério de Eeles, por ser mais relaxado, também é classificado como Li-Fraumeni-*Like*. A Tabela 3.6 apresenta o critério de Eeles, aonde podemos notar que não é dada importância ao tipo de câncer desenvolvido pelo probando, mas sim àqueles desenvolvidos nos seus parentes de primeiro ou segundo graus.

Critério	Situação
1	Dois parentes de primeiro ou segundo grau com tumores típicos de LFS (sarcoma, leucemia ou câncer de mama, cérebro ou córtex adrenal) em qualquer idade;

Tabela 3.6: *Critérios de Eeles para o diagnóstico da LFL*

Nota-se, com isso, que, dada a quantidade de condições existentes em cada um dos critérios descritos, o diagnóstico de um paciente e de seus familiares pode vir a tomar muito tempo e sofrer de erros humanos. Assim, na próxima seção, apresentaremos um conjunto de ontologias que formaliza os critérios apresentados anteriormente para que seja possível automatizar a classificação de pacientes segundo esses critérios.

Capítulo 4

Metodologia para construção da *Li-Fraumeni Ontology*

Conforme apresentado na Seção 2.2.2, diversas são as metodologias existentes para construção de ontologias. Cada uma delas possui um conjunto de ferramentas e técnicas que as tornam apropriadas para determinados domínios de conhecimento. A TOVE e a Enterprise são apropriadas para o domínio industrial e empresarial; a Methontology foi construída e validada no domínio da Química e, posteriormente, na área jurídica; a UPON foi aplicada visando a área da gestão da cadeia de suprimentos. Isso não impede, entretanto, que essas metodologias sejam aplicadas na construção de uma base de conhecimentos em outras áreas, como é o caso da Biomedicina. Nesse cenário, apresentamos a metodologia empregada na construção da *Li-Fraumeni Ontology*, os problemas enfrentados e as soluções encontradas durante a sua construção. Apresentaremos, também, uma proposta de solução para o problema do desenvolvimento colaborativo de ontologias e que foi essencial para a viabilização da *Li-Fraumeni Ontology*.

Por se tratar de uma ontologia com múltiplos domínios do conhecimento (biomédico, humanas), os critérios levados em consideração quanto à escolha da metodologia de construção da *Li-Fraumeni Ontology* foram:

1. Agilidade: Por se tratar de uma ontologia de aplicação, a sua construção não poderia se estender muito além do tempo levado para construção do software que faria uso da sua estrutura. Além do que, para minimizar os riscos de inconsistências, consideramos importante ciclos de lançamento de versões mais curtos.

2. Colaboração: As equipes de desenvolvimento eram multidisciplinares, ou seja, eram compostas por médicos, biomédicos, e profissionais da TI, e tinham regimes de trabalho, disponibilidades e entendimentos distintos sobre alguns subdomínios da Síndrome de *Li-Fraumeni*. Era essencial, portanto, que a metodologia permitisse que a construção da ontologia ocorresse de forma incremental e assíncrona, por meio de versionamentos das ontologias envolvidas;

3. Simplicidade: Usamos o conceito de simplicidade para conceitualizar o conjunto de práticas "desburocratizadas", ou seja, sem excessos de papéis, formulários, diagramas ou regras impostas para a criação da ontologia.

Levando em conta os critérios acima expostos e as metodologias abordadas na Seção 2.2.2, escolhemos a METHONTOLOGY, a UPON, a RapidOWL e a metodologia 101 para uma avaliação

mais detalhada. A escolha das metodologias foi direcionada para que a análise não ficasse extensa desnecessariamente ao avaliarmos metodologias que, claramente, não se encaixavam nos critérios acima, como a Enterprise ou a TOVE, ambas direcionada para o ambiente empresarial. Para cada metodologia, avaliamos os três critérios acima de forma qualitativa. Ao final, apresentamos um quadro-resumo onde atribuímos os conceitos de relevância **alto**, **médio**, **baixo** e **ausente**.

4.1 Análise das metodologias.

4.1.1 METHONTOLOGY

Apesar da METHONTOLOGY ser uma metodologia mais antiga e ter sido criada anteriormente à Web Semântica, ela apresenta elementos simples que são capazes de nortear a construção de ontologias em qualquer subdomínio. Ela é fundamentada no uso de estágios de evolução definidos previamente por meio de respostas a questões de competências. Estas, por sua vez, são norteadas pelos objetivos traçados para o uso da ontologia a ser construída antes do início do seu desenvolvimento, na etapa de Planejamento (Figura 4.1). A METHONTOLOGY não obriga o uso de diagramas ou formulários predefinidos, apesar de sugerir modelos de documentos, como as especificações sobre necessidades para a construção de ontologias (*Ontology requirements specification*). Mesmo as etapas e os estágios sugeridos como parte do ciclo de vida das ontologias não precisam seguir uma sequência pré-definida, excluindo-se, obviamente, o Planejamento e a Implementação, que devem ser, respectivamente, o inicio e o fim do processo.

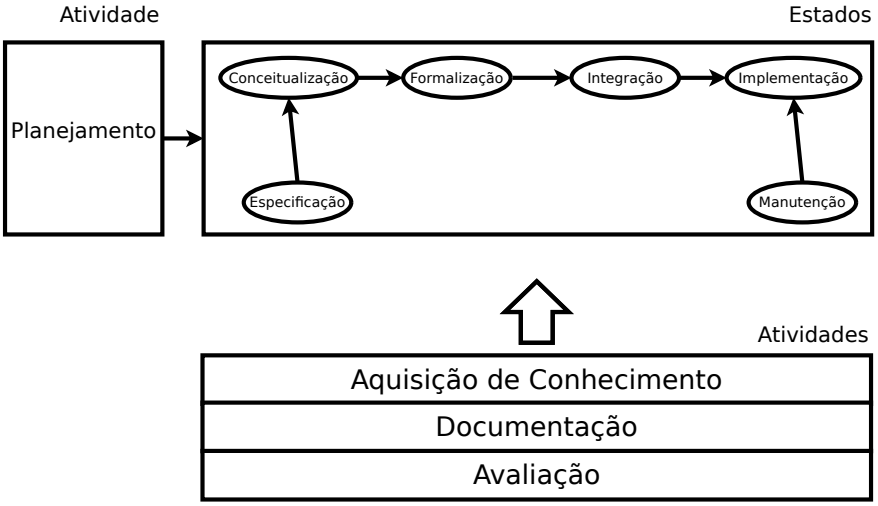
Assim, podemos avaliar a METHONTOLOGY baseado nos três critérios sugeridos:

1. **Agilidade**: Suas etapas e atividades podem ser tão curtas quanto se queira, ou seja, se definirmos etapas curtas de desenvolvimento, então as atividades serão concluídas, também, em curtos espaços de tempo. A não exigência de formulários e questionários ou até mesmo o uso de linguagem natural na especificação das questões de competência acelera o processo de planejamento da ontologia, mesmo que, futuramente, ela venha a ser atualizada com novos termos e conceitos.

2. **Colaboração**: Apesar da metodologia possuir uma etapa de integração com outras fontes de conhecimento, ela não está bem definida quanto ao desenvolvimento colaborativo (distribuído) de ontologias. Vale observar que as exigências inerentes à Web Semântica (formalismos, plataforma comum de compartilhamento de conhecimento, ...) não haviam sido descritas, à época, e, por essa razão, acreditamos não ter sido mencionada. Mesmo dessa forma, a METHONTOLOGY sugere o uso de relatórios estruturados de integração com outras fontes de conhecimento, que podem sugerir o uso combinado desta com alguma outra técnica de desenvolvimento colaborativo.

3. **Simplicidade**: Como já mencionado, a METHONTOLOGY não obriga o uso de documentação específica para a construção de ontologias, bem como não atribui uma sequência predefinida de etapas que devam ser cumpridas. Também, em cada fase do ciclo de vida, são descritos apenas os objetivos a serem alcançados durante sua execução, sem determinar como eles serão alcançados. Essa flexibilidade permite adaptações da metodologia, como a fusão com outras técnicas de documentação, de testes ou mesmo a não execução de determinada etapa do processo de desenvolvimento por razões outras.

Figura 4.1: *Diagrama de atividades da METHONTOLOGY. Adaptado de [FLGPJ97].*



Concordamos, após essa análise, com o exposto por Fernández-López em [FL99], que diz:

> *[...] None of the methodologies are fully mature if we compare them with the IEEE standard; [...] METHONTOLOGY is the most mature; however, recommendations for the predevelopment processes are needed, and some activities and techniques should be specified in more detail [...]*

O padrão IEEE mencionado acima se refere ao IEEE 1074-1995[1], que descreve os padrões para o ciclo de vida de desenvolvimento de software e que foi utilizado como base e modelo de comparação pelo autor. Também corroboramos com a ideia de que os processos (principalmente as etapas de integração e colaboração) devam ser melhor especificadas, com sugestão de ferramentas que auxiliem o essa etapa.

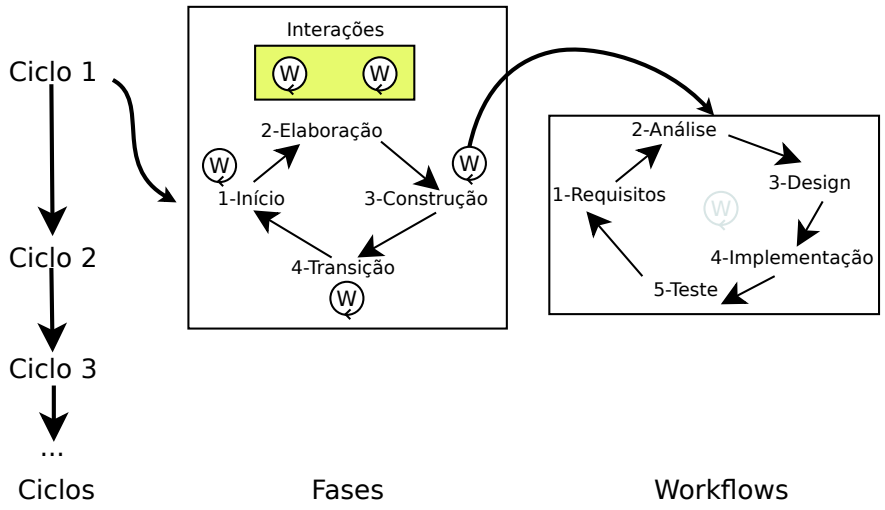
4.1.2 UPON - *Unified Process ONtology building*

A UPON foi desenvolvida em 2005 derivada da metodologia UP - *Unified Process*, uma metodologia madura oriunda da Engenharia de Software, e que faz uso do conjunto de ferramentas de diagramação da UML - *Unified Modeling Language*, como diagramas de caso de uso e diagramas de classe. A principal contribuição da metodologia UPON, segundo [DNMN09], é fornecer um conjunto robusto, maduro e consistente de ferramentas para a construção, não somente de ontologias de domínio, mas também de ontologias de aplicação. O foco da UPON é orientar o processo de desenvolvimento de ontologias de grande porte, mas podendo ser útil, também, em ontologias de pequeno porte.

A condução da metodologia se fundamenta, basicamente, em ciclos de execução, que podem terminar por lançar uma nova versão da ontologia em questão. Cada ciclo é composto por um conjunto de quatro fases; cada fase possui um conjunto de interações que, por sua vez, compreende um conjunto de *workflows*. A Figura 4.2 exemplifica de forma esquemática e genérica as etapas da metodologia UPON.

[1] https://standards.ieee.org/findstds/standard/1074-1995.html

Figura 4.2: *Diagrama esquemático da metodologia UPON. Adaptado de [DMN05].*



Observa-se que a metodologia UPON possui um grande número de etapas e requer uma sequência pre-determinada de passos a serem seguidos. Os ciclos determinam as versões principais de uma ontologia a serem lançadas. Em cada ciclo, é necessário cumprir 4 fases de desenvolvimento, cada fase com uma quantidade específica de interações. Em cada interação existe um conjunto de 5 *workflows* que permitem que partes da tarefa da engenharia do conhecimento ou da construção de ontologias sejam alcançadas gradativamente, como a conceitualização, as questões de competência, os requisitos iniciais ou até mesmo os testes. Se considerarmos, por exemplo, 10 ciclos de desenvolvimento (10 versões estáveis lançadas), 2 interações em cada fase, cada fase com 5 *workflows* e cada *workflow*, em média, com 5 atividades (quantidade de atividades em cada *workflow* não é fixa), teremos, então, para cada ciclo de lançamento, um total de 4 fases, 8 interações, 40 *workflows* e mais de 100 atividades. Ao todo, durante os 10 ciclos, foram executadas 40 fases, 80 interações, 400 execuções de *workflows*, num total de 2000 atividades. Uma metodologia com uma grande quantidade de atividades, fases e interações pressupõe uma equipe relativamente grande e bem organizada, com nítida separação de tarefas e composta por analistas, engenheiros do conhecimento e especialistas de domínio. Para equipes pequenas ou de apenas um analista/especialista de domínio/engenheiro do conhecimento, a aplicação da metodologia tende a ser excessivamente trabalhosa e burocrática.

Por se tratar de uma metodologia baseada na UP, a UPON pode ser aplicada na modelagem de qualquer ontologia, seja ela de domínio, meta-ontologias, de aplicação ou de tarefas, e em qualquer domínio do conhecimento (biomédico, engenharia, industrial, química, ...). Essa escalabilidade também é extensível ao tamanho das ontologias, à sua complexidade e ao nível de conhecimento das equipes de desenvolvimento quanto à metodologia UP. Outro detalhe importante se refere à flexibilidade de uso das etapas da metodologia UPON. Apesar de poder ter suas etapas, fases e atividades adaptadas a diferentes cenários [DNMN09], essa facilidade de adaptação se limita a metodologias oriundas da UP, restringindo razoavelmente essa flexibilidade.

Assim, segundo os critérios sugeridos, classificamos a UPON conforme segue:

1. **Agilidade**: Apesar da metodologia ser baseada em um método maduro de Engenharia de Software, o mesmo não pode ser considerado ágil devido à quantidade de etapas do ciclo

de desenvolvimento da ontologia. O tempo de lançamento de uma versão pode se tornar excessivamente longo e burocrático.

2. **Colaboração**: A metodologia UPON não menciona, em sua descrição, nenhum mecanismo ou ferramenta voltados para o desenvolvimento colaborativo. Acreditamos que a metodologia possa ser utilizada de maneira combinada com outras ferramentas para esse fim, como o IBM *Rational Team Concert*[2].

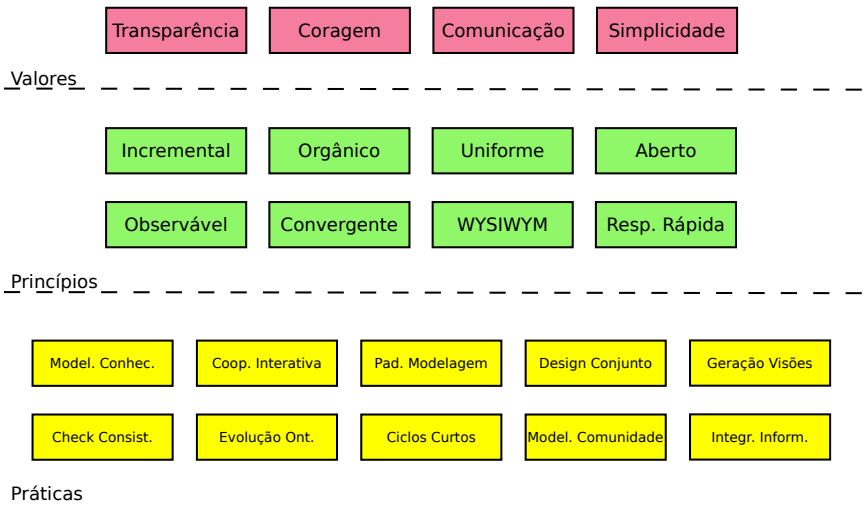
3. **Simplicidade**: Devido à quantidade de fases e ciclos de interação, consideramos a UPON uma metodologia complexa, com muitos diagramas e muita exigência de documentação. Esse pode ser um critério impeditivo para o desenvolvimento ágil de ontologias.

4.1.3 RapidOWL

Seguindo a tendência das metodologias ágeis, a RapidOWL tem como objetivo tornar mais eficientes as tarefas de elicitação, processamento do conhecimento e comunicação entre especialistas de domínio e engenheiros do conhecimento. Suas práticas foram inspiradas nas técnicas de XP (*eXtreme Programming*) aplicadas à área da Engenharia do Conhecimento.

Diferentemente das metodologias tradicionais, as metodologias ágeis acordam mais importância ao dinamismo das mudanças dos dados e do conhecimento do que à rigidez dos fluxogramas e do ciclo de vida. Esse paradigma permite que a RapidOWL seja voltada basicamente a princípios ágeis de modelagem, como valores a serem mantidos e princípios de modelagem a serem observados, deixando de lado a descrição rígida de fases e *workflows*. A Figura 4.3 apresenta um esquema ilustrativo dos princípios, valores e práticas a serem adotadas pela metodologia.

Figura 4.3: *Diagrama esquemático da metodologia RapidOWL. Adaptado de [AH06].*



Os valores perseguidos pela RapidOWL são os mesmos valores definidos pela *eXtremeProgramming* e que representam os objetivos perseguidos pelos desenvolvedores durante o processo de construção da ontologia. Esses valores norteiam os princípios da metodologia e que devem estar presentes

[2] *Rational Team Concert é uma ferramente de desenvolvimento colaborativo para equipes da IBM http://www-03.ibm.com/software/products/pt/rtc*

na ontologia. As práticas devem ser conduzidas a fim de alcançar os princípios estabelecidos e mantendo os valores sempre em um horizonte visível. Essa combinação, segundo Auer *et al.* [AH06], evita a criação de práticas rígidas de modelagem e permite flexibilidade ao processo de modelagem. Segundo os critérios definidos anteriormente, temos:

1. **Agilidade**: Pela própria natureza dos princípios que norteiam a metodologia, a RapidOWL é considerada ágil, com ciclos curtos de desenvolvimento e uma prática voltada à gestão rápida de riscos, evitando altos custos de correção. A ausência de um fluxo rígido de etapas a serem seguidas pode ser considerado um grande diferencial em relação às demais metodologias.

2. **Colaboração**: Por ser uma metodologia essencialmente ágil, já carrega consigo os valores e princípios para o desenvolvimento colaborativo das ontologias. Apesar de não descrever nem estabelecer explicitamente um conjunto de ferramentas para esse fim, pode se beneficiar da existência de ferramentas e técnicas desenvolvidas especificamente para as metodologias ágeis como SCRUM e XP. Técnicas de versionamento também podem ser incorporadas a fim de estabelecer ciclos curtos e controlados de lançamento de novas versões estáveis da ontologia.

3. **Simplicidade**: A simplicidade é considerada um dos valores básicos da metodologia ágil (Figura 4.3) e, portanto, está presente na RapidOWL. Ela não obriga a documentação através de formulários, questionários ou diagramas pois isso poderia atrasar o processo de desenvolvimento e, com isso, estender o tempo de conclusão dos ciclos de desenvolvimento.

4.1.4 Guia para o Desenvolvimento de Ontologias 101

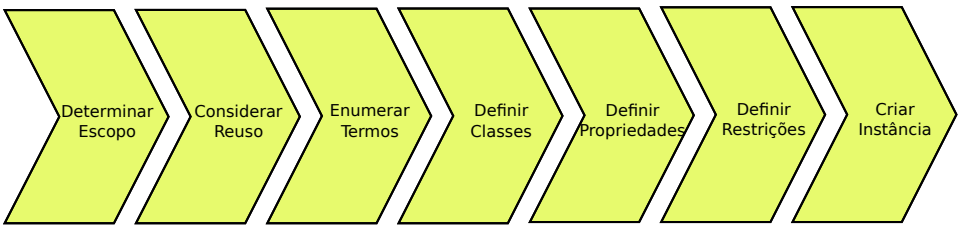
A metodologia de Noy & McGuiness [NM01] (que chamaremos, deste ponto em diante, de metodologia 101) é considerada uma das mais conhecidas e simples para construção de ontologias. Seu principal objetivo é fornecer um guia passo a passo, para novatos, visando a construção de ontologias. A metodologia 101 estabelece alguns conceitos essenciais para a construção de ontologias, como a ausência de uma metodologia única capaz de ser considerada correta em detrimento de outras. Apesar da metodologia ter sido criada há mais de quatorze anos, esse princípio continua sendo verdade em tempos atuais, mesmo com o aparecimento de diversas outras metodologias bem mais completas do que as existentes à época. A bem da verdade, o que existe são situações diversas em que algumas metodologias são mais adequadas do que outras para a resolução de alguns tipos de problemas e, com isso, acabam por ser mais frequentemente utilizadas.

A metodologia 101 sugere, assim, um conjunto de sete passos que abordam desde a construção dos requisitos até a sua implementação, passando pelas etapas de conceitualização, refinamento e integração com outras ontologias (ver Figura 4.4). Um fato importante nesse contexto é que a linguagem OWL ainda não havia sido lançada em 2001 e, por essa razão, a metodologia 101 não menciona, em nenhum momento, o uso de linguagens para a construção das ontologias. A consequência desse fato pode ser vista sob dois aspectos divergentes: *(i)* fator positivo, pois a metodologia pode ser considerada flexível, na medida em que não estabelece uma linguagem específica para a formalização da base de conhecimentos e, com isso, permite que ela seja facilmente adaptada para futuros padrões ou recomendações de linguagem, ou; *(ii)* fator negativo, pois a ausência da linguagem OWL reflete o fato de que a metodologia não levou em consideração os princípios da Web Semântica e, consequentemente, esteja desalinhada com os atuais fundamentos de formalização e compartilhamento de conhecimento estabelecidos pela Web Semântica e recomentados pela W3C.

Uma importante contribuição da metodologia 101 foi o uso de **questões de competência** para nortear o processo de conceitualização e para ser utilizado, posteriormente, como instrumento de avaliação do produto final (no caso, a ontologia). Questões de competência representam os requisitos que uma ontologia deve possuir, ou seja, são o conjunto de perguntas que uma ontologia deve ser capaz de responder após ter sido concluída [NM01]. Já Robert Steven (http://studentnet.cs.manchester.ac.uk/pgt/2014/COMP60421/slides/Week2-CQ.pdf) diz que as questões de competência ajudam a capturar o escopo, o conteúdo e a avaliação da ontologia em questão. Quem primeiro definiu seu conceito foi Gruninger & Fox [GF95], separando as questões de competência formais das informais e utilizando-as para testar a completude da ontologia na sua etapa final de desenvolvimento. As questões de competência informais são aquelas relacionadas com o cenário que motivou a criação da ontologia. Já as questões de competência formais são aquelas levantadas após a construção da terminologia da ontologia e diz respeito a todos os axiomas formalmente estabelecidos para essa determinada ontologia. A metodologia 101 simplificou o uso das questões de competência na medida em que definiu seu uso apenas para estabelecer o conjunto de requisitos de uma ontologia.

Outro ponto importante na metodologia 101 diz respeito ao desenvolvimento colaborativo, que não é tratado. Apesar de já poder contar com a Internet para disseminação de dados e informações na época da sua publicação, a metodologia 101 não levou em consideração os princípios estabelecidos para a Web Semântica, como plataforma comum de distribuição das ontologias. Essa lacuna abre espaço para o uso de qualquer outra metodologia que auxilie o desenvolvimento colaborativo, como o OntoMaven.

Figura 4.4: *Conjunto de sete passos da metodologia 101. Adaptado de [NM01].*



Portanto, classificamos a metodologia 101 conforme descrito abaixo:

1. **Agilidade**: Apesar de não levar em conta as metodologias ágeis para a criação de ontologias, os seus ciclos de desenvolvimento pode ser considerados de velocidade moderada devido à quantidade de passos. Apesar da metodologia reconhecer que o desenvolvimento da ontologia deva seguir o modelo incremental, ela não define claramente o seu ciclo de vida. Outro fator que conta negativamente para a agilidade é que as etapas a serem seguidas durante a construção da ontologia são dependentes umas das outras e segue um modelo sequencial, ou seja, a saída de uma etapa serve de entrada para a etapa seguinte. Esse fato impede que o fluxo de construção da ontologia seja adequado diferentemente daquele estabelecido pela metodologia.

2. **Colaboração**: A metodologia 101 não descreve nenhuma ferramenta ou mecanismo para o desenvolvimento colaborativo de ontologias. Essa lacuna permite que outras abordagens sejam utilizadas, como o uso do OntoMaven ou de outras ferramentas de versionamento e de armazenamento compartilhado.

3. **Simplicidade**: A metodologia é considerada simples na medida em que não enrijece a construção da ontologia com formulários, relatórios e documentações. Ela sugere o uso de técnicas de documentação e de análise dos requisitos, mas não atrela esses artefatos à condução da metodologia.

4.2 Comparativo das metodologias escolhidas

Apresentamos na Tabela 4.1 um resumo das metodologias avaliadas, onde atribuímos os conceitos de relevância **alto**, **médio**, **baixo** e **ausente** a cada uma delas baseado na análise feita na Seção 4.1. Usamos **alto** para indicar que a metodologia implementa mecanismos explícitos para determinado critério; **médio** quando a metodologia não indica, explicitamente, mecanismos para determinado critério, mas permite facilmente que algum outro mecanismo não pertencente à metodologia seja utilizado conjuntamente sem prejuízo para o andamento normal da modelagem da ontologia; **baixo** quando, além de não indicar nenhum mecanismo para determinado critério, a adaptação de algum outro mecanismo externo à metodologia é de difícil implementação e pode prejudicar a aplicação da metodologia e; **ausente** quando não existe nenhuma indicação explícita de nenhum mecanismo para determinado critério e também não permite a adaptação de outro mecanismo externo.

Tabela 4.1: *Tabela comparativa das metodologias analisadas segundo os critérios definidos no início do capítulo.*

		Critérios		
		Agilidade	Colaboração	Simplicidade
Metodologias	METHONTOLOGY	MÉDIO	AUSENTE	ALTO
	UPON	BAIXO	BAIXO	BAIXO
	RapidOWL	ALTO	MÉDIO	ALTO
	Guia 101	MÉDIO	AUSENTE	ALTO

Podemos notar que a UPON e a metodologia 101 tiveram os piores indicadores dentre as metodologias avaliadas. A metodologia 101 e a METHONTOLOGY, no computo geral, tiveram desempenho equivalente. Vale ressaltar que ambas são metodologias antigas, que precedem ao aparecimento da Web Semântica. A metodologia RapidOWL foi a única que obteve desempenho médio no quesito colaboração, se apresentando como uma boa opção.

De fato, o que observamos após essa análise é que nenhuma das metodologias avaliadas apresentou um conjunto de ferramentas capaz de fornecer um suporte direto ao desenvolvimento colaborativo de ontologias. Entretanto, a etapa de desenvolvimento das ontologias precisava desse suporte colaborativo, pois a validação de regras e a definição dos termos usados nas ontologias precisavam ser feitas por equipes que se encontravam geograficamente distantes. A solução encontrada, então, foi utilizar o que tem de melhor na METHONTOLOGY e na RapidOWL, quanto aos critérios

simplicidade e **agilidade**, e propor uma nova abordagem, mais simples, para o desenvolvimento colaborativo da *Li-Fraumeni Ontology*.

A seguir, apresentaremos a metologia utilizada para abordar o problema do desenvolvimento colaborativo de ontologias.

4.3 Desenvolvimento colaborativo da *Li-Fraumeni Ontology*

Durante a fase de conceitualização da *Li-Fraumeni Ontology*, os especialistas de domínio (*domain expert*), os engenheiros do conhecimento (*knowledge engineers*) e o construtor da ontologia estavam geograficamente distantes. Tornou-se necessário, então, utilizar uma metodologia de trabalho que permitisse a construção, a validação e o controle das diversas versões das ontologias que foram desenvolvidas no decorrer do tempo, sem perder a agilidade e a simplicidade necessários ao projeto.

Ao longo de um projeto de desenvolvimento colaborativo, consideramos como essenciais à implantação de um ambiente colaborativo, algumas funcionalidades, como o ***(i)* controle de versão**, ***(ii)* lista de tarefas**, ***(iii)* lista de e-mail** e ***(iv)* rastreamento de erros** (*bug tracker*). O controle de versão é uma ferramenta que permite aos desenvolvedores controlar os ciclos de lançamento de um determinado produto e também as mudanças no seu código-fonte. A lista de tarefas permite documentar as tarefas que serão concluídas e também as que já foram concluídas, servindo como fonte de documentação. Uma lista de e-mails permite que a comunicação entre os colaboradores seja feita de forma mais rápida e prática. Por fim, um sistema de rastreamento de erros auxilia no controle do lançamento de versões e documenta as mudanças de código, que pode ser essenciais, sem importância, melhorias e urgentes.

Para abordar as funcionalidades *(ii)*, *(iii)* e *(iv)*, utilizamos uma conhecida plataforma gratuita, que reúne todas essas funcionalidades, chamada BitBucket[3]. O BitBucket permite o uso de um repositório Git[4] gratuito e privado para pequenas equipes (na versão paga, permite um número ilimitado de colaboradores por equipe). A escolha pelo BitBucket levou em consideração a facilidade de gerenciamento do projeto, o uso da plataforma Git e, principalmente, por ser a única solução gratuita a oferecer repositórios privados. A Figura 4.5 mostra a tela inicial do repositório Git, no BitBucket, configurado para o desenvolvimento da *Li-Fraumeni Ontology*.

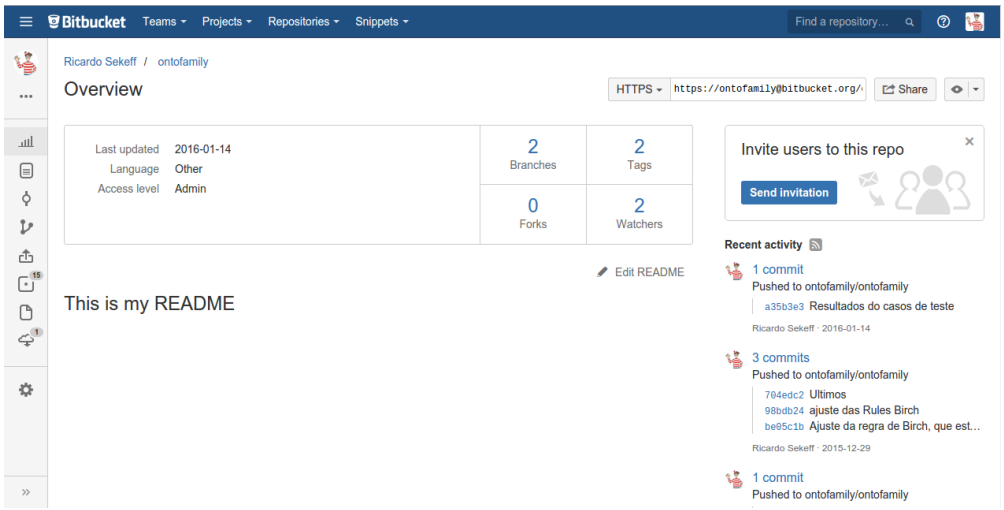
A respeito do controle de versão (funcionalidade *(i)*), preferimos não utilizar as ferramentas da plataforma BitBucket e acabamos implementando um conjunto de práticas que permitiram o lançamento de novas versões e o controle das mudanças efetuadas em cada uma delas. Decidimos por essa abordagem porque um dos domínios utilizados na *Li-Fraumeni Ontology* (CDOnto) tem um histórico de mudanças ao longo do tempo muito significativo, que não nos permite desconsiderá-los em versões mais recentes. Uma delas ocorreu na mudança de versão da ICD-9 para a ICD-10. Segundo [AMA14], a mudança de versão (ICD-9 para ICD-10) não é fácil e pressupõe análise detalhada dos prontuários e procedimentos médicos. Já o portal MedicAid.gov[5] afirma não existir um mapeamento claro e definitivo entre os códigos ICD-9 e ICD-10 pois, apesar de alguns códi-

[3] https://bitbucket.org/

[4] Git é uma plataforma de código aberto para gerenciamento de código-fonte e controle de versão utilizado largamente, com rapidez e eficiência, em projetos de qualquer tamanho. https://git-scm.com/

[5] https://www.medicaid.gov/medicaid-chip-program-information/by-topics/data-and-systems/icd-coding/icd-10-changes-from-icd-9.html

Figura 4.5: *Tela de Overview do BitBucket para o Projeto Ontofamily.*



gos possuírem correspondência 1-para-1, a grande maioria dos códigos possuem correspondência 1-para-muitos, muitos-para-muitos ou, até mesmo, nem possuem correspondência. Baseado nesse cenário e na influência que o mesmo iria causar à *Li-Fraumeni Ontology* futuramente (está previsto para 2018 o lançamento da ICD11[6]), decidimos abordar o lançamento de versões baseado em ontologias intermediárias que importam os conceitos anteriores e acrescentando/expandindo os novos conceitos, mantendo, assim, a compatibilidade entre novas e antigas versões das ontologias. A Figura 4.6 apresenta um esquema diagramático do modelo proposto para o controle de novas versões. Essa abordagem permite que os usuários da *Li-Fraumeni Ontology* possam utilizar quaisquer das versões publicadas, inclusive mais de uma simultaneamente, sem que ele precise reescrever código ou redefinir termos e conceitos.

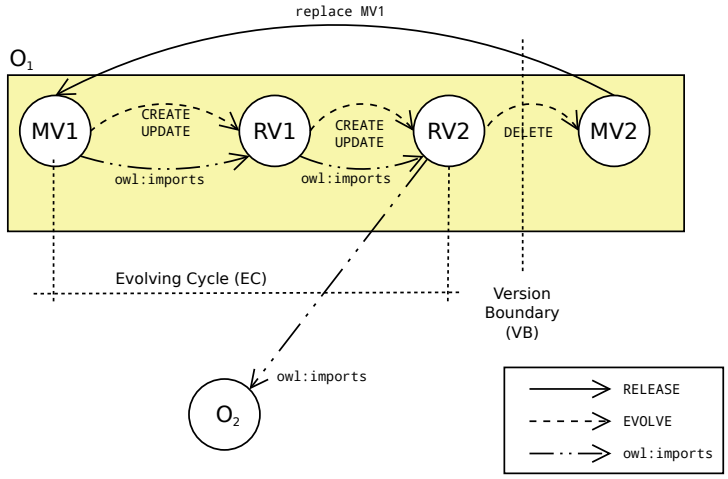
A metodologia conta com os seguintes conceitos básicos:

- *Minimal Version* (MV): É a versão inicial que será usada como base para um ciclo de evolução;
- *Release Version* (RV): São versões da MV criadas por meio de modificações `UPDATE` e `CREATE`
- *Evolving Cycle* (EC): É o período de evolução de uma ontologia compreendido entre uma MV e a última RV antes da VB;
- *Version Boundary* (VB): É o limite dado para um ciclo de evolução e que é disparado quando uma modificação `DELETE` é aplicada à MV daquele EC, obrigando a atualizar a antiga MV pela nova e incorporando todas as versões lançadas naquele EC até à modificação que gerou a VB.

Inicialmente, mapeamos 3 tipos de modificações que podem disparar o lançamento de uma nova versão da ontologia: **`CREATE`**, **`UPDATE`** e **`DELETE`**. Modificações `CREATE` apenas adicionam novos conceitos, regras, axiomas e propriedades sem que os mesmos entrem em contradição com regras já existentes, pois isso obrigaria o desenvolvedor a **excluir** o axioma anterior. Nesse caso, teríamos, além de uma modificação `CREATE`, uma outra modificação `DELETE`. Modificações `UPDATE` expandem conceitos já existentes, sem criar novos conceitos nem ir de encontro a outros já existentes. A modificação de `DELETE` refere-se à exclusão de um conceito, seja porque ele não exista mais, seja

[6] http://www.who.int/classifications/icd/revision/en/

Figura 4.6: *Proposta de metodologia para controle de versões em desenvolvimento colaborativo de ontologias.*

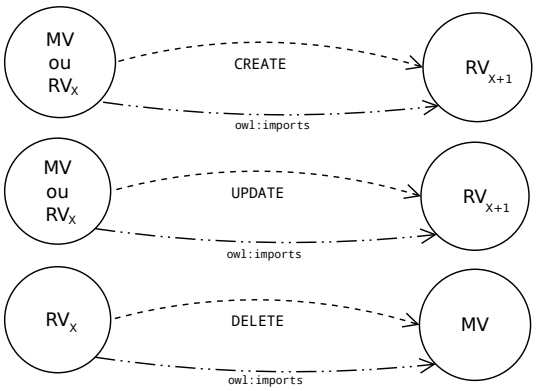


A MV1 representa a primeira Minimal Version da ontologia O_1. Observamos que, em razão de modificações CREATE ou UPDATE (ou as duas juntas), cria-se a versão RV_1, que também importa os conceitos de MV_1. Também em razão de modificações do tipo CREATE ou UPDATE (ou as duas juntas), uma segunda versão RV_2 é lançada. Enfim, RV_2 precisou de modificações do tipo DELETE e, em razão disso, estabelece-se um Version Boundary, todas as modificações de RV_1 e RV_2 e as modificações de exclusão são materializadas na MV_2, que substitui a MV_1, recomeçando o EC novamente. No esquema, é possível ver que uma ontologia O_2 está fazendo uso da segunda versão (RV_2) da ontologia O_1.

porque ele deva ser corrigido com axiomas diferentes daqueles já existentes. A Figura 4.7 mostra as 3 situações de modificação `CREATE`, `UPDATE` e `DELETE`.

Essas 3 modificações serão usadas como "gatilho" para duas operações distintas: **`EVOLVE`** e **`RELEASE`**. A operação `EVOLVE` cria uma nova RV baseada na RV anterior. Ela é disparada sempre que modificações `CREATE` e `UPDATE` forem acionadas. A operação `RELEASE` é mais complexa, pois ela materializa todas as RV do EC em uma nova MV, substituindo a anterior. Ela será acionada sempre que uma operação `DELETE` for necessária.

Figura 4.7: *Os três tipos de modificações em uma ontologia (`CREATE`, `UPDATE` e `DELETE`) e suas respectivas consequências.*



Alguns cuidados devem ser observados na aplicação da metodologia:

1. As modificações `CREATE` e `UPDATE` podem ser aplicadas separadamente ou simultaneamente na mesma RV;

2. A modificação `DELETE` deve ser aplicada separadamente das demais operações, pois esta será

responsável por criar um VB e gerar novas MV;

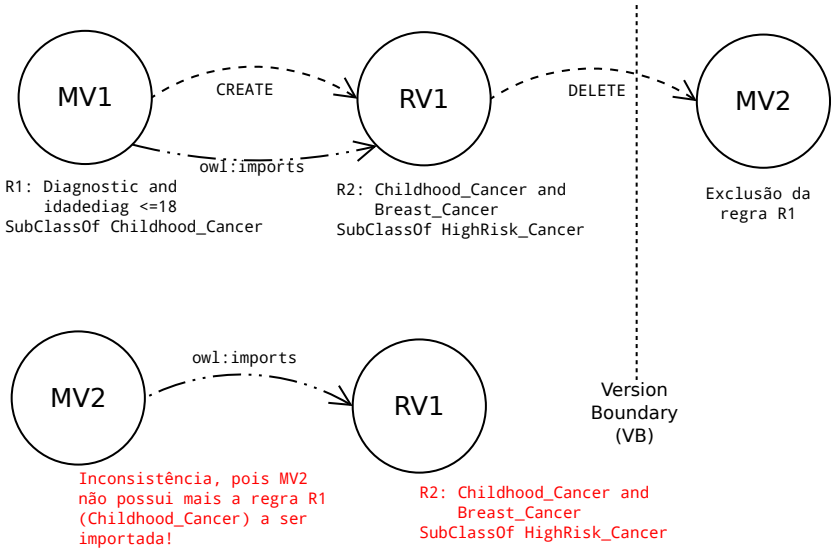
3. Sempre que uma nova MV for criada, todas as RV daquele EC devem ser materializadas na nova MV.

4. Uma VB marca o horizonte de eventos da nova versão, ou seja, o ponto de onde não é mais possível retroceder para versões anteriores. Por essa razão, modificações `DELETE` devem ser implementadas após um certo nível de discussão entre os especialistas de domínio.

5. Considere RV_x de uma ontologia $\mathcal{O}$ uma das suas *release versions*, em que x representa a versão em que $\mathcal{O}$ se encontra no momento (RV_2 indica que a ontologia $\mathcal{O}$ encontra-se na segunda *release version*). Assim, será verdade que toda RV_x importará os conceitos de RV_{x-1}. Com isso, garantimos que as RV mais recentes possuam todos os conceitos das RV anteriores e os inicialmente definidos pela MV.

Inicialmente, para as operações de `RELEASE`, havíamos pensado em manter as RV e materializar apenas a modificação de `DELETE` na nova MV. Entretanto, o modelo poderia facilmente apresentar problemas, pois as RVs criadas anteriormente referem-se a uma MV específica que, ao ser modificada (exclusão de um conceito, por exemplo), poderia gerar inconsistências com uma dada RV que importa, da antiga MV, o conceito que não mais existe. A Figura 4.8) ilustra uma situação de inconsistência resultante da não materialização das RVs na nova MV.

Figura 4.8: *Situação em que, após um RELEASE, substituiu-se a MV1 pela nova MV2, mantendo-se, entretanto, as mesmas RVs.*



Como essas RVs não foram materializadas na MV2, então haverá uma inconsistência devido à regra R1 não mais existir na MV1, mas RV1 continua a referenciá-la por meio da regra R2. Por essa razão que todas as RVs devem ser materializadas na nova MV.

Assim, quando um *Evolving Cycle* atingir a *Version Boundary*, uma sequência de ações devem ser tomadas:

1. Materializar todas as RV's e a MV anterior na nova MV;

2. Proceder com a operação de `DELETE` na nova MV ;

3. Fazer uma varredura e remover todos os axiomas e regras que referenciem a(s) regra(s) excluída(s) no passo anterior, fazendo a documentação da exclusão nos campos de anotação adequados (Notes ou Changes);

4. Substituir a MV antiga pela nova MV.

Algumas vantagens na abordagem colaborativa que propomos podem ser observadas. A primeira delas diz respeito à não obrigatoriedade no uso de ferramentas especiais para o uso dessa metodologia (ferramentas simples como Protegé podem ser utilizadas para criar as referências de importação). O controle das novas RV, MV e VB fica sob a responsabilidade dos desenvolvedores da ontologia proporcionando, assim, uma flexibilidade do processo de evolução das versões. Outra vantagem é a possibilidade dos usuários das ontologias poderem trocar as versões utilizadas (RVs) sem maior esforço, bastando apenas modificar a cláusula `owl:imports` para a versão que se deseja utilizar. Por exemplo: sejam duas ontologias $\mathcal{O}_1$ e $\mathcal{O}_2$ tal que $\mathcal{O}_1$ importa RV_3 da $\mathcal{O}_2$. Isso quer dizer que a ontologia $\mathcal{O}_1$ faz uso dos conceitos presentes na versão 3 da ontologia $\mathcal{O}_2$. Suponha que uma nova RV de $\mathcal{O}_2$ (RV_4) acaba de ser lançada e implementa um conjunto de axiomas que entram em conflito com alguns outros axiomas da $\mathcal{O}_1$. Nesse caso, é possível optar por continuar utilizando a RV_3 sem nenhum prejuízo para $\mathcal{O}_1$ até que ele possa ser revisado e, com isso, passe a utilizar a RV_4.

Por fim, propomos um conjunto de anotações que devem ser incorporadas às ontologias (RVs e MVs) a fim de permitir uma melhor documentação do processo de desenvolvimento. Sugerimos, também, o uso de uma ferramenta automatizada para a geração de documentação, como o OWLDoc[7]. Essa ferramenta é um *plugin* do editor de ontologias Protegé e fornece uma documentação no formato HTML estático, no mesmo estilo da documentação JavaDoc, gerada a partir da própria estrutura do arquivo OWL. Os campos de anotações sugeridos são:

- `ProposalNumber`: Número de controle (*surrogate*) que identifica essa proposta de versão. Ela pode estar associada a uma *issue* ou a um conjunto de *issues* no *issue tracker*.
- `RevisorsName`: Nomes dos revisores da ontologia.
- `Version`: Número que identifica a versão da ontologia. Pode ser usado algum esquema de numeração para versionamentos, como a *sequence-based identifiers*, *change significance*, etc.
- `ReleaseDate`: Data em que a versão foi lançada. No caso das MV, essa é a data em que ocorre o *Version Boundary*.
- `Changes`: Texto contendo uma descrição em linguagem natural de tudo o que mudou nesta versão em relação à última. Também é o conjunto de mudanças que motivou o lançamento da versão.
- `MinimalVersionOntologyAssociated`: Esse é o nome da ontologia MV associada às RV. Caso esta seja a própria ontologia MV, esta anotação ficará em branco.
- `Notes`: Quaisquer informações adicionais que se julgue necessária para a documentação da versão.

[7] http://protegewiki.stanford.edu/wiki/OWLDoc

Até o momento, identificamos apenas essas anotações como sendo relevantes para a construção da ontologia. Entretanto, nada impede que novos campos sejam identificados e incorporados às anotações das versões.

4.4 Conclusão

Apesar de já existirem diversas ferramentas que auxiliam o processo de desenvolvimento colaborativo, a área da Ontologia ainda carece de metodologias mais maduras que incorporem os benefícios da metodologia ágil com a capacidade de colaboração sem a necessidade de um conjunto de ferramentas especializadas para esse fim. As metodologias analisadas, ou abordam as tarefas de colaboração de maneira muito superficial, sem detalhar procedimentos (UPON) ou mesmo deixando à cargo de outras metodologias (RapidOWL) ou então não levam em consideração essa possibilidade, como a METHONTOLOGY e a metodologia 101. Isso, entretanto, não inviabiliza a aplicação dessas metodologias ou desmerece o esforço que elas apresentam na tentativa de apresentar um método estruturado de gestão de fontes de conhecimento.

Diante desse cenário, propomos uma metodologia que aliou o melhor de duas metodologias analisadas (METHONTOLOGY e RapidOWL) para a construção da *Li-Fraumeni Ontology*. Com o intuito de proporcionar um ambiente propício para o seu desenvolvimento, utilizamos o repositório BitBucket, que proporciona um conjunto de funcionalidades importantes ao desenvolvimento colaborativo de códigos e propusemos, também, metodologia para gerenciar o controle de versão das ontologias criadas. Essa metodologia se mostrou simples, de fácil uso e de rápida aplicação. A documentação das versões pode ser feita por meio de anotações nas próprias ontologias, facilitando futuras documentações geradas automaticamente por meio de ferramentas destinadas para esse fim, como a OWLDoc.

Julgamos, assim, que a metodologia cumpriu com o proposto, atingindo os objetivos esperados de maneira rápida, simples e com uma curva de aprendizado baixa, pois não exige nenhum conhecimento técnico além do que já é necessário no domínio das ontologias. Na próxima Seção, então, apresentaremos o resultado da aplicação da nossa metodologia: o processo de desenvolvimento colaborativo da *Li-Fraumeni Ontology*.

Capítulo 5

Li-Fraumeni Ontology

Neste capítulo, apresentaremos as três ontologias que foram desenvolvidas com o objetivo de representar o conhecimento necessário para a classificação de famílias que reúnem critérios clínicos da Síndrome de *Li-Fraumeni*, juntamente com suas etapas de desenvolvimento, os obstáculos encontrados e decisões de projeto. Na primeira parte, apresentaremos a ontologia de informações familiares, que chamamos de *Genealogy Ontology (GenOnto)* (Seção 5.1). Esta ontologia conterá a descrição das relações familiares, bem como axiomas e regras de inferência para a descoberta de novas relações familiares entre pacientes que não tenham sido explicitamente descritas no `TBox`. Em seguida, apresentaremos a ontologia para dados clínicos *Clinical Data Ontology (CDOnto)* (Seção 5.2), que contém a descrição de todas as informações/critérios clínicos que serão colhidos por meio da análise de exames e de consulta clínica com o paciente. Os critérios clínicos são informações sobre o paciente obtidas por meio de perguntas e respostas feitas a ele e/ou a outros membros de sua família, sem que haja necessidade de nenhum tipo de procedimento cirúrgico ou laboratorial. Por fim, apresentaremos a *Li-Fraumeni Ontology (LFOnto)* (Seção 5.3), que contém os axiomas e as regras necessárias para a inferência de pacientes que possuam a Síndrome de *Li-Fraumeni*.

Segundo [NM01], alguns dos motivos que impulsionam o desenvolvimento de uma ontologia são o compartilhamento de conhecimento e reuso de domínio. Por essa razão, optamos por modelar separadamente os três domínios de conhecimento (*Li-Fraumeni*, Dados Clínicos e Árvore Genealógica) em razão de algumas facilidades de projeto, dentre elas a modularização e a facilidade de posterior correções e manutenções nas ontologias, facilitando o reuso da base de conhecimentos. Essa forma de modelar o conhecimento por meio de três ontologias não sugere o uso de técnicas de integração de ontologias, mas apenas o reuso do conhecimento causado pela intersecção de domínios (ver Seção 2.2.7).

5.1 GenOnto - Genealogy Ontology

Segundo [Gui05], uma **ontologia de referência** deve ser construída com o único objetivo de fazer a melhor descrição possível do domínio tratado. Uma ontologia de referência, portanto, pode ser utilizada para modelar outras situações que estejam dentro do mesmo domínio abordado pela ontologia de referência, mas que façam uso de apenas alguns dos conceitos descritos por ela, quer seja por questões de restrição de domínio, quer seja por manutenção da decibilidade e da computabilidade.

No contexto da Síndrome de *Li-Fraumeni*, os critérios clínicos para classificação de um paci-

ente como portador dessa síndrome passam, obrigatoriamente, por uma análise do seu histórico familiar, ou seja, é necessário investigar as ocorrências de casos oncológicos de membros da família do paciente, sejam eles antepassados distantes ou não. Apesar de existirem diversos modelos de ontologias que representam árvores genealógicas, não encontramos nenhuma ontologia que pudesse ser diretamente utilizada em todas as situações que envolvem o uso desse tipo de domínio, sendo necessárias adaptações ao modelo que for escolhido (ver Seção 2.2.5). Alguns dos obstáculos que contribuem para essa dificuldade são:

a) As discrepâncias existentes nas definições de termos, nomes dos indivíduos e localidades podem dificultar o reconhecimento de nomes;

b) Diferenças culturais, como padrão de sobrenomes, relações de parentesco, etc. Em alguns países, como França, Espanha, Bélgica e Luxemburgo, relações incestuosas[1] entre indivíduos de mesma linhagem não são proibidas. Em alguns países como Suécia, Alemanha e Grécia, o incesto é considerado crime. Já a poligamia é uma prática permitida em alguns países da África e da Oceania e nos países do Oriente Médio. Nesses casos, uma ontologia que possua uma propriedade `hasPartner` não será modelada da mesma forma em cada um desses países e deverá sofrer adaptações;

c) Computabilidade dos modelos de árvores genealógicas existentes. Uma ontologia que modele muitos graus de parentesco entre indivíduos (relações de quinto grau, por exemplo) levando em consideração o gênero (diferença entre **tio** e **tia**, **pai** e **mãe**, por exemplo) poderá demorar um tempo muito grande para concluir a inferência de todas as relações entre todos os indivíduos (dependendo da quantidade de indivíduos e de axiomas no `ABox`, esse tempo poderá ser impraticável).

Além dos problemas citados acima, decidir trabalhar com Assertivas de Mundo Aberto (*OWA: Open-World Assumption*[2]) ou Assertivas de Mundo Fechado (*CWA: Closed-World Assumption*[3]) influencia diretamente na complexidade da ontologia. No modelo *OWA*, qualquer assertiva que não esteja explicitamente declarada ou não possa ser deduzida pelos motores de inferência não pode ser concluída nem como verdadeira nem como falsa; enquanto que no modelo *CWA*, apenas as assertivas explicitamente declaradas são consideradas verdadeiras. Segundo [ACM12], a maioria das ontologias que descrevem genealogias assumem o modelo *CWA* para que seja mantida a computabilidade e a decidibilidade. A GenOnto assume o modelo *CWA* pelas mesmas razões acima.

Conforme descrito anteriormente, desenvolver uma ontologia de informações familiares (genealogia) esbarra em diversos problemas. A heterogeneidade e a imprecisão das informações, o propósito destinado à essa ontologia e, principalmente, as diferenças culturais entre as sociedades são exemplos de barreiras explícitas que dificultam a construção de um modelo conceitual único de genealogias. Como exemplo, considere uma relação `a hasPartner b` que descreve o casamento de uma pessoa `a` com outra pessoa `b`. Essa relação poderia ser descrita como sendo uma relação **simétrica** (*symmetric*), **irreflexiva** (*irreflexive*) e **funcional** (*functional*). Relações **simétricas** são aquelas

[1] Segundo o dicionário Michaelis, incesto é definido como **"[...] união sexual entre parentes (consanguíneos ou afins), condenada pela lei, pela moral e pela religião."**

[2] Utilizaremos o termo em inglês por se tratar de um termo amplamente consagrado e conhecido na literatura e nos trabalhos científicos de sua área.

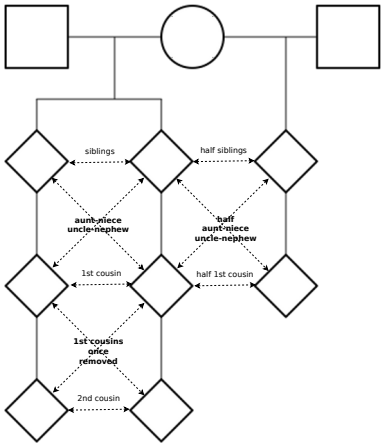
[3] Aqui, daremos preferência pelos termos originais em inglês pelo fato dos mesmos serem largamente usados na literatura especializada.

equivalentes à sua relação inversa, ou seja, `a` *`hasPartner`* `b` é equivalente a `b` *`hasPartner`* `a`. Relações **irreflexivas** são aquelas que podem relacionar um indivíduo a si próprio, como por exemplo `a` *`hasMother`* `a`. Por fim, propriedades **funcionais** são aquelas que relacionam um indivíduo qualquer a um e somente um outro indivíduo. Ou seja, se `a` *`hasMother`* `b`, então `a` *`hasMother`* `c` não será permitido. Assim, seja uma ontologia $\mathcal{O}$ que tenha sido modelada levando em consideração a propriedade *`hasPartner`* conforme descrito acima. Em uma sociedade poligâmica, esta propriedade causaria inconsistências a partir do momento em que existisse uma pessoa que possuísse mais de uma esposa ligada a ele por meio da propriedade *`hasPartner`* em decorrência desta ser do tipo **funcional**. A solução para esse problema exigiria um "relaxamento" do axioma que descreve *`hasPartner`* como funcional. Sem esse axioma, no entanto, a ontologia poderia acusar a ocorrência de inconsistências se aplicada para representar as relações familiares de uma sociedade monogâmica. Outro problema reside no propósito destinado a essa ontologia. Modelar relações familiares que não sejam necessariamente consanguíneas (irmãos adotivos, por exemplo) pode resultar em uma ontologia bem diferente de outra cujas relações tenham o propósito de modelar apenas relações consanguíneas entre indivíduos (representar informações sobre herança genética, por exemplo). Implicitamente, podemos citar a computabilidade e a decidibilidade como elementos decisivos que podem comprometer a usabilidade da ontologia. Árvores genealógicas com muitos graus de parentesco podem demorar um tempo impraticável para inferir todas as relações entre os indivíduos. É por essa razão que a maioria das ontologias que descrevem as relações familiares são do tipo *CWA*. Entretanto, mais uma vez, dependendo do propósito, uma ontologia que assuma *CWA* pode não ser adequada a uma determinada finalidade (se o propósito for descobrir relações familiares não explícitas no modelo conceitual, uma ontologia *CWA* poderá limitar o alcance do motor de inferência deixando de fora resultados que poderiam ser considerados válidos).

Durante a construção das classes e das relações da *GenOnto*, foram levadas em consideração apenas as relações consanguíneas entre os indivíduos. Apesar de cada sociedade poder adotar um critério próprio para a definição dos graus de parentesco, o padrão adotado neste trabalho segue a recomendação da literatura genética, aonde o grau de parentesco está diretamente relacionado ao número de genes compartilhados entre os indivíduos [BMB+02]. Uma tabela representativa dos graus de parentesco entre indivíduos da mesma família pode ser vista em 5.1. Relações homoafetivas também não foram levadas em consideração neste trabalho em razão destas serem incapazes (até o momento) de gerar descendentes com material genético herdado exclusivamente dos cônjuges. Entretanto, é sabido que os fatores ambientais também podem influenciar diretamente a expressão gênica de uma determinada doença em um indivíduo. Segundo [HMC+13], a suscetibilidade de um indivíduo a doenças complexas, incluindo o câncer, é multifatorial, envolvendo múltiplos fatores de risco genéticos e ambientais.

Como o propósito do modelo conceitual da *GenOnto* era representar apenas as relações capazes de proporcionar a transmissão de carga genética entre indivíduos, as classes e propriedades que não serviam a esse propósito não foram descritas. Também não foram consideradas especificidades de sexo, como a diferença entre **irmão** e **irmã**, ou **pai** e **mãe** nem as propriedades que dessas classes se depreendem por não serem relevantes para o motor de inferência nem para o conjunto de critérios da Síndrome de *Li-Fraumeni*. Em contrapartida, o atributo `gender` foi modelado na ontologia como um dado específico de cada paciente oriundo da base de dados do *A.C. Camargo Cancer Center*. Assim, futuramente, pode-se dar um uso adequado a esses dados durante o processo de descoberta

Tabela 5.1: *Grau de parentesco sob o aspecto da consaguinidade. Adaptado de [CGC].*

Relationship	***Example Relationships***	***Shared Genes***	***Pedigree***
First Degree	*Parent-Child* *Siblings*	1/2	
Second Degree	*Half siblings* *Uncle-niece* *Aunt-nephew*	1/4	
Third Degree	*First cousins* *Half uncle-niece* *Half aunt-nephew*	1/8	
Fourth Degree	*First cousins once removed* *Half first cousins*	1/16	
Fifth Degree	*Second cousins*	1/32	

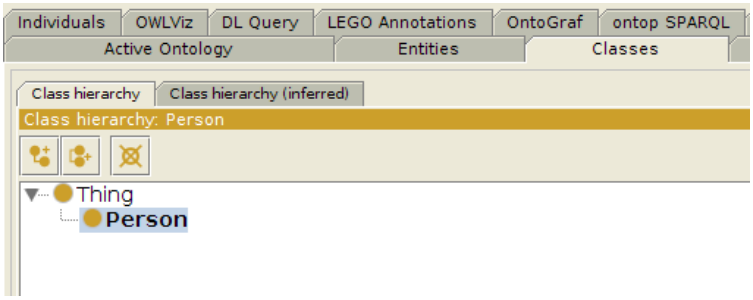
de conhecimento ou utilizá-lo para acrescer a ontologia com novos axiomas e regras.

O ponto de partida para a construção da *GenOnto* foi a definição de apenas uma classe primitiva chamada `Person` (Figura 5.1) aonde seriam classificados todos os pacientes e seus familiares. A opção por não modelar outras classes como `Siblings` ou `Parents` fundamenta-se no fato de que, para o problema da Síndrome de *Li-Fraumeni*, elas não se mostram essenciais para a classificação de pacientes, além de aumentar o tempo de classificação dos indivíduos pelo motor de inferência. Assim, a escolha das classes, propriedades e atributos foram norteadas pelas questões de competência. As questões de competência servem para direcionar a construção de novas ontologias, bem como "certificar", futuramente, se a ontologia construída consegue responder a todos os questionamentos previamente elaborados acerca do domínio em questão [NM01]. As questões de competência definidas foram:

- Quem são os parentes de primeiro (segundo ou terceiro) grau de um determinado indivíduo?
- Quem são os parentes vivos de um determinado indivíduo?
- Quem são os parentes de primeiro ou segundo graus de um determinado indivíduo?
- Liste dois ou mais parentes distintos, um de primeiro grau e outro de segundo grau, de um determinado individuo.

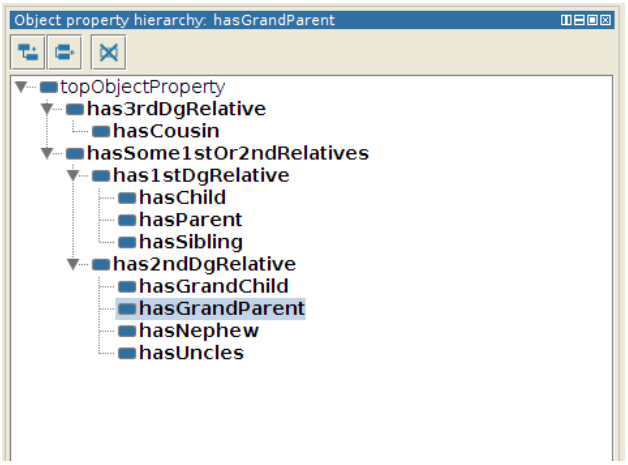
Para a *GenOnto*, mais do que descrever apenas as relações familiares como `hasPartner`, `hasChild` ou `hasSibling`, é importante classificar os graus de parentesco (grau de consanguinidade) existentes entre os indivíduos de uma mesma família, como por exemplo, se um indivíduo `a` é parente de primeiro grau de um outro indivíduo `b`, ou simplesmente classificar todos os parentes de segundo grau de um indivíduo `c`. Essa exigência do modelo conceitual se justifica nos diferentes critérios clínicos para classificação de pacientes portadores da Síndrome de *Li-Fraumeni* (*Classic*, Chompret, Eeles e Birch).

Figura 5.1: *Classe* **Person** *no Protegé.*



Inicialmente, foi estabelecida uma hierarquia para as propriedades da *GenOnto* em que duas propriedades foram definidas como subclasses diretas de `topObjectProperty`: são elas `has3rdDgRelative` e `hasSome1stOr2ndRelatives`. Essas propriedades não são atribuídas diretamente aos indivíduos. A propriedade `has3rdDgRelative` possui apenas uma subpropriedade: `hasCousin`. Já a propriedade `hasSome1stOr2ndRelatives` foi criada com o objetivo de permitir que haja uma disjunção entre os indivíduos classificados como `has1stDgRelative` ou `has2ndDgRelative` pois, em um dos critérios para a classificação da Síndrome de *Li-Fraumeni Classic*, faz-se necessário buscar os indivíduos que são classificados tanto como parentes de primeiro grau como parentes de segundo grau do probando. Conforme veremos mais adiante (Seção 5.3), a modelagem desse critério foi feita por meio de regras seguindo o padrão SWRL[4] e a mesma não permite disjunções (cláusulas `OR`) no corpo da fórmula (Cláusulas de Horn[5]). A propriedade `has1stDgRelative` possui três sub-propriedades que definem as relações de consanguinidade de primeira ordem: `hasPartner`, `hasChild` e `hasSibling`. A propriedade `hasSibling` é a única que não foi definida por meio de expressões de classe. Algumas dificuldades em modelar esse tipo de relação por meio de expressões de classe são apresentadas em [SSMJ13], como, por exemplo, atribuir a característica **irreflexiva** à propriedade `hasSibling`, causando uma inconsistência na ontologia [6]. Aqui, também foi usada uma regra escrita em SWRL a fim de contornar a restrição de decidibilidade da OWL2 DL (Algoritmo 5.1). A Tabela 5.2 contém uma descrição resumida das propriedades definidas na *GenOnto* conforme a Figura 5.2.

Figura 5.2: *Propriedades da GenOnto no Protegé.*



[4] https://www.w3.org/Submission/SWRL/
[5] Cláusula de Horn: cláusula em que, pelo menos, um dos seus literais (átomos) é não-negativo [Hor51]
[6] http://www.w3.org/TR/owl2-syntax/#The_Restrictions_on_the_Axiom_Closure

Algoritmo 5.1: *Listagem de regra SRWL para definir propriedade* ***hasSibling***

```
1    hasChild(?parent, ?otherChild), hasParent(?child, ?parent),
2      DifferentFrom (?child, ?otherChild) -> hasSibling(?child, ?otherChild)
```

Propriedade	Características	Descrição
has3rdDgRelative	Symmetric Range: Person Domain: Person	$has3rdDgRelative \sqsupseteq hasCousin$
hasCousin	Symmetric Range: Person Domain: Person	PropertyChain: *hasUncles o hasChild*
hasSome1stOr2ndRelatives	Range: Person Domain: Person	$hasSome1stOr2ndRelatives \sqsupseteq (has1stDgRelative \bigsqcup has2ndDgRelative)$
has1stDgRelative	Symmetric Range: Person Domain: Person	$has1stDgRelative \sqsupseteq (hasChild \bigsqcup hasParent \bigsqcup hasSibling)$
hasChild	InverseOf: *hasParent* Range: Person Domain: Person	$hasChild \sqsubseteq has1stDgRelative$
hasParent	InverseOf: *hasChild* Range: Person Domain: Person	$hasParent \sqsubseteq has1stDgRelative$
hasSibling	Symmetric Range: Person Domain: Person	$X.hasParent(Y.hasChild(Z)) \vdash X.hasSibling(Z)$
has2ndDgRelative	Symmetric Range: Person Domain: Person	$has2ndDgRelative \sqsupseteq (hasGrandChild \bigsqcup hasGrandParent \bigsqcup hasNephew \bigsqcup hasUncles)$
hasGrandParent	InverseOf: *hasGrandChild* Range: Person Domain: Person	PropertyChain: *hasParent o hasParent*
hasGrandChild	InverseOf: *hasGrandParent* Range: Person Domain: Person	
hasNephew	InverseOf: *hasUncles* Range: Person Domain: Person	PropertyChain: *hasSibling o hasChild*
hasUncles	InverseOf: *hasNephew* Range: Person Domain: Person	

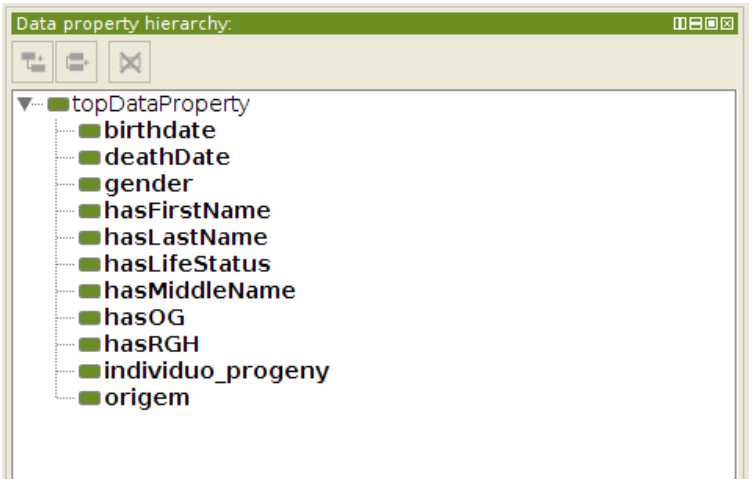
Tabela 5.2: *Hierarquia de propriedades da* GenOnto.

Por fim, foram criados alguns *Data Properties*[7] com o objetivo de enriquecer o modelo de histórico familiar e permitir que, futuramente, possam ser utilizadas na criação de novas regras (Figura 5.3).

A seguir, apresentaremos a ontologia *CDOnto*, responsável por representar todos os conceitos relacionados a dados clínicos de pacientes portadores da Síndrome de *Li-Fraumeni*. Vale ressaltar que a *CDOnto* importa a ontologia *GenOnto* para a construção de alguns axiomas.

[7] Aqui, daremos preferência pelos termos originais em inglês pelo fato dos mesmos serem largamente usados na literatura especializada.

Figura 5.3: *Lista com os Data Properties da GenOnto*



5.2 CDOnto - Clinical Data Ontology

Para a modelagem da *CDOnto*, foram usadas duas outras ontologias de referência como ponto de partida: ICD-10[8] e a ICD-O[9]. Essas ontologias descrevem uma estrutura de conceitos e metadados relacionados à Classificação Internacional de Doenças e Problemas Relacionados à Saúde (CID) versão 10, doravante chamado CID-10, e à Classificação Internacional de Doenças para Oncologia, doravante chamada CID-O. Veremos a seguir qual a contribuição dessas duas ontologias para a *CDOnto* e quais outros critérios clínicos foram usados na sua construção.

Inicialmente, a estrutura hierárquica de conceitos referentes a dados clínicos dos pacientes foram modelados juntamente com a *Li-Fraumeni Ontology*. Ou seja, junto à hierarquia de conceitos relacionadas à Síndrome de *Li-Fraumeni*, encontravam-se os conceitos modelados da CID-10. Foram deixados de fora os capítulos que não estavam relacionados a neoplasias. Observou-se, posteriormente, que os conceitos dos dados clínicos não pertenciam ao domínio da *Li-Fraumeni Ontology*, mas sim, a domínios de conhecimento diferentes. Segundo [UG96], o desenvolvimento de ontologias deve ser motivado por situações-problema que surgem ao longo do desenvolvimento de aplicações e de softwares. No contexto da Síndrome de *Li-Fraumeni*, as situações-problema que envolviam os dados clínicos motivaram a criação de *competency questions* que sugeriam um novo domínio do conhecimento levando, naturalmente, à criação de uma nova ontologia.

Nesse cenário, destacamos a importância de construir uma ontologia própria para o domínio de dados clínicos que pudesse responder às seguintes *competency questions*:

- Quais os pacientes que possuem ou já possuíram algum tipo de neoplasia (por exemplo, Sarcoma)?
- Quais os tipos de neoplasia que determinado paciente possui ou já possuiu?
- Qual a idade do paciente no momento de um dado diagnóstico (idade do paciente no primeiro diagnóstico)?

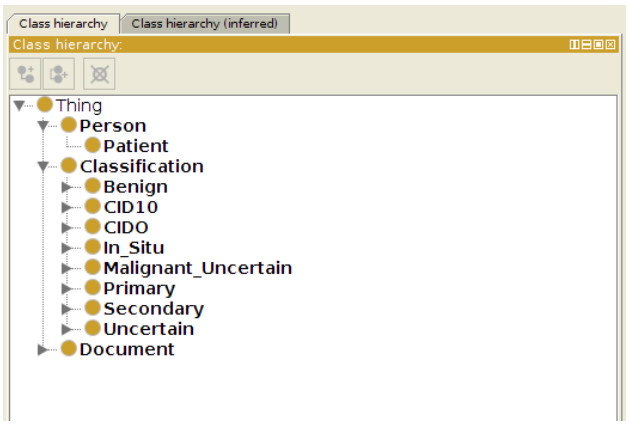
Entretanto, ao invés de construir uma ontologia completa desde o início, foi usada uma ontologia de referência para a construção da hierarquia de conceitos das neoplasias a partir do SNOMED-

[8] http://www.nlm.nih.gov/research/umls/Snomed/us_edition.html
[9] Disponível em http://seer.cancer.gov/icd-o-3

CT[10]. A SNOMED-CT é um conjunto de nomenclaturas médicas organizadas sistematicamente com o objetivo de prover códigos, nomes de termos, sinônimos e definições médicos usados em laudos e relatórios clínicos (ver Seção 2.2.6).

Nesse cenário, apesar da estrutura da *CDOnto*, até o momento, conseguir responder de forma correta às *competency questions* estabelecidas, foi constatado que algumas situações reais presentes nos bancos de dados do *A.C. Camargo Cancer Center* ficaram de fora do conjunto de respostas a essas questões, apesar de também fazerem parte dela. Um exemplo disso é quando um paciente é portador de algum tipo de neoplasia e seu diagnóstico está codificado usando a nomenclatura CID-O, e não a CID-10. Essa situação é bem mais comum do que os diagnósticos codificados apenas com CID-10. Dessa forma, a estrutura da *CDOnto* sofreu alterações na sua hierarquia de conceitos, que passou a conter também a estrutura de classes da CID-O (Figura 5.4).

Figura 5.4: CDOnto *após a inclusão do CID-O.*



Uma das classes modeladas na *CDOnto* foi a `Cancer_Diagnostic`, subclasse da classe `Document` (esta descreve todo e qualquer documento que pertence a um paciente, como um diagnóstico, o resultado de um teste genético, ou laudo, etc), que descreve os tipos de diagnósticos de câncer envolvidos na Síndrome de *Li-Fraumeni*. Entretanto, dependendo do tipo de doença e dos critérios clínicos envolvidos, essa classe pode ser estendida a fim de englobar mais tipos de diagnósticos de câncer ou de outros tipos de diagnósticos. A classe `Cancer_Diagnostic` é descrita por meio do seguinte axioma:

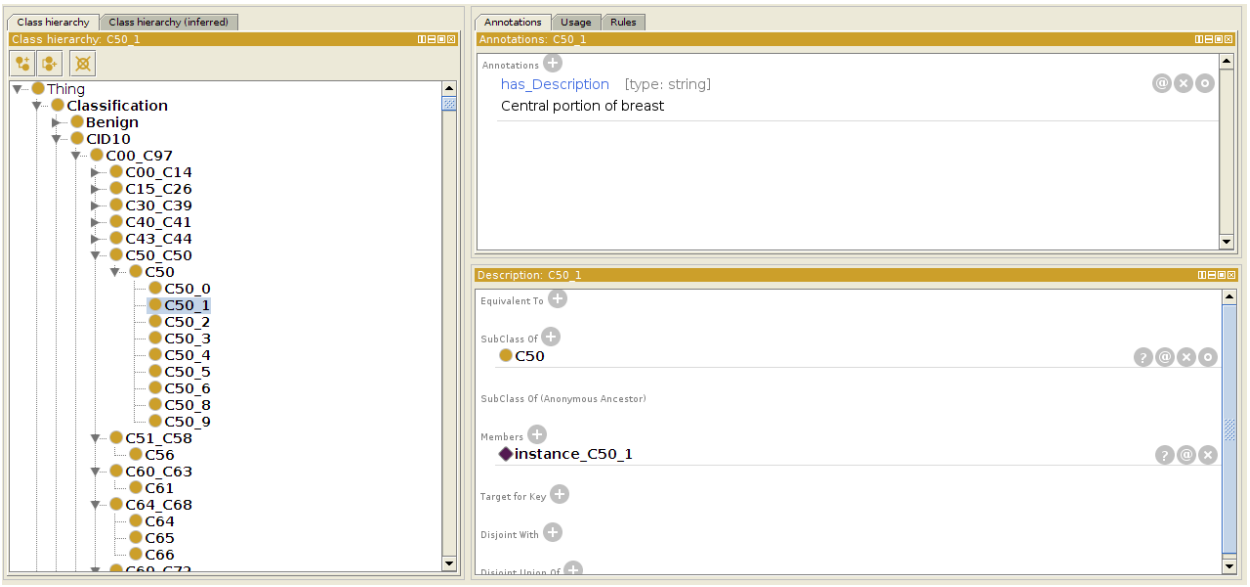
$$Document(d) \wedge d.hasDiagnosticCode(c) \wedge (ICD10(c) \vee ICDO(c)) \vDash Cancer\_Diagnostic(d)$$

Ainda em relação à classe `Cancer_Diagnostic`, esta é definida como disjunta da superclasse `Classification`, que é outro conceito descrito na *CDOnto*. Esta última é composta pelas subclasses `ICD10` e `ICDO` (respectivamente representando CID-10 e CID-O), que descrevem dois diferentes sistemas de codificação de doenças utilizados no cenário médico. Cada uma das subclasses que descrevem um código CID-10 ou CID-O está representada por meio de um indivíduo (objeto) do código em questão. Esse indivíduo é relacionado às instâncias da classe `Document` por meio da propriedade `hasDiagnosticCode`. Esta última classe conceitua todo e qualquer documento atribuído a um paciente. É possível que alguns documentos não sejam classificados posteriormente

[10]Baixado do *website* http://www.nlm.nih.gov/research/umls/Snomed/us_edition.html sob a licença de usuário NLM-0341109653

como `Cancer_Diagnostic`. A Figura 5.5 mostra um exemplo da hierarquia de classes para o código $C50.1$, que representa uma neoplasia maligna na porção central da mama.

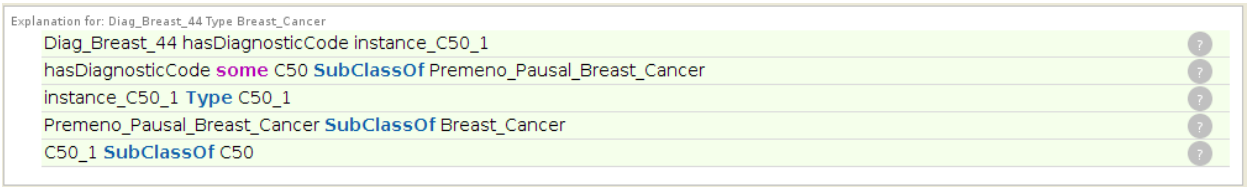
Figura 5.5: *Hierarquia de classes para C50_1*



O axioma 5.1 apresenta como é utilizada a propriedade `hasDiagnosticCode`, baseada na hierarquia de classes da `C50_1` (Figura 5.5), e como ela infere a classificação de um diagnóstico em uma das subclasses de `Cancer_Diagnostic`. No exemplo, um diagnóstico chamado ***Diag_Breast_44*** representa um documento diagnóstico de câncer de um paciente qualquer que recebeu como laudo o código $C50.1$. O resultado é que esse diagnóstico será classificado automaticamente como `Breast_Cancer`, conforme mostra a Figura 5.6, quando o motor de inferência for acionado.

$$Document(d) \wedge d.hasDiagnosticCode(c) \wedge C50(c) \vDash Breast\_Cancer(d) \tag{5.1}$$

Figura 5.6: *Resultado do uso da ferramenta Explain, no Pellet: O diagnóstico Diag_Breast_44 recebeu como laudo o código C50.1 e, após ligar o motor de inferência, foi classificado como* `Breast_Cancer`.

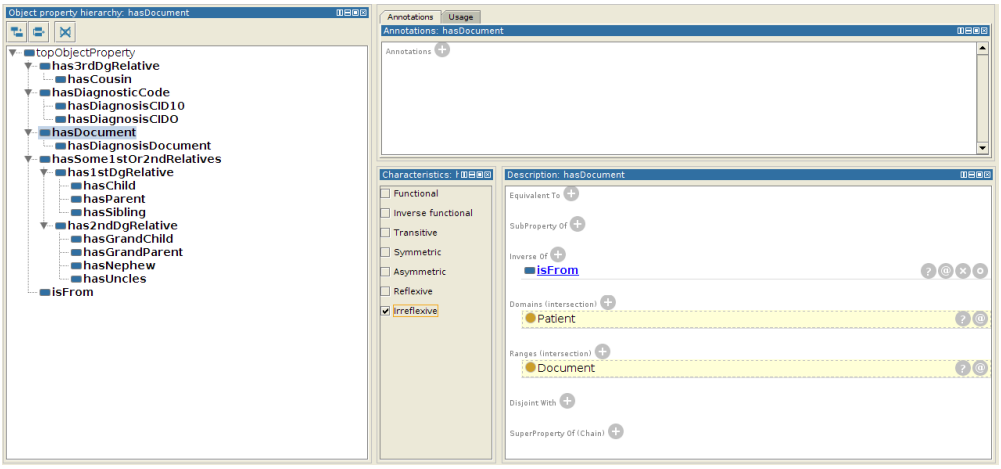


Observa-se até o momento que apenas a relação entre códigos de doenças e diagnósticos foi apresentada através da propriedade `hasDiagnosticCode`. É necessário estabelecer um "vínculo" entre os diagnósticos e os pacientes com seus familiares para que o motor de inferência possa classificar corretamente os pacientes como portadores ou não da Síndrome de *Li-Fraumeni*. Entretanto, a classe `Person` não faz parte do domínio da *CDOnto*, havendo a necessidade de que a *GenOnto* seja importada para o interior daquela.

Para que a relação entre diagnósticos e pacientes fosse estabelecida, foi modelada uma propriedade `hasDocument`. Ela estabelece uma relação irreflexiva e assimétrica entre indivíduos da classe

`Person` e indivíduos da classe `Document` (Figura 5.7). Uma relação inversa de *`hasDocument`*, chamada *`isFrom`*, também foi modelada para representar a que pessoa determinado diagnóstico pertence.

Figura 5.7: *A relação `hasDocument` e suas características.*



Em uma situação real, um paciente pode ter mais de um diagnóstico ao longo do tempo, apesar de propriedades, em uma ontologia, sempre relacionarem um indivíduo a outro indivíduo através de uma relação binária[11]. Esse fato limita a existência de um axioma do tipo

$$Paciente(p) \ \wedge \ p.hasDocument(d_1, d_2, d_3, \ldots, d_n) \tag{5.2}$$

Além disso, seria possível modelar a relação *`hasDocument`* de diversas formas diferentes:

hasDocument(“João”, *Diag*1)
hasDocument(“João”, *Diag*1, 10/01/1999)
hasDocument(“João”, *Diag*1, 10/01/1999, “Hospital das Clínicas”)
hasDocument(“João”, *Diag*1, 10/01/1999, “Hospital das Clínicas”, “Dr. Camanducaia Melo”)
⋮

A indefinição na quantidade de argumentos que a propriedade *`hasDocument`* possui (aridade) foi um dos motivos principais para o uso de reificação na modelagem dessa classe. A reificação pode ser vista como uma técnica em que conceitos abstratos são instanciados por meio de indivíduos a fim de que possam ser manipulados e quantificados posteriormente. Algumas das vantagens no uso de reificação está na flexibilidade de uso e na mudança de granularidade das propriedades, e na modelagem de propriedades mais complexas [BL04].

Dessa forma, uma abstração, como um diagnóstico (um documento que possui como laudo um código CID-10 ou CID-O), passa a ser “personificado” por meio de um indivíduo (por exemplo ***Diag_Breast_44***), podendo ser relacionado a outro indivíduo da classe `Person` através da relação *`hasDocument`* (Figura 5.8). Isso nos permite situar um diagnóstico no tempo (data do diagnóstico), recuperá-lo através de consultas e quantificá-lo (quantos diagnósticos uma pessoa pos-

[11] http://www.w3.org/TR/owl2-syntax/#Object_Properties

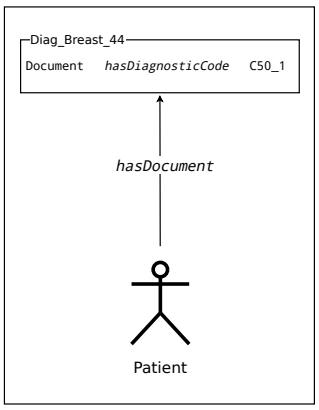


Figura 5.8: *Reificação da relação* `Document hasDiagnosticCode C50_1`.

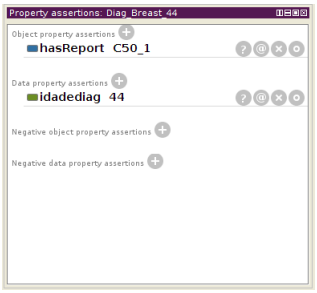
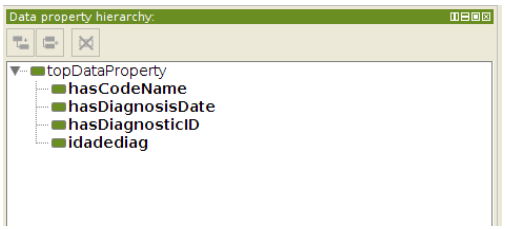


Figura 5.9: *O atributo* `idadediag`*, que descreve a idade do paciente na época em que o diagnóstico foi emitido, só pode ser modelado posteriormente porque a relação* `hasDocument` *foi reificada.*

sui), além de poder adicionar, posteriormente, mais informações que não foram previstas durante o processo inicial de modelagem da ontologia (Figura 5.9).

Alguns *Data Properties* (atributos) foram adicionados posteriormente em razão do processo de importação dos dados das bases de dados do *A.C. Camargo Cancer Center* terem permitido o completo mapeamento dos atributos dessas bases e também da escolha daqueles que eram essenciais para o processo de inferência. São eles `hasDiagnosisDate`, `hasDiagnosisID` e `idadediag` (Figura 5.10).

Figura 5.10: *Taxonomia de dados da* ***CDOnto****.*



Por fim, o conjunto de regras que classificam os diagnósticos codificados segundo a taxonomia da CID-10 e da CID-O foram definidos por meio de axiomas de classe[12] usando a propriedade `hasDiagnosticCode`. Como exemplo, apresentamos as expressões de classe 5.2, cujos axiomas estabelecem a relação dos diagnósticos com o Câncer de Próstata e com o Linfoma[13].

Algoritmo 5.2: *Listagem de regra SRWL para definir propriedade* `hasSibling`

```
1  hasDiagnosticCode some (C61 or T_C61) SubClassOf Prostate_Cancer
2  hasDiagnosticCode some (9823_3 or 9827_3 or 9835_3 or C81_C96) SubClassOf
       Lymphoma
```

A seguir será apresentada a ontologia ***LFOnto***, que reutiliza os conceitos modelados em ***CDOnto*** e ***GenOnto*** para construir um conjunto de conceitos da Síndrome de *Li-Fraumeni*, e permite que o motor de inferência possa classificar pacientes como portadores da síndrome, conforme os critérios de classificação vistos na Seção 3.2.

[12] Axiomas de classe são axiomas que permitem a construção de classes sem que seja dado nenhum nome para ela. Fonte: https://www.w3.org/TR/owl-ref/#ClassAxioms

[13] Não apresentaremos todos os axiomas aqui nesta Seção em razão da sua grande quantidade, o que tornaria a leitura monótona e cansativa.

5.3 LFOnto - Li Fraumeni Ontology

A ***LFOnto*** foi modelada usando os conceitos existentes da Síndrome de *Li-Fraumeni* juntamente com as regras que definem os quatro critérios clínicos de diagnóstico da síndrome: *Classic* , Chompret, Eeles e Birch (Capítulo 3.2). Esses critérios clínicos, apesar de bem definidos e largamente usados no auxílio ao diagnóstico de pacientes portadores da Síndrome de *Li-Fraumeni*, não possuem ontologias modeladas que descrevem seus conceitos e regras para classificação da síndrome. Sobre ontologias que definem o domínio da Síndrome de *Li-Fraumeni*, existem algumas que expressam apenas os conceitos médicos sobre a síndrome, se comportando como um *Thesaurus*. Para alcançarmos essa conclusão, foi realizada uma pesquisa nos repositórios de ontologias médicas mais utilizados: TONES, Gene Ontology, BioPortal, Ontobee, NCI, OLS e Swoogle. Os termos de busca utilizados nessa pesquisa foram `Li-Fraumeni, LFL, LFS, Chompret, Birch e Classic Li-Fraumeni`. Alguns desses repositórios retornaram resultados positivos quanto a existência de ontologias do domínio da *Li-Fraumeni*. Entretanto, em nenhuma dessas ontologias foram encontradas referências ou citações aos quatro conjuntos de critérios clínicos de diagnóstico da síndrome. Em todas as ontologias encontradas, a hierarquia de classes modelada era muito complexa, com muitos conceitos e hierarquia complexa[14] (Figura 5.11), o que faz aumentar o tempo de classificação de pacientes pelo motor de inferência. A Tabela 5.3 apresenta uma síntese com a tabulação dos resultados dessa pesquisa. Na primeira coluna são apresentados os repositórios de ontologias pesquisados. A coluna **Domínio Li-Fraumeni** informa se foi encontrada alguma ontologia com o domínio da *Li-Fraumeni*. A mesma interpretação cabe para a coluna **Domínio Critérios Clínicos**, em que foram levados em conta pelo menos um dos quatro critérios (*Classic*, Chompret, Birch e Eeles).

Repositório Pesquisado	Domínio *Li-Fraumeni*	Domínio Critérios Clínicos
TONES	Não	-
Gene Ontology	Não	-
BioPortal	Sim	Não
Ontobee	Sim	Não
NCI	Sim	Não
OLS	Sim	Não
Swoogle	Sim	Não

Tabela 5.3: *Repositórios de ontologias em que foram pesquisadas ontologias do domínio da Síndrome de Li-Fraumeni e dos 4 critérios clínicos.*

Assim, como o objetivo da ***LFOnto*** é prover axiomas para a classificação automática de pacientes em um ou mais critérios clínicos da Síndrome de *Li-Fraumeni*, o uso das ontologias encontradas não se mostrou útil pelos seguintes motivos:

1. Ontologias encontradas possuem muitos conceitos, ocasionando um alto custo de tempo no processo de inferência;

2. Ontologias encontradas são classificadas como OWL *Full*.

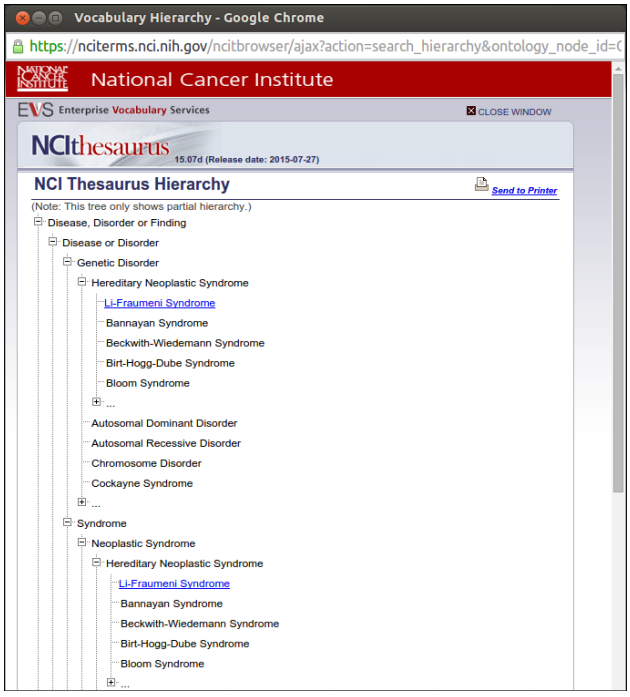
Diante desse fato, mostrou-se necessária a modelagem de uma ontologia de aplicação que pudesse ser usada como base de conhecimento para algum mecanismo de inferência e, ao mesmo tempo, apresentasse custo de inferência aceitáveis enquanto ferramenta de descoberta de conhecimento.

[14] A ontologia disponível no repositório NCI é classificada como *OWL Full* e possui mais de 100 mil classes.

A modelagem da ***LFOnto*** foi norteada pelas seguintes *competency questions*:

- Quais pacientes são classificados como portadores da Síndrome de *Li-Fraumeni*;
- Um determinado paciente é portador da Síndrome de *Li-Fraumeni*?
- Um paciente, quando classificado como portador da Síndrome de *Li-Fraumeni*, atende às regras de qual(is) dos quatro critérios de classificação: Chompret, *Classic*, Birch e/ou Eeles?
- Quais os atributos e critérios clínicos (diagnósticos) de um paciente classificado como portador da Síndrome de *Li-Fraumeni*?
- Quais os familiares de um paciente classificado como portador da Síndrome de *Li-Fraumeni* e seus respectivos graus de parentesco?
- Quais outros familiares do probando podem ser classificados como portador da Síndrome de *Li-Fraumeni*?

Figura 5.11: *Hierarquia de classes para o domínio da Síndrome de Li-Fraumeni, segundo o NCI, possui mais de 100 mil conceitos. Disponível em https://nciterms.nci.nih.gov/ncitbrowser/ConceptReport.jsp?dictionary=NCI_Thesaurus&version=15.07d&code=C98781*

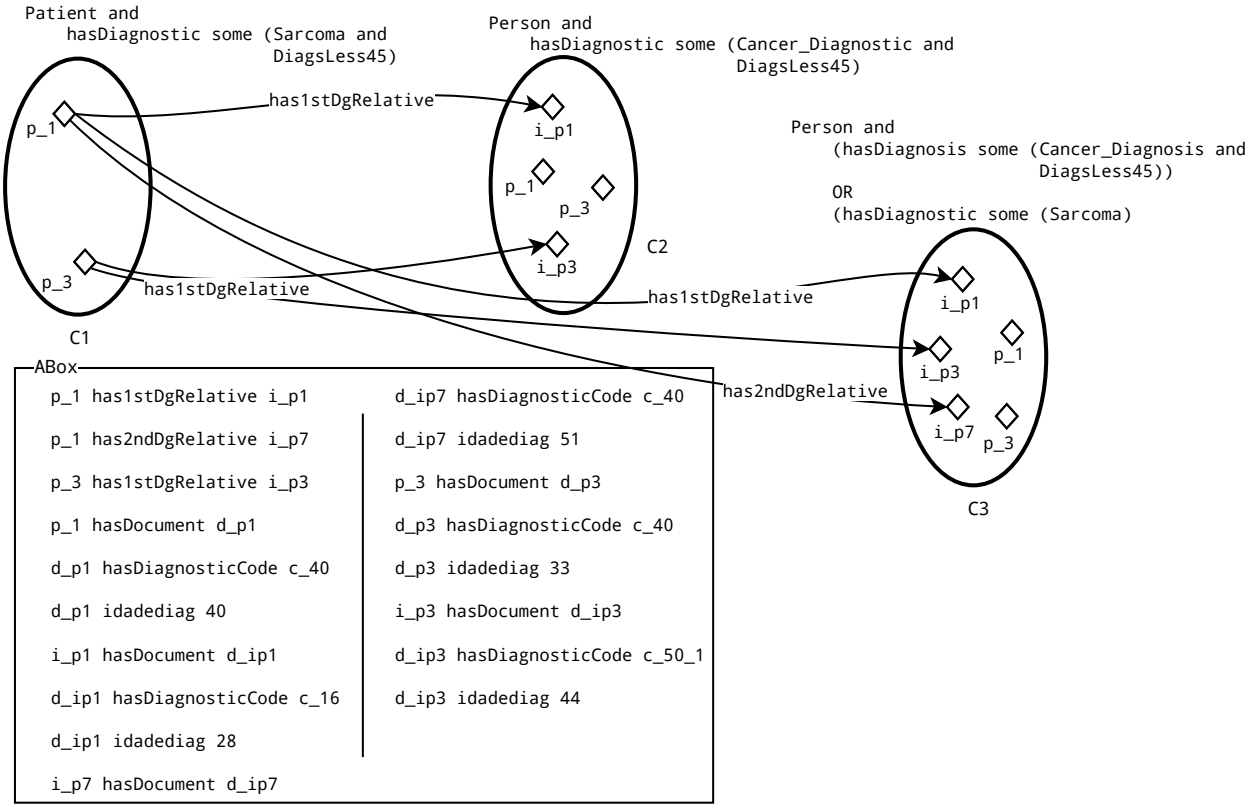


A modelagem dos axiomas para a classificação automática dos pacientes como membros de uma das classes ***Classic***, **Chompret**, **Eeles** e/ou **Birch** só foi possível através de regras SWRL, pois o uso de expressões de classe, conforme descreve a W3C [W3Cc], manipula apenas conjuntos de indivíduos que satisfazem as mesmas propriedades estabelecidas por ela. Por exemplo:

Seja a classe `Patient` o conjunto de todos os pacientes considerados probando da Síndrome de *Li-Fraumeni*. Cada um desses pacientes possui um conjunto de familiares pertencentes à classe `Person` e que, por sua vez, também podem ser considerados membros de `Patient`, caso sejam probandos. A relação entre os pacientes e seus familiares é definida por alguns axiomas pertencentes à GenOnto, dentre eles `has1stDgRelative`, `has2ndDgRelative` e `has1stOr2ndDgRelative`.

Um paciente **p_1** se relaciona com um familiar **i_p1** considerado parente de primeiro grau através do axioma **p_1** *has1stDgRelative* **i_p1**. Cada um dos probandos possui um ou mais diagnósticos pertencentes à classe `Document`. Um paciente **p_1** se relaciona com um diagnóstico **d_p1** através da relação **p_1** *hasDiagnostic* **d_p1**. Um esboço do `ABox` dessa ontologia pode ser observado na Figura 5.12. Em seguida, serão criadas três expressões de classe, uma para cada critério da Síndrome de *Li-Fraumeni* Clássica, e uma quarta expressão que irá classificar os indivíduos que atendem a todos os critérios da Síndrome de *Li-Fraumeni* Clássica. O intuito é mostrar que não será possível o uso somente de expressões de classe para a classificação de indivíduos como Síndrome de *Li-Fraumeni Classic*.

Figura 5.12: *Classes `C1`, `C2`, `C3` e os `ABox` e `TBox` do exemplo.*



Seja, então, uma classe chamada `C1` em que todos os pacientes que atendem ao critério 1 da Síndrome de *Li-Fraumeni* Clássica estão classificados. A expressão de classe 5.3 descreve `C1`, classificando todos os pacientes que possuem diagnóstico de Sarcoma descoberto antes dos 45 anos.

Algoritmo 5.3: *Expressão de Classe para definir da classe `C1`*

```
1   C1 EquivalentOf Patient and hasDocument some (Sarcoma and DiagsLess45)
```

Em seguida, seja uma classe `C2` que classifica todas as pessoas (não só pacientes) que possuem algum diagnóstico de câncer descoberto antes dos 45 anos, descrita pela expressão de classe 5.4.

Algoritmo 5.4: *Expressão de Classe para definir da classe `C2`*

```
1   C2 EquivalentOf Person and hasDocument some (Cancer_Diagnostic and DiagsLess45
        )
```

Por fim, seja a classe `C3` que classifica todas as pessoas que possuem algum diagnóstico de câncer antes dos 45 anos ou um diagnóstico de Sarcoma não importando a idade em que ele foi descoberto,

descrita pela expressão de classe 5.5. A Figura 5.12 também representa as classes `C1`, `C2` e `C3` com um possível resultado da classificação dos indivíduos pelo motor de inferência. É possível notar que alguns indivíduos pertencem tanto à classe `C2` quanto à classe `C3` por satisfazer a ambas as condições.

Algoritmo 5.5: *Expressão de Classe para definir da classe `C3`*

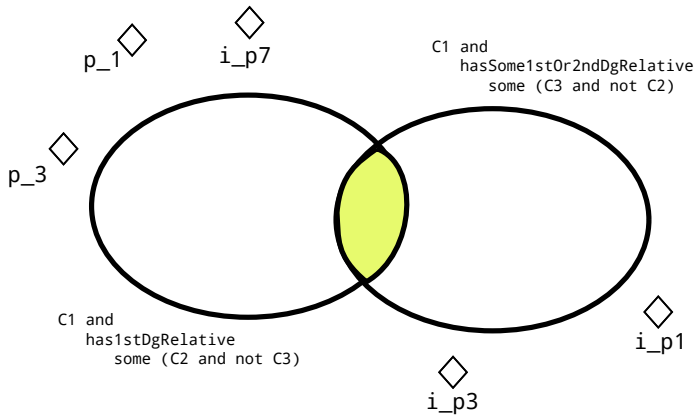
```
1   C3 EquivalentOf Person and ((hasDocument some (Cancer_Diagnostic and
        DiagsLess45))
2                              or  (hasDocument some Sarcoma))
```

Em seguida, definimos `Classic` como sendo um conceito capaz de classificar os indivíduos que atendam a todos os critérios da Síndrome de *Li-Fraumeni* Clássica, conforme a Tabela 3.2 na Seção 3.2.1, e descrita pela expressão de classe 5.6.

Algoritmo 5.6: *Expressão de Classe para definir a classe `Classic` e que gera resultados inconsistentes.*

```
1   Classic EquivalentOf C1 and has1stDgRelative some (C2 and not C3)
2                           and hasSome1stOr2ndDgRelative some (C3 and not C2)
```

Figura 5.13: *Resultado da aplicação da expressão de classe 5.6.*



Entretanto, ressaltamos a importância de separarmos os indivíduos de `C2` e `C3`, pois, se não o fizermos, poderemos selecionar dois indivíduos iguais (que representam a mesma pessoa) e tratarmos como se fosse dois indivíduos distintos. Ou seja, **`p_3`** poderá ser classificado como `Classic`, pois ele possui um parente de primeiro grau em `C2` e outro parente, de primeiro ou segundo grau, em `C3`. Só que esses parentes são, na verdade, a mesma pessoa (**`i_p3`**). Para evitar esses erros de classificação, introduzimos as cláusulas `C2 and not C3` para o primeiro critério e `C3 and not C2` para o segundo critério. Observando novamente na Figura 5.12 nota-se que o paciente **`p_1`** possui um parente de primeiro grau em `C2`, **`i_p1`**, e dois parentes de primeiro ou segundo grau em `C3`: **`i_p1`** e **`i_p7`**. De acordo com a classificação da Síndrome de *Li-Fraumeni* Clássica, o paciente **`p_1`** atende aos critérios clínicos da Síndrome de *Li-Fraumeni* Clássica e deve, para tanto, ser classificado como tal. Entretanto, ao aplicar a expressão de classe 5.6, o paciente **`p_1`** não será classificado como `Classic` (Figura 5.13), pois o indivíduo **`i_p1`**, que é seu parente de primeiro grau, está presente tanto na classe `C2` quanto na classe `C3`, violando a condição `(C2 and not C3)` da expressão de classe 5.6. Uma solução para esse problema seria garantir que o indivíduo presente no critério *`C2`*

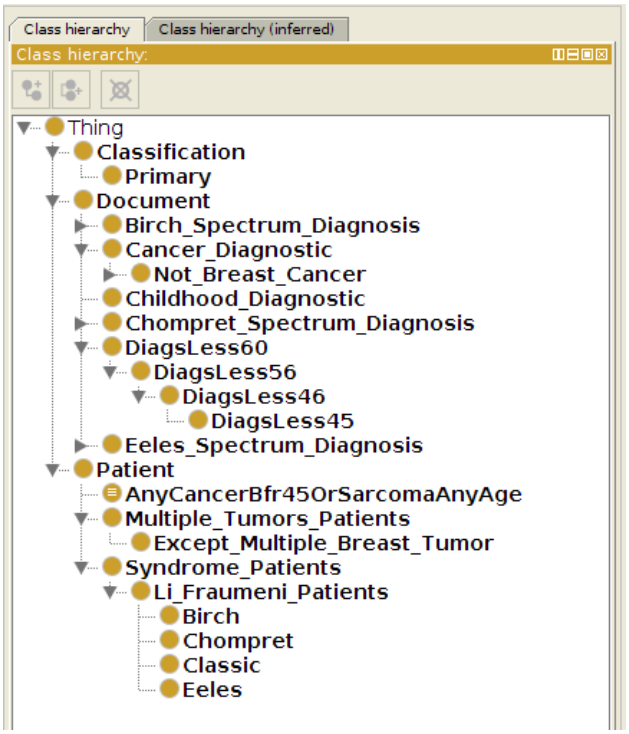
fosse diferente daquele presente no critério *C3*, algo que, com expressões de classe, não é possível alcançar, pois não existe um átomo do tipo `owl:differentFrom` nas expressões de classe. A regra 5.7 classifica os pacientes segundo o critério *Classic* sem a necessidade de nenhuma das expressões de classe acima.

Algoritmo 5.7: *Regra SWRL para a definição da classe* `Classic` *da Síndrome de Li-Fraumeni.*

```
1  Patient(?p), hasDocument(?p, ?diag_1), Sarcoma(?diag_1), DiagsLess45(?diag_1),
2   has1stDgRelative(?p, ?r_1), hasDocument(?r_1, ?diag_2), DiagsLess45(?diag_2),
3   hasSome1stOr2ndRelatives(?p, ?r_2), AnyCancerBfr45OrSarcomaAnyAge(?r_2),
4   DifferentFrom (?r_1, ?r_2) -> Classic(?p)
```

Observa-se a presença da classe `AnyCancerBfr45OrSarcomaAnyAge`, que descreve todos os indivíduos que possuem algum tipo de diagnóstico de câncer antes dos 45 anos ou diagnóstico de Sarcoma em qualquer idade. O propósito dessa classe é construir um conjunto baseado em uma disjunção, própria do critério *Classic*, e que não é permitido em regras SWRL (Cláusulas de Horn).

Figura 5.14: *Taxonomia da LFOnto.*



A classe `DiagsLess45` (Figura 5.14) foi definida como subclasse de `Document`, importada de ***CDOnto***, e descreve todos os documentos que foram emitidos para pacientes que tinham menos de 45 anos à época. A classificação de um diagnóstico como pertencente à classe `DiagsLess45` também é feita de forma automática através de uma regra SWRL. Essa abordagem permite que outras regras para idades de pacientes em diagnósticos sejam adicionadas posteriormente e, assim, a ontologia seja estendida para outros critérios (`DiagsLess60`, `DiagsLess56`, etc).

A classe `Multiple_Tumors_Patients` classifica todos os pacientes que tiveram mais de um diagnóstico de tumor primário. Esse conceito de tumores primários é utilizado na classificação de pacientes considerados LFL segundo o critério de Chompret. Foi possível classificar indivíduos como portadores de múltiplos tumores através da expressão de classe 5.8.

Algoritmo 5.8: *Regra SWRL para a definição da classe* `Multiple_Tumors_Patients` *da Síndrome de Li-Fraumeni.*

```
1  Patient and (hasDocument min 2 (Document and (hasDiagnosticCode some Primary)))
2              SubClassOf Multiple_Tumors_Patients
```

Outra classe também utilizada no critério de Chompret foi `Except_Multiple_Breast_Tumor`, em que são classificados pacientes que tiveram múltiplos tumores primários diferentes de Câncer de Mama. Optou-se por não usar a operação de complemento `(not Breast_Cancer)` na classificação desses diagnósticos em razão do grande tempo consumido para inferir todos os diagnósticos que não fizessem parte do grupo de múltiplos tumores primários de Câncer de Mama (ver Seção 2.2.5). Assim, foram criadas duas classes de apoio na ontologia *CDOnto*, chamadas `Breast_Cancer` e `Not_Breast_Cancer`, aonde foram classificados, respectivamente, todos os diagnósticos de Câncer de Mama e todos os diagnósticos que não são de Câncer de Mama. Os critérios utilizados para a separação dos diagnósticos entre essas duas classes foram definidos através de entrevistas com médicos oncologistas e estão de acordo com [WHO92] e [FPJ+00]. A expressão de classe 5.9 mostra como os indivíduos são classificados como membros de `Except_Multiple_Breast_Tumor`.

Algoritmo 5.9: *Regra SWRL para a definição da classe* `Except_Multiple_Breast_Tumor` *da Síndrome de Li-Fraumeni.*

```
1   Patient and (hasDocument min 2 ((Document and Not_Breast_Cancer)
2           and (hasDiagnosticCode some Primary))) SubClassOf
                Except_Multiple_Breast_Tumor
```

Por fim, na ***LFOnto***, foram criadas três classes igualmente importantes na classificação de pacientes dentro dos critérios LFL: a `Birch_Spectrum_Diagnosis`, `Eeles_Spectrum_Diagnosis` e `Chompret_Spectrum_Diagnosis`. Elas agrupam diagnósticos que atendem aos critérios dos espectros de cada critério, definidos pela literatura. No caso dos critérios para Chompret, os tumores que fazem parte do grupo de risco são sarcoma de tecido mole, osteosarcoma, câncer de mama pre-menopausal, tumor de cérebro, carcinoma adreno-cortical, leucemia ou câncer de pulmão[15]. Já o critério LFL para Birch ou Eeles, os tumores que fazem parte do grupo de risco são sarcoma, câncer de mama, câncer de cérebro, câncer adreno-cortical ou leucemia. Especificamente para o critério Birch, ainda fazem parte do critério os Melanomas e os cânceres de Próstata e Pâncreas[16]. Ainda sobre o critério de Birch, outro espectro de tumores é levado em consideração: o espectro dos tumores infantis (classe `Childhood_Diagnosis`). Nesse caso, todos os diagnósticos cuja idade do paciente no primeiro diagnóstico for abaixo de 18 anos[17] (inclusive) serão classificados como `Childhood_Diagnosis`.

Em seguida, testaremos o uso das ontologias desenvolvidas em casos de teste controlados de forma a validar os termos e conceitos a fim de classificar famílias e pacientes que atendem aos critérios da Síndrome de *Li-Fraumeni*.

[15] http://www.cancer.net/cancer-types/li-fraumeni-syndrome
[16] http://www.cancer.net/cancer-types/li-fraumeni-syndrome
[17] Não existe ao certo um intervalo bem definido estabelecendo o limite de idade para que um câncer seja considerado câncer infantil. Alguns estudos consideram a faixa etária entre 0 e 19 anos [Soc14]; outros levam em conta crianças e adolescentes até 14 anos. No Brasil, existe um consenso entre os médicos quanto à faixa etária, até os 18 anos, apesar de não existir nenhum estudo mais aprofundado sobre esse assunto.

Capítulo 6

Testes e Resultados

Este capítulo apresenta a metodologia utilizada para avaliar os resultados da classificação das famílias *Li-Fraumeni* utilizadas neste estudo (Seção 6.1). Em seguida (Seção 6.2), apresentaremos os resultados obtidos através dos casos de teste gerados por uma ferramenta automatizada (OpenGLiFS), construída com o objetivo de gerar, aleatoriamente, árvores genealógicas que apresentassem características dos quatro critérios que definem famílias *Li-Fraumeni*: *Classic*, Birch, Eeles e Chompret. Essas famílias geradas contribuíram para ajustar as regras criadas nas ontologias e encontrar possíveis anomalias em suas modelagens, pois elas foram geradas através de um processo simples em que não houve introdução de erros humanos (digitação) nem dados desnecessários ao processo de classificação. A construção dessa ferramenta será descrita na Seção 6.2.2 e a descrição do conjunto de testes, na Seção 6.2.3. Após os ajustes nas ontologias, o processo de classificação foi executado nas famílias *Li-Fraumeni* que foram extraídas diretamente da base de dados do *A.C. Camargo Cancer Center*. Os resultados dessa classificação são apresentados e comparados àqueles alcançados com as famílias *Li-Fraumeni* geradas aleatoriamente na Seção 6.3. Por fim, na Seção 7, discutiremos os resultados alcançados durante o processo de construção das ontologias e na classificação das famílias. Também proporemos a expansão da ontologia proposta para a modelagem de outra síndrome de caráter hereditário: Síndrome de *Lynch*.

6.1 Metodologia

Processos de inferência, dependendo do tamanho dos ABox e TBox, podem ser longos e consumir grande quantidade de memória. Em nosso cenário, espera-se que a classificação dos pacientes consuma grande quantidade de recursos computacionais, como tempo de CPU e memória física, pois o ABox esperado para a classificação de pacientes possui um grande número de famílias, cada uma contendo o maior número de informações possível sobre seu histórico. Isso resulta em uma grande quantidade de axiomas a serem utilizados pelo motor de inferência durante a tarefa de classificação. Consequentemente, a validação das regras pode demorar muito tempo se aplicada diretamente nos casos reais, inviabilizando o rastreio de erros e inconsistências nas regras e axiomas.

Assim, com o intuito de validar mais precisamente as regras de classificação nas ontologias e dentro de uma janela de tempo aceitável, os casos de testes foram gerados de maneira controlada e direcionada para cada um dos critérios Li-Fraumeni a serem validados. Essa etapa se fez necessária porque não existe garantia de que os arquivos de famílias reais estejam isentos de erros, sejam eles humanos (digitação) ou técnicos (dados corrompidos no sistema). Dessa forma, os resultados obtidos

de rodadas com os dados reais poderiam indicar inferências incorretas (famílias falso-positivas, por exemplo). Os testes implementados foram de natureza experimental e os resultados coletados foram analisados de acordo com os critérios de falsos-positivos e falsos-negativos. Com isso, os ajustes foram efetuados e as famílias foram submetidas, novamente, ao motor de inferência. Apenas após a correta classificação de todas as famílias de teste é que consideramos as ontologias prontas para a classificação dos casos reais. Sem a realização dessa etapa de testes, seria muito difícil avaliar a confiança da classificação das famílias reais, pois é possível que algumas dessas famílias possuam erros de dados (erros humanos ou falhas técnicas, como um dado corrompido) que comprometam o resultado final (uma família classificada segundo os critérios Clássicos de maneira incorreta, por exemplo). As etapas para a elaboração dos testes foram:

1. Construção de um software gerador de famílias *Li-Fraumeni* que serão usadas como famílias de teste (OpenGLiFS).

2. Usar o OpenGLIFS para gerar 51 famílias *Li-Fraumeni* para cada um dos quatro critérios da síndrome, totalizando 204 famílias. Essa amostra foi selecionada considerando um grau de confiança de xxx$ e uma margem de erro de y%.

3. Rodar o classificador nas 204 famílias, separadas por critério da Síndrome de *Li-Fraumeni*, e contabilizar a Matriz de Confusão. Com os resultados, calcularemos a Acurácia, a Sensibilidade, a Prevalência, a Cobertura (*Recall* ou TPR) e o valor *F-Measure*.

6.2 Testes

Para construir um conjunto de dados de teste consistente, desenvolvemos uma aplicação, usando linguagem Java, para gerar uma grande quantidade de famílias Li-Fraumeni. Batizada de OpenGLiFS, essa ferramenta gera diversos arquivos, um por família, e os salva em formato OWL/RDF (para uso pelo mecanismo de inferência) e GEDCOM (para fins de visualização), tornando possível a identificação de possíveis erros de modelagem das regras.

Foram encontradas duas soluções que têm a capacidade de gerar árvores genealógicas aleatórias. A primeira chama-se *Random Family Tree Generator* 3.1[1] e consiste em uma ferramenta on-line com a capacidade de gerar famílias baseadas em parâmetros como o intervalo, em anos, de simulação (determina a quantidade de gerações existentes). A ferramenta apresentou, nos testes, problemas de travamento e estouro de memória. A segunda ferramenta, chamada de *Random Royal Family Tree Generation*[2], é um algoritmo escrito em Python que gera descendentes segundo as leis de sucessão sálicas[3]. Alguns obstáculos foram determinantes para que essas ferramentas não fossem utilizadas na geração do conjunto de testes: *(i)* ambas as ferramentas não permitem o controle da quantidade de gerações criadas (profundidade da árvore), pois estas são criadas segundo parâmetros que são definidos aleatoriamente, como quantidade de indivíduos do sexo feminino ou a data da morte de um indivíduo, que pode ser impeditivo para a continuidade daquela linhagem; *(ii)* ambas as ferramentas geram apenas uma única família por simulação, aumentando consideravelmente o tempo necessário para gerar um conjunto de testes volumoso; *(iii)* ambas as ferramentas apresentaram problemas

[1] http://mcdemarco.net/tools/family-tree-generator/lineage.html
[2] https://www.reddit.com/r/worldbuilding/comments/1tpj38/random_royal_family_tree_generation/
[3] http://eprints.whiterose.ac.uk/6093/

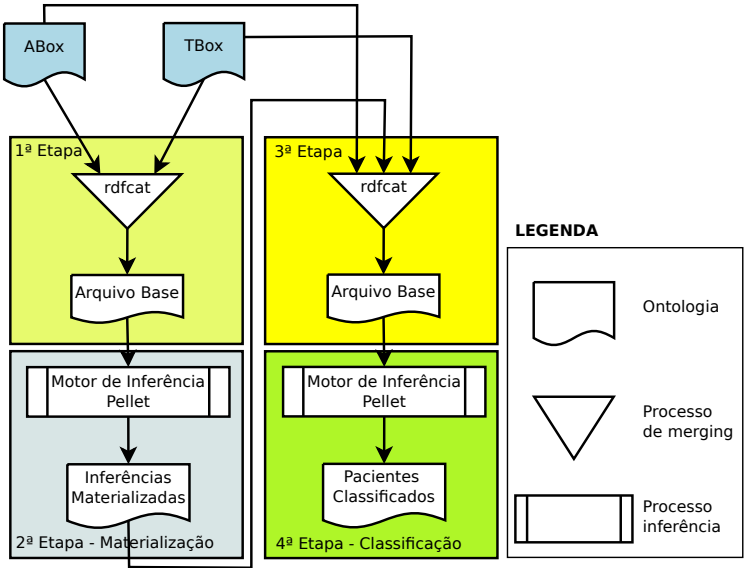
durante a execução da simulação. No caso da *Random Family Tree Generator* 3.1, a execução do código travou por diversas vezes. No caso do *Random Royal Family Tree Generation*, o código demorou mais de 13 segundos, em uma das simulações, para gerar uma única família, apesar de não ter travado sua execução em nenhuma das vezes. Além disso, ambos os códigos utilizavam dados que não eram importantes para a construção do conjunto de dados de teste. Assim, a decisão de construir um gerador aleatório de famílias Li-Fraumeni próprio se apresentou como uma solução aceitável em razão do tempo dedicado ao seu desenvolvimento ser menor do que a curva de aprendizado para modificar as soluções já existentes.

A seguir, será apresentado o gerador aleatório de famílias Li-Fraumeni bem como o seu processo de desenvolvimento e o conjunto de testes gerado por ela. Também apresentaremos o software desenvolvido Directed-Extract-LiFraumeni como ferramenta de classificação e extração das famílias *Li-Fraumeni*.

6.2.1 Directed-Extract-LiFraumeni - Ferramenta de Classificação e Extração Direta de Famílias *Li-Fraumeni*.

Conforme será descrito na Seção 6.4, o uso do motor de inferência Pellet apresentou instabilidades durante a classificação e extração das famílias. Seu uso, tanto como *plugin* no Protegé quanto como ferramenta *standalone* (uso por meio de linha de comando) não conseguia classificar todos os indivíduos de uma mesma família, ocasionando erro nos resultados apresentados. Uma das soluções para o problema seria a correção do algoritmo de classificação do próprio motor de inferência. O erro encontrado no Pellet está devidamente documentado no repositório de erros desde 2009, mas ainda não foi solucionado[4]. Corrigir o algoritmo, entretanto, fugia do escopo deste trabalho de pesquisa, o que nos levou adotar uma outra solução mais simples, porém com a penalização de aumentar o tempo de processamento.

Figura 6.1: *Etapas do processo de classificação das famílias Li-Fraumeni.*



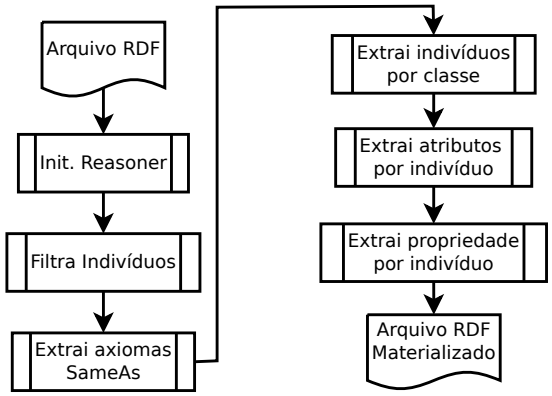
Baseado na ferramenta *open-source* `Directed-Inference`[5], desenvolvida pela equipe de Informática Médica do *A.C. Camargo Cancer Center* para a materialização dos dados oriundos das

[4] http://clark-parsia.trac.cvsdude.com/pellet-devel/ticket/420
[5] Disponível em https://github.com/djogopatrao/directed-inference

diversas bases de dados no repositório de triplas, desenvolvemos outra cujo objetivo é materializar as inferência da Síndrome de *Li-Fraumeni*. Adotamos, para isso, um processo de classificação das famílias de teste em quatro etapas, duas delas usadas para unir (*merge*) `ABox` e `TBox` (1ª e 3ª etapas) e duas outras para classificar os pacientes (2ª e 4ª etapas) por meio do motor de inferência Pellet (Figura 6.1). As etapas de classificação (2ª e 4ª etapas) foram compostas por mais seis etapas, todas dependentes do motor de inferência Pellet. Podemos descrever o que é realizado em cada etapa da Figura 6.2 conforme segue:

- **Init. Reasoner**: O motor de inferência Pellet é inicializado sobre a ontologia de entrada e todas as inferências possíveis são extraídas e salvas em memória.
- **Filtra indivíduos**: A ferramente filtra apenas indivíduos cujos IRI's foram passados como parâmetro de entrada. Se esse parâmetro for vazio, então a ferramenta irá processar a classificação para todos os indivíduos.
- **Extrai axiomas SameAs**: Tenta inferir todos os indivíduos que são iguais entre si, evitando, assim, que regras e axiomas sejam aplicados separadamente para indivíduos que representam a mesma entidade real. Por exemplo: se um mesmo indivíduo tiver nomes diferentes em dois sistemas distintos, de onde eles foram importados, então a ferramenta poderá tratá-los como se fossem o mesmo indivíduo, desde que sejam referenciados pela mesma chave (RGH, por exemplo).
- **Extrai indivíduos por classe**: Ocorre a extração dos indivíduos que foram classificados em cada uma das classes da ontologia.
- **Extrai propriedades por indivíduo**: Para cada indivíduo filtrado, serão extraídas suas propriedades, inferidas ou não.
- **Extrai atributos por indivíduo**: De forma análoga à etapa anterior, serão extraídos os atributos de todos os indivíduos filtrados.

Figura 6.2: *Etapas executadas pela ferramenta* `Directed-Extract-LiFraumeni`.



Por fim, os dados extraídos em cada etapa serão materializados em um novo arquivo de saída no formato RDF para que o mesmo seja utilizado, ou como entrada para um novo processo de classificação (ou *merging*), ou como saída definitiva.

6.2.2 OpenGLiFS - Open Genealogical Li-Fraumeni Families Generator

Para construção dos arquivos de teste, foi desenvolvida uma ferramenta (OpenGLiFS), usando linguagem Java, que gera, aleatoriamente, um conjunto de famílias segundo parâmetros passados como argumento na chamada principal, como quantidade de famílias geradas, número de gerações (profundidade da árvore genealógica), o intervalo mínimo e máximo de filhos que cada casal pode gerar e o tipo de síndrome que cada família deveria possuir (*Classic*, Birch, Eeles e Chompret). A estrutura gerada em cada árvore genealógica respeita as propriedades e as classes definidas nas ontologias Genonto, CDOnto e na LFOnto.

Figura 6.3: *Tipos de nós diferentes: Indivíduo Solteiro e Casal.*

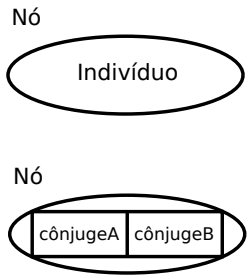
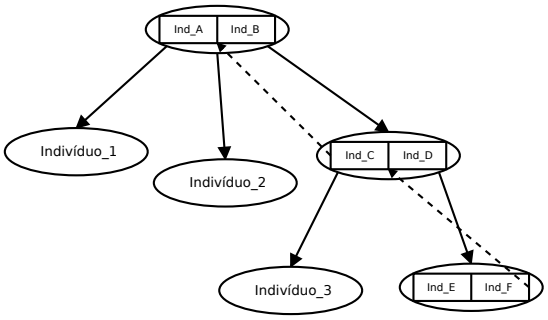


Figura 6.4: *Exemplo da estrutura de uma árvore genealógica.*



A estrutura criada para a geração das famílias é simples e pode ser adaptada facilmente para qualquer tipo de síndrome familiar. A ferramenta gera uma família usando uma estrutura de dados em forma de árvore (Figura 6.4), em que cada nó dessa árvore pode ser um indivíduo solteiro (nó `Individuo`) ou um novo casal (nó `Casal`, formado por dois nós `Individuo`, nomeados `ConjugeA` e `ConjugeB`). Nesse último caso, um dos cônjuges pertencerá à linhagem genética da família, enquanto o outro cônjuge será representado sem as informações dos seus progenitores. A Figura 6.3 representa os dois tipos de nós existentes. Em um nó do tipo `Casal`, apenas um dos cônjuges possui um ponteiro levando até seus progenitores, conforme pode ser observado na Figura 6.4, em que uma linha tracejada liga o cônjuge que pertence à linhagem dessa família a seus pais (no exemplo, o indivíduo `Ind_C` é filho do casal formado pelos indivíduos `Ind_A` e `Ind_B` e o indivíduo `Ind_D` não tem seus progenitores representados). Mostraremos, a seguir, os algoritmos que são responsáveis por criar toda a estrutura do heredograma em uma família hipotética.

O construtor, representado pelo pseudocódigo 6.1, assume a responsabilidade de atribuir o primeiro casal à raiz da árvore genealógica. Em seguida, é chamado o método responsável por construir toda a árvore genealógica, segundo os atributos passados, a partir desse casal progenitor.

Algoritmo 6.1: *Pseudocódigo do Construtor* `FamilyTree()`

```
1    FamilyTree(casal, num_geracoes, criterio_lifraumeni, nome_familia){
2        max_geracoes <- num_geracoes
3        criterio_familia <- criterio_lifraumeni
4        sobrenome <- nome_familia
5        raiz <- casal
6    }
```

Os algoritmos 6.2 e 6.3 constroem o restante da árvore genealógica. A chamada inicial do método `GerarFamilia()` é `GerarFamilia(0,4,6)`. Os valores 4 e 6 foram escolhidos para garantir que cada geração tenha uma quantidade razoável de irmãos, aumentando, assim, a possibilidade de que essa geração venha a ter descendentes. Este, por sua vez, resgata o nó raiz e inicia a recursão na linha 3. Como `lista_de_individuos`, nesta primeira chamada, só conterá o nó raiz (que é um casal), criado pelo construtor, então a construção da geração está garantida a partir do casal progenitor. Em seguida, o algoritmo escolhe aleatoriamente um probando (linha 5), garantidas as condições para tal (algoritmo 6.5). Por fim, as linhas 6 até 9 garantem um pequeno percentual aleatório de famílias isentas da síndrome (algoritmo 6.6), conforme será explicado na Seção 6.2.3.

Algoritmo 6.2: *Listagem da função `GerarFamilia()`*

```
1   GerarFamilia(num_geracoes, min_irmaos, max_irmaos){
2       lista_de_individuos <- RetornaRaizDaArvore();
3       ConstroiGeracao(lista_de_individuos, 0, min_irmaos, max_irmaos);
4
5       probando <- EscolheProbando();
6       sorteio <- EscolhaAleatoria(0,1);
7       if (sorteio >= 0.2) then {
8           GerarCasosCancer();
9       }
10  }
```

O algoritmo 6.3 é recursivo e percorre a árvore genealógica **em largura**. Ou seja, ele primeiro percorre todos os nós da geração atual (linha 4) construindo a geração seguinte com os filhos dos nós `Casal` da geração atual (linhas 8 e 11). Em seguida, recursivamente, avança uma geração, passando todos os indivíduos da geração seguinte (linha 15) como parâmetro.

Algoritmo 6.3: *Pseudocódigo da função `ConstroiGeracao()`*

```
1   ConstroiGeracao(lista_de_individuos_geracao, num_geracoes, min_irmaos,
        max_irmaos){
2       if (num_geracoes <= max_geracoes) then {
3           filhos_da_geracao <- null;
4           for (cada nó em lista_de_individuos_geracao) do{
5               if (nó é Individuo) then{
6                   ▷criar atributos do indivíduo;
7               } else {
8                   filhos_do_casal <- ReproduzCasal(min_irmaos, max_irmaos);
9                   for (cada nó_filho em filhos_do_casal) do {
10                      adiciona_no(nó_filho,nó);
11                      adiciona_no(nó_filho,filhos_da_geracao);
12                  }
13              }
14          }
15           ConstroiGeracao(filhos_da_geracao, num_geracoes++, min_irmaos,
                max_irmaos);
16      }
17  }
```

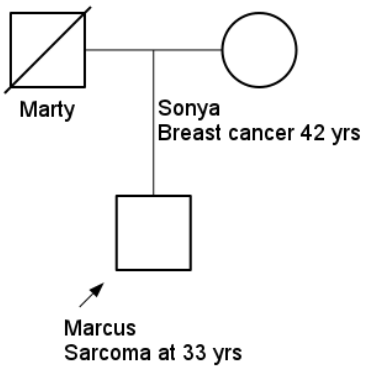
O algoritmo 6.4 apresenta, de maneira genérica, como ocorre a construção dos filhos de um casal.

Algoritmo 6.4: *Pseudocódigo da função* `ReproduzCasal()`

```
1   Lista de nó ReproduzCasal(min_irmaos, max_irmaos){
2       lista_de_filhos <- null;
3       for (i <-min_irmaos; i<=max_irmaos; i++){
4           if (EscolhaAleatoria(0,1) <= 0.5) {
5               novo_no <- criar um nó tipo Individuo;
6               ▷ define os atributos do nó Individuo
7           } else {
8               novo_no <- criar um nó tipo Casal;
9               ▷ define os atributos do nó Casal
10          }
11          adiciona_no(novo_no,lista_de_filhos);
12      }
13      return lista_de_filhos;
14  }
```

Em seguida, escolhe-se, aleatoriamente, um probando dentre todos os indivíduos existentes nessa família (algoritmo 6.5). Essa escolha, por mais aleatória que se faça, precisa ser feita de maneira direcionada a fim de que ela não escolha um probando que seja incapaz de possuir as características necessárias a um determinado tipo de família *Li-Fraumeni* (não possuir parente de primeiro grau ou mais de um parente de segundo grau, por exemplo).

Figura 6.5: *Exemplo de uma família gerada automaticamente em que o probando não possui parentes suficientes para ser classificado como uma família Li-Fraumeni clássica.*



A Figura 6.5 mostra uma parte da árvore genealógica de uma família gerada aleatoriamente que possui apenas 2 gerações em que um dos parentes de primeiro grau é falecido por causas outras. Como, nesta família, o probando não possui mais de 2 parentes de primeiro ou segundo graus (condição para que haja a possibilidade de uma família ser classificada como *Li-Fraumeni Classic*), então ela não serve como caso de teste. A condição determinante para que um indivíduo seja escolhido probando é que ele tenha, pelo menos, dois parentes de primeiro grau e dois parentes de segundo grau, pois cada um dos critérios de classificação *Li-Fraumeni* exige, para o probando, pelo menos um parente de primeiro grau e um parente de segundo grau.

Algoritmo 6.5: *Pseudocódigo da função* `EscolheProbando()`

```
1   nó EscolheProbando(){
2       escolha_certa <- false;
3       while (!escolha_certa) do {
4           probando <- escolha aleatoria entre todos os individuos da familia
```

```
5           if (probando tem pelo menos 2 parentes diferentes de primeiro grau) AND
                (probando tem pelo menos 2 parentes diferentes de segundo grau)
                then {
6                   escolha_certa <- true;
7           }
8       }
9       return probando;
10  }
```

Algoritmo 6.6: *Pseudocódigo da função* `GerarCasosCancer()`

```
1   GerarCasosCancer(){
2       case (criterio_familia){
3
4       Classic:{
5           probando <- RetornaProbando();
6           ▷atribua casos de cancer aos parentes de primeiro e segundo graus,
                segundo as condições da Li-Fraumeni Clássica
7           }
8       Eeles:{
9           probando <- RetornaProbando();
10          ▷atribua casos de cancer aos parentes de primeiro e segundo graus,
                segundo as condições de Eeles
11          }
12      Birch:{
13          probando <- RetornaProbando();
14          ▷atribua casos de cancer aos parentes de primeiro e segundo graus,
                segundo as condições de Birch
15          }
16      Chompret:{
17          probando <- RetornaProbando();
18          ▷atribua casos de cancer aos parentes de primeiro e segundo graus,
                segundo as condições de Chompret
19          }
20      }
21  }
```

Posteriormente, as ocorrências de câncer são criadas para o probando e para seus parentes de primeiro e segundo graus, de acordo com os critérios estabelecidos para cada tipo da Síndrome de *Li-Fraumeni*. Outros dados também são criados para cada indivíduo acometido por um caso de tumor. Com o intuito de deixar as famílias geradas com o mínimo de informação possível, apenas os dados envolvidos no processo de inferência foram criados, como idade do paciente no momento do primeiro diagnóstico, identificador do documento diagnóstico e o código CID do câncer que acometeu o paciente, além do seu nome para fins de identificação.

Uma das limitações impostas durante o processo de criação das famílias foi o limite máximo de gerações e de filhos por casal, em cada geração. Essa limitação se fez necessária por duas razões básicas: *(i)* deixar o tamanho das famílias geradas aleatoriamente o mais próximo possível do tamanho das famílias reais cadastradas nos sistemas do *A.C. Camargo Cancer Center* e; *(ii)* limitar o uso de memória do computador nos testes de inferência. De fato, durante o processo de coleta de informações de um probando, nem sempre o mesmo possui todos os dados de seus familiares, deixando lacunas de informação nos arquivos de famílias. Em nossas famílias geradas aleatoriamente,

consideramos que o probando sabe apenas o suficiente para classificar sua família como portadora de uma mutação LFS ou LFL. Outro ponto a ser considerado determinante para a limitação do número de gerações é a quantidade de memória alocada nos testes de inferência. O uso de memória durante esse processo pode crescer exponencialmente e, dessa maneira, ocasionar estouro de limite no uso do *heap space*[6] da Máquina Virtual Java (JVM).

Por fim, cada árvore genealógica criada é salva em dois formatos distintos: OWL/RDF, para uso nos testes de inferência, e GEDCOM, para que a estrutura da árvore possa ser visualizada graficamente em qualquer aplicativo que processe esse tipo de formato. O formato GEDCOM[7] é considerado um padrão para troca de informações genealógicas entre aplicações e largamente utilizado por diversas ferramentas, proprietárias ou não. A estrutura de dados usada nos arquivos GEDCOM é baseada em estruturas hierárquicas de marcas (*tags*) que codificam nomes, famílias, origens, eventos, etc. A listagem 6.7 representa um exemplo do código escrito no formato GEDCOM para uma família pequena. Apresentamos na Seção 9.3, dois heredogramas gerados automaticamente pela ferramenta OpenGLIFS e renderizados pela ferramenta *on-line* Invitae[8]: a Figura 9.6, positiva para o critério Eeles, e a Figura 9.7, positiva para o critério *Classic*.

Algoritmo 6.7: *Exemplo de um extrato de um arquivo GEDCOM.*

```
0 HEAD
1 GEDC
2 VERS 5.5
2 FORM LINEAGE-LINKED
1 CHAR UTF-8
1 LANG Portuguese_Brazil
1 SOUR MYHERITAGE
2 NAME Minha Família
2 VERS 7.0.0.7143
2 _RTLSAVE RTL
2 CORP MyHeritage.com
1 SUBM @U1@
1 DEST MYHERITAGE
1 DATE 07 JAN 2016
2 TIME 21:59:45 GMT-3
1 _RINS I4,F1,N0,M0,R0,S0,U1,L0,P0,Q0,IF4,FF1,SCd
1 _UID 568F0851F322C11D40024E8C76C32220
1 _DESCRIPTION_AWARE Y
0 @U1@ SUBM
1 RIN MH:U1
0 @I1@ INDI
1 RIN MH:I1
1 _UID 568F089BB9AA511D90024E8C76C32220
1 _UPD 07 JAN 2016 21:53:47 GMT-3
1 NAME Carlos /Andrade/
2 GIVN Carlos
2 SURN Andrade
1 SEX M
...
```

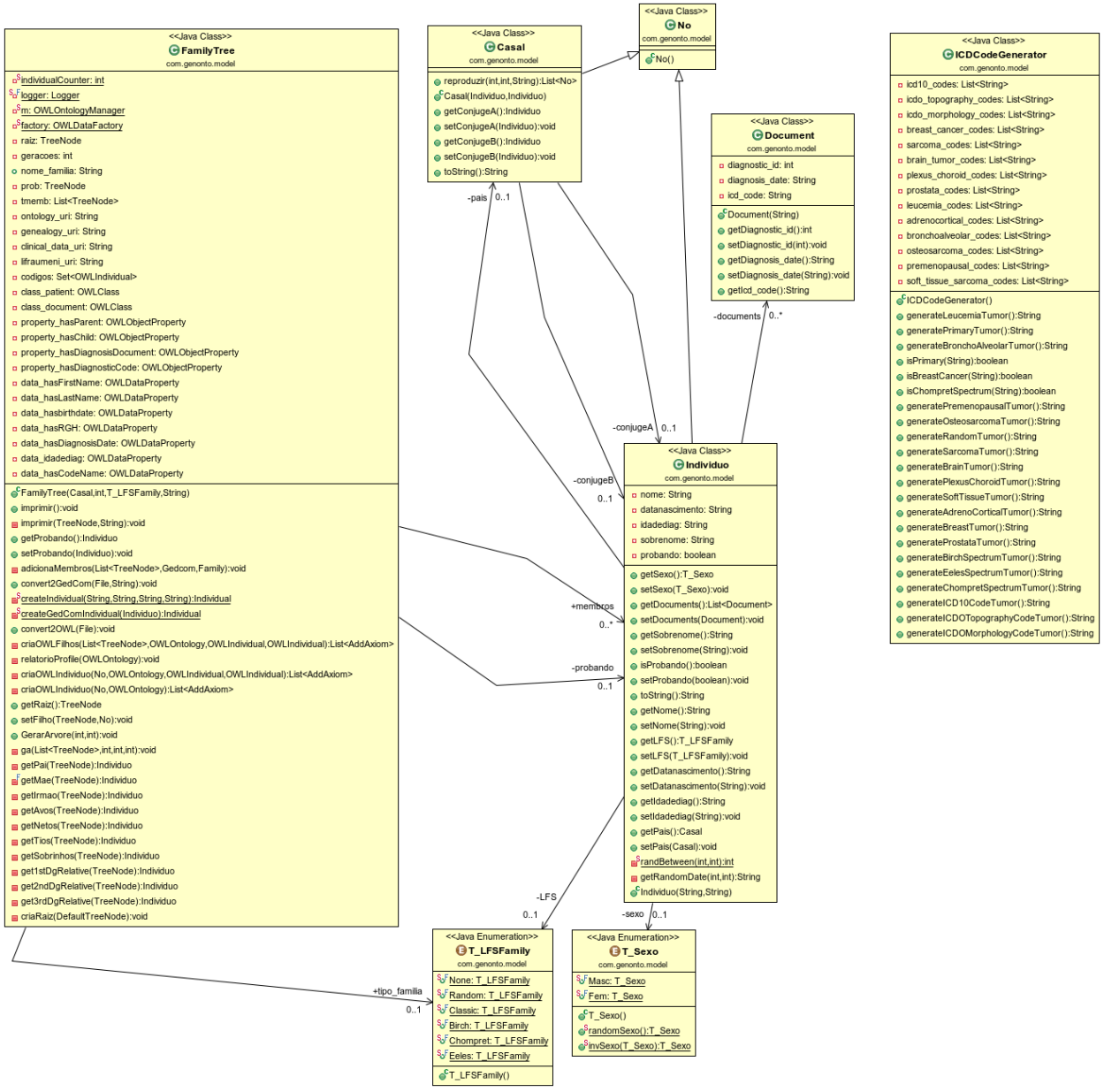
[6] Java *Heap Space* é a área de memória alocada para as aplicações Java. Ela é criada e gerenciada pela JVM - *Java Virtual Machine*. Fonte: http://javaeesupportpatterns.blogspot.com.br/2012/02/java-heap-space-what-is-it.html.

[7] http://homepages.rootsweb.ancestry.com/~pmcbride/gedcom/55gctoc.htm

[8] https://familyhistory.invitae.com/login/?next=/pedigrees/

O formato GEDCOM, entretanto, não possui nenhuma estrutura que permita o uso de mecanismos de inferência para descoberta de conhecimento. Isso faz dele apenas uma estrutura de dados para o armazenamento de indivíduos, famílias e eventos. Por essa razão, usamos nossa ferramenta para converter a árvore genealógica no formato OWL/RDF a fim de que possa ser submetido a um mecanismo de inferência. Um Diagrama de Classes completo com todos os métodos e atributos das classes da OpenGLiFS pode ser observado na Figura 6.6.

Figura 6.6: *Diagrama de Classe UML da ferramenta geradora de famílias aleatórias OpenGLiFS.*



A seguir, descreveremos o conjunto de testes usado na classificação de indivíduos, bem como os resultados alcançados durante essa etapa.

6.2.3 Conjunto de testes

Conforme explicado anteriormente, um conjunto de testes foi usado com o objetivo de verificar os erros e acertos das regras de inferência usadas na classificação dos pacientes. Ele foi gerado por

meio da ferramenta OpenGLiFS, desenvolvida especificamente para essa finalidade, pois, devido à Síndrome de *Li-Fraumeni* ser uma síndrome rara no mundo, a sua ocorrência no Brasil é igualmente baixa, o que nos proporcionaria uma base de testes com poucos casos. Ainda neste viés, como o objetivo era apenas testar a correção das regras, os arquivos de teste foram gerados com o mínimo de ocorrências de casos de câncer; apenas o necessário para identificação de uma determinada classificação foi gerado em cada um desses arquivo. Espera-se, com isso, que as diferenças entre o processo de inferência nos arquivos de teste e nos arquivos reais residam somente no campo do tempo de inferência e da memória utilizada, e não na precisão da classificação ou nas taxas de falsos positivos ou falsos negativos.

Assim, para cada um dos quatro critérios *Li-Fraumeni*, foram gerados 51 arquivos de testes, totalizando 204 arquivos, cada um contendo uma única família com vários indivíduos. A quantidade de indivíduos variou, de uma família para outra, influenciada pela quantidade de gerações e pelo número de casais em cada geração. Ou seja, se uma determinada geração em uma família fosse criada aleatoriamente e nenhum casal fosse gerado, então o heredograma dessa família deveria finalizar nessa mesma geração. Para cada critério *Li-Fraumeni*, cada família recebeu, aleatoriamente, um conjunto mínimo de condições para que a mesma fosse identificada dentro desse critério, conforme descrito na Seção 3.2. Ainda, para cada conjunto de teste de um determinado critério, um número, sorteado aleatoriamente, de famílias não reuniu as condições mínimas necessárias para a sua classificação, devendo ser identificadas como casos negativos da síndrome. Queremos, com isso, introduzir alguns poucos casos negativos na amostra a fim de verificar a taxa de **falsos-positivos**. De maneira resumida, para cada um dos critérios *Li-Fraumeni*, temos:

$$51 \text{ casos } \begin{cases} x \text{ positivos} \\ 51 - x \text{ negativos} \end{cases}$$

, em que x é um valor inteiro definido aleatoriamente (sempre maior que a metade dos casos mais um).

A Tabela 6.1 relaciona cada um dos conjuntos de teste usados para cada um dos critérios de classificação *Li-Fraumeni* com a quantidade de indivíduos que cada família possui. O sinal + ao lado de cada família indica que a mesma deverá ser classificada como portadora da Síndrome de *Li-Fraumeni* no seu respectivo critério. A seguir, iremos descrever os resultados obtidos pelo motor de inferência usando os casos de teste. Apresentaremos também uma Matriz de Confusão para cada caso, aonde iremos discutir as taxas de Precisão (*precision*), Sensibilidade (*sensitivity*), Prevalência (*prevalence*) e de Acurácia (*accuracy*).

6.3 Classificação

Tanto os testes quanto o processo de classificação usando os arquivos de famílias do *A.C. Camargo Cancer Center* foram realizados no servidor `Brucutu`, da Universidade de São Paulo, que possui 24 processadores Intel® Xeon® com frequência de 2.4GHz cada, 6 núcleos por processador, 2 *threads* por núcleo, arquitetura de 64 bits, *cache* L1 de 32kbytes, L2 de 256kbytes e L3 de 112Mbytes, e 126Gbytes de memória RAM. O sistema operacional utilizado por essa máquina é o Debian versão 8.2 de 64 bits e *kernell* Linux versão 3.16 e a máquina virtual Java® (JVM) foi a 1.7.0_91 (JDK `IcedTea` 2.6.3). Durante os testes, foi configurado um *heap space* de 10Gbytes

Tabela 6.1: *Tabela resumo dos casos de teste para os quatro critérios Li-Fraumeni.*

Famílias Eeles	Indiv.		Famílias Birch	Indiv.		Famílias Classic	Indiv.		Famílias Chompret	Indiv.	
0_Airal.owl	724		0_Ed.owl	142	+	0_Shauz.owl	112	+	0_Vjint.owl	292	+
1_Ais.owl	1017	+	1_Brauvaust.owl	96	+	1_Jghough.owl	6		1_Ek.owl	291	
2_Memib.owl	12	+	2_Ut.owl	119	+	2_Leiv.owl	280	+	2_Jitoib.owl	247	+
3_Choob.owl	416	+	3_Oowoist.owl	96	+	3_Oon.owl	243	+	3_Aush.owl	4	+
4_Auf.owl	301	+	4_Jikoist.owl	516	+	4_Glar.owl	132	+	4_Autjir.owl	263	
5_Vessouss.owl	254	+	5_Eihat.owl	93		5_Caf.owl	587	+	5_Jil.owl	277	+
6_Prof.owl	230	+	6_Ustouh.owl	350	+	6_Eiv.owl	195	+	6_Diss.owl	405	+
7_Stauvath.owl	51	+	7_Yk.owl	516	+	7_Auf.owl	206	+	7_Aumjith.owl	195	+
8_Yp.owl	181	+	8_Kjmaib.owl	9		8_Oom.owl	477	+	8_Vjd.owl	144	+
9_Ooshoosh.owl	105	+	9_Kloghjipf.owl	706	+	9_Oubjis.owl	160	+	9_Sharjl.owl	6	
10_Hoipfjig.owl	190	+	10_Eil.owl	178	+	10_Ab.owl	575	+	10_Ott.owl	139	+
11_Yzutt.owl	84	+	11_Fjissjch.owl	301	+	11_Ost.owl	6		11_Auth.owl	198	+
12_Thet.owl	71		12_Aik.owl	474	+	12_Oinont.owl	442	+	12_Eissaut.owl	124	
13_Audooz.owl	681	+	13_Ast.owl	7		13_Oish.owl	694	+	13_Roughint.owl	669	+
14_Aud.owl	209	+	14_Auloh.owl	396	+	14_Eintait.owl	6		14_Ouboud.owl	5	+
15_Heithjib.owl	292	+	15_Lywysh.owl	3		15_Blussoun.owl	141	+	15_Auttoz.owl	445	+
16_Audis.owl	530	+	16_Aush.owl	171	+	16_Oiw.owl	55	+	16_Jw.owl	632	+
17_Oissaus.owl	160	+	17_Vaitt.owl	203	+	17_Ybouss.owl	268	+	17_Egjz.owl	200	+
18_Eintainn.owl	6		18_Ybaun.owl	249	+	18_Keipf.owl	85	+	18_Oist.owl	133	+
19_Eiv.owl	439	+	19_Stud.owl	232	+	19_Ub.owl	491	+	19_Theihas.owl	234	
20_Qyg.owl	277		20_Nithot.owl	291	+	20_Sysh.owl	147	+	20_Evish.owl	9	
21_Ed.owl	68	+	21_Aimov.owl	82	+	21_Bloif.owl	702	+	21_Flainauk.owl	283	+
22_Oibjin.owl	95	+	22_Yk.owl	109	+	22_Out.owl	8		22_Djpfauth.owl	317	+
23_Igh.owl	239	+	23_Dageil.owl	2		23_Klainnaun.owl	223	+	23_Tazuch.owl	299	
24_Pout.owl	87	+	24_Ouk.owl	346	+	24_Epow.owl	306	+	24_Gloughouch.owl	496	+
25_Stadjish.owl	189	+	25_Blook.owl	369	+	25_Yst.owl	100	+	25_It.owl	562	+
26_Fleiw.owl	8		26_Qjivatt.owl	251	+	26_Noozjd.owl	51	+	26_Bleistyr.owl	562	+
27_Toist.owl	66	+	27_Shonn.owl	336	+	27_Ewath.owl	24	+	27_Jnn.owl	45	
28_Echoif.owl	255	+	28_Sypfeiw.owl	477	+	28_Xouzjish.owl	315	+	28_Qootaigh.owl	211	+
29_Geih.owl	324	+	29_Xoodaih.owl	353	+	29_Tyweik.owl	118	+	29_Enn.owl	244	
30_Aufauw.owl	220	+	30_Af.owl	298		30_Apfim.owl	598	+	30_Yzach.owl	740	
31_Bjss.owl	156	+	31_Dozynt.owl	159	+	31_Hoiv.owl	710	+	31_Uthoun.owl	740	+
32_Ykit.owl	423	+	32_Eig.owl	346	+	32_Flysh.owl	149	+	32_Ooz.owl	7	+
33_Ov.owl	129	+	33_Fooshih.owl	7		33_Brun.owl	13	+	33_Epf.owl	461	+
34_Jss.owl	380	+	34_Bam.owl	301		34_Booveip.owl	272	+	34_Floud.owl	474	+
35_Aunned.owl	234	+	35_Rypfoog.owl	160	+	35_Brozop.owl	473	+	35_Up.owl	520	+
36_Kym.owl	628	+	36_Broiveitt.owl	116	+	36_Aussatt.owl	150	+	36_Tooh.owl	224	+
37_Haith.owl	76	+	37_Bleikjb.owl	653	+	37_Oul.owl	5		37_Poukip.owl	224	+
38_Pouf.owl	171	+	38_Oul.owl	5		38_Shennauf.owl	6		38_Oil.owl	231	
39_Glaisjid.owl	421	+	39_Beitjit.owl	445	+	39_Djich.owl	432	+	39_Gaik.owl	62	+
40_Aizoim.owl	194	+	40_Xour.owl	237	+	40_Oghah.owl	447	+	40_Ynnjigh.owl	76	+
41_Kjs.owl	227	+	41_Ooch.owl	81	+	41_Ith.owl	10	+	41_Up.owl	76	+
42_Yhoup.owl	8		42_Hath.owl	65	+	42_Jistoof.owl	164	+	42_Midyf.owl	129	+
43_Oigh.owl	163	+	43_Im.owl	53	+	43_Jizuss.owl	74	+	43_Stjiwim.owl	343	+
44_Jsh.owl	296	+	44_Kywuk.owl	277	+	44_Oustaunn.owl	693	+	44_Dailil.owl	188	+
45_Gaugett.owl	281	+	45_Anuf.owl	717	+	45_Gonneh.owl	544	+	45_Jich.owl	103	+
46_Mour.owl	132	+	46_Uchaz.owl	190	+	46_Djgypf.owl	146	+	46_Brybois.owl	309	+
47_Aih.owl	216	+	47_Esoinn.owl	122		47_Kadjir.owl	6		47_Destoud.owl	298	
48_Oodes.owl	6		48_Nedjih.owl	458	+	48_Klowoss.owl	504	+	48_Naukych.owl	436	
49_Oint.owl	688	+	49_Jp.owl	277	+	49_Klauk.owl	243	+	49_Dessaul.owl	256	
50_Imauf.owl	492	+	50_Akeiz.owl	662	+	50_Chen.owl	134	+	50_Oonnant.owl	607	

O sinal +, ao lado de cada família, indica que a mesma foi gerada atendendo às condições do seu respectivo critério.

para o processo a fim de que se evitasse esgotamento de memória durante sua execução. A Tabela 6.2 mostra um resumo das configurações.

6.4 Resultados para o conjunto de testes

O processo completo de inferência levou aproximadamente 29 horas para finalizar a classificação das 204 famílias e foi executado como segue:

1. Foi realizado um processo de fusão (*merge*) entre o `ABox` (arquivo com os dados das famílias

Tabela 6.2: *Resumo do hardware, sistema operacional e ambiente Java utilizados.*

Hardware	
Cpu	24
Arquitetura	64 bits
Frequência por CPU	2.4GHz
Núcleos por CPU	6
Threads por Núcleo	2
Cache L1 - Dados	32kb
Cache L1 - Instr.	32kb
Cache L2	256kb
Cache L3	12Mb
Memória RAM Total	126Gb
Sistema Operacional	
Distro	Debian
Versão Distro	8.2
Versão Kernel	3.16
Arquitetura	64 bits
Java	
Versão JVM	1.7.0_91
Versão JDK	IcedTea 2.6.3
Heap Space	10Gbytes

e indivíduos) e o `TBox` por meio da ferramenta `rdfcat`. Essa etapa se fez necessária em razão do motor de inferência apresentar instabilidades quanto às marcações de importação da linguagem OWL (`owl:imports`);

2. Foi executada a 1ª fase de inferência: o algoritmo de extração `Directed-Extract-LiFraumeni` é executado, usando o arquivo resultante da etapa anterior, e materializa as inferências descobertas em forma de triplas rdf para um arquivo rdf usado como base;

3. Foi realizado um novo processo de fusão entre o `ABox`, o `TBox` e o arquivo base gerado na etapa anterior, por meio da ferramenta `rdfcat`.

4. Foi executada a 2ª fase de inferência: o algoritmo de extração `Directed-Extract-LiFraumeni` é executado, usando o arquivo resultante da etapa anterior, e materializa, definitivamente, as inferências descobertas em forma de triplas `rdf` para um arquivo usado como saída;

A inferência em duas etapas foi necessária porque o motor de inferência Pellet, usado nesse experimento, possui um *bug* documentado (e já relatado em outros trabalhos, como [DCtTdK11]) que resulta em uma ordenação parcial incorreta das classes a serem processadas. Com isso, alguns indivíduos deixam de ser classificados corretamente, mesmo que estes atendam aos critérios estabelecidos nos axiomas e regras SWRL. Um exemplo do que foi descrito ocorreu durante os testes com o critério *Li-Fraumeni* Clássico, em que um determinado indivíduo possuía todas as condições para ser classificado como portador da Síndrome de *Li-Fraumeni* clássica, mas nunca era classificado como tal. O uso de consultas *SPARQL* construídas para selecionar os indivíduos que atendiam ao critério Classic retornava corretamente os resultados esperados (consulta 6.8), o que indicava que algo no processo de inferência do Pellet não estava funcionando corretamente. Após uma consulta

no seu repositório de problemas[9], descobrimos se tratar de um problema de ordenação parcial das classes, o que foi comprovado ao submetermos o resultado da primeira inferência novamente ao algoritmo de extração `Directed-Extract-LiFraumeni` e conseguirmos obter o resultado esperado. Optamos, assim, por conduzir a pesquisa usando essa abordagem de duas etapas, pois a outra alternativa fugia ao escopo deste trabalho (corrigir o algoritmo de inferência no Pellet).

O resultado foi tabulado separadamente para cada critério e será descrito na próxima seção.

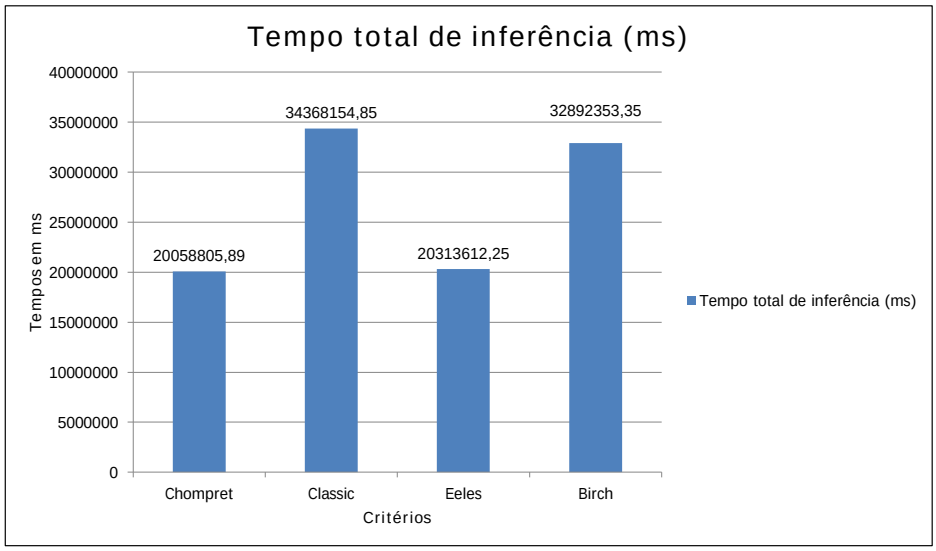
Algoritmo 6.8: *Consulta SparQl para retornar todos os indivíduos que atendem ao critério Classic*

```
1    select ?p ?d1 ?r1 ?d2 ?r2
2    where {
3       ?p a go:Patient;
4          d:hasDiagnosisDocument ?d1.
5       ?d1 a c:Sarcoma.
6       ?d1 a lfs:DiagsLess45.
7
8       ?p go:has1stDgRelative ?r1.
9       ?r1 d:hasDiagnosisDocument ?d2.
10      ?d2 a lfs:DiagsLess45.
11
12      ?p go:hasSome1stOr2ndRelatives ?r2.
13      ?r2 a lfs:AnyCancerBfr45OrSarcomaAnyAge.
14
15      ?r1 owl:differentFrom ?r2.
16 }
```

6.4.1 Tempo de Inferência

O tempo de inferência foi tabulado separadamente para cada critério a fim de observarmos a contribuição de cada um no tempo total de classificação, conforme apresentado na figura 6.7.

Figura 6.7: *Gráfico com os tempos totais de inferência divididos por critério.*

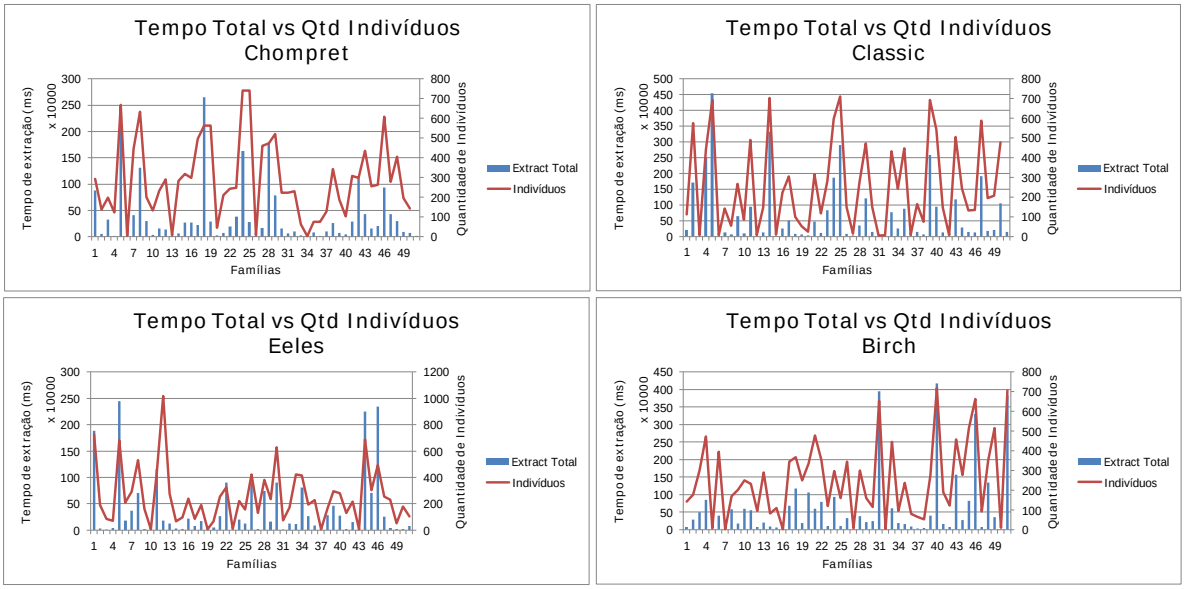


Também observamos a relação existente entre a quantidade de indivíduos em cada família e

[9] http://clark-parsia.trac.cvsdude.com/pellet-devel/ticket/420

o tempo de inferência para cada critério. Esse resultado pode ser observado na Figura 6.8. Nela, apresentamos a variação da quantidade de indivíduos (linha horizontal) para cada uma das famílias (eixo horizontal) de um determinado critério e o respectivo tempo total de inferência para aquela família (barra vertical).

Figura 6.8: *Tempos totais de inferência em relação à quantidade de indivíduos.*



Um detalhamento maior do tempo gasto em cada uma das 6 etapas da ferramenta `Directed-Extract-LiFraumeni` também foi plotado no Gráfico 9.4 (Capítulo 9). Os tempos calculados nesse gráfico representam a soma dos tempos gastos nas duas fases do processo (materialização - 1ª e 2ª etapas e classificação - 3ª e 4ª etapas).

6.4.2 Classificação das famílias

Para medir o grau de precisão, sensibilidade e acurácia da classificação, apresentamos uma série de Matrizes de Confusão (Tabela 6.3) para cada critério analisado a fim de comparar os resultados esperados com aqueles atingidos pelo algoritmo. Adicionalmente, apresentamos uma Matriz de Confusão considerando o total das amostras em conjunto (Tabela 7.1a). Esses resultados servirão de base para a comprovação da eficiência do algoritmo usado para classificação dos pacientes, segundo os critérios de Acurácia, Precisão, Sensibilidade e Prevalência. Também, um gráfico comparativo resumido das taxas de Acurácia, Prevalência, Sensibilidade e Precisão é apresentado na Seção 9.1 do Capítulo 9 (Figura 9.1).

Para cada Tabela de Confusão, foram calculadas, também, as taxas de Precisão (*precision*), Sensibilidade (*sensitivity*), também conhecida por TPR (*true positive rate*) ou *recall*, Acurácia (*accuracy*) e Prevalência (*prevalence*). A **Precisão** indica a taxa de acertos para os casos positivos (dentre os casos classificados como positivos, quantos são realmente positivos); a **Sensibilidade** indica a taxa de verdadeiros-positivos em relação ao conjunto de famílias realmente positivas (dentre as famílias que são realmente positivas, quantas o algoritmo acertou); a **Acurácia** indica quanto o

Tabela 6.3: *Matrizes de Confusão para os critérios Eeles, Birch, Classic e Chompret*

(a) *Matriz de Confusão para o critério Eeles*

		Esperado	
	Casos: 51	Positivo	Negativo
Alcançado	Positivo	43	0
	Negativo	1	7

Acurácia	98,04%	***TPR***	97,73%
Prevalência	86,27%	***FPR***	0%
Precisão	100,00%	***F-Measure***	0,988
Sensibilidade	97,73%		

(b) *Matriz de Confusão para o critério Birch*

		Esperado	
	Casos: 51	Positivo	Negativo
Alcançado	Positivo	41	0
	Negativo	0	10

Acurácia	100,00%	***TPR***	100%
Prevalência	80,39%	***FPR***	0%
Precisão	100,00%	***F-Measure***	1
Sensibilidade	100,00%		

(c) *Matriz de Confusão para o critério Classic*

		Esperado	
	Casos: 51	Positivo	Negativo
Alcançado	Positivo	44	0
	Negativo	0	7

Acurácia	100,00%	***TPR***	100%
Prevalência	86,27%	***FPR***	0%
Precisão	100,00%	***F-Measure***	1
Sensibilidade	100,00%		

(d) *Matriz de Confusão para o critério Chompret*

		Esperado	
	Casos: 51	Positivo	Negativo
Alcançado	Positivo	36	0
	Negativo	1	14

Acurácia	98,04%	***TPR***	97,30%
Prevalência	72,55%	***FPR***	0%
Precisão	100,00%	***F-Measure***	98,63%
Sensibilidade	97,30%		

Tabela 6.4: *Matriz de Confusão considerando todas as 204 Famílias de Teste conjuntamente.*

		Esperado	
	Casos: 204	Positivo	Negativo
Alcançado	Positivo	164	0
	Negativo	2	38

Acurácia	99,02%	***TPR***	98,80%
Prevalência	81,37%	***FPR***	0%
Precisão	100,00%	***F-Measure***	0,9939
Sensibilidade	98,80%		

classificador está correto (quantas famílias foram classificadas corretamente, independentemente de serem positivas ou negativas) e; **Prevalência** indica a proporção de casos positivos em relação ao número total de famílias. Além dessas, outras variáveis úteis foram calculadas para que a eficiência da classificação pelo motor de inferência fosse medido, como as **Taxas de Verdadeiro Positivo (TPR)**, **Taxas de Falso Positivo (FPR)** e a média ponderada entre a precisão e a TPR, chamada de ***F-Measure***. O FPR mede a taxa de erro para falsos positivos, ou seja, quando um caso deveria ser classificado como negativo, mas foi classificado como positivo. O valor *F-Measure*, calculado por meio da média ponderada entre a precisão e a TPR (*recall*), representa uma medida de desempenho do algoritmo para uma determinada classe, em uma análise estatística de classificação binária (Fórmula 6.1). Bons classificadores possuem valores *F-Measure* próximos de 1.

$$\text{F-Measure} = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{6.1}$$

Por fim, para avaliar o grau de influência da quantidade de indivíduos em cada família e o tempo gasto com a inferência e a sua classificação, construímos um gráfico de dispersão para apresentar o resultado do teste de correlação de *Pearson*. Optamos por utilizar este teste porque ele representa o grau de correlação entre variáveis de comportamento linear. A Figura 6.9 apresenta os resultados alcançados separadamente em cada critério.

Considerando a amostra conjuntamente, ou seja, as 204 famílias sem a separação por critérios *Li-Fraumeni*, obtivemos o gráfico de dispersão com o teste de *Pearson* na Figura 6.10.

Figura 6.9: *Gráficos de Dispersão para cada um dos quatro critérios Li-Fraumeni.*

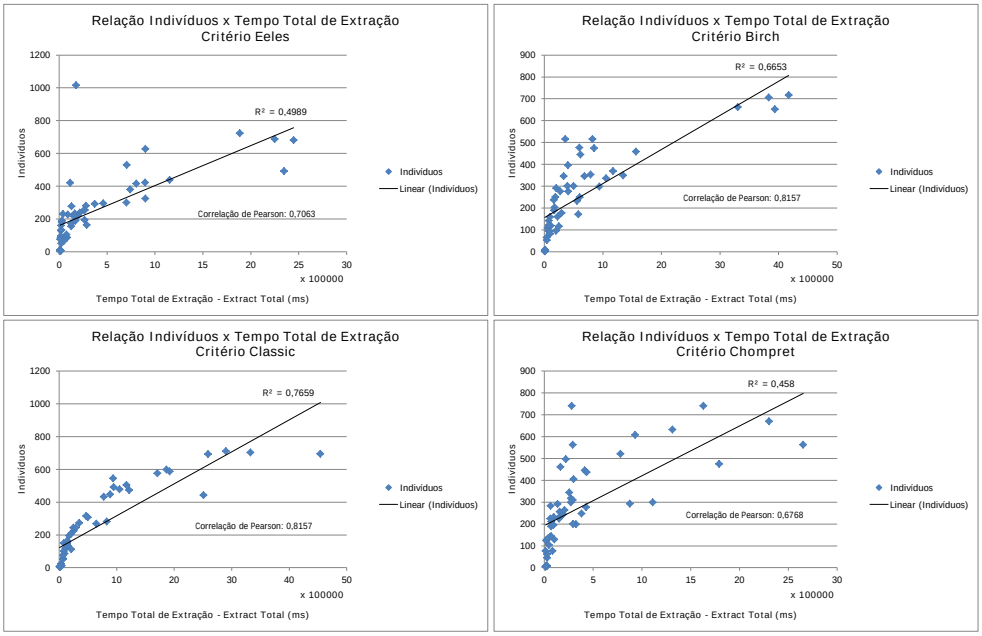
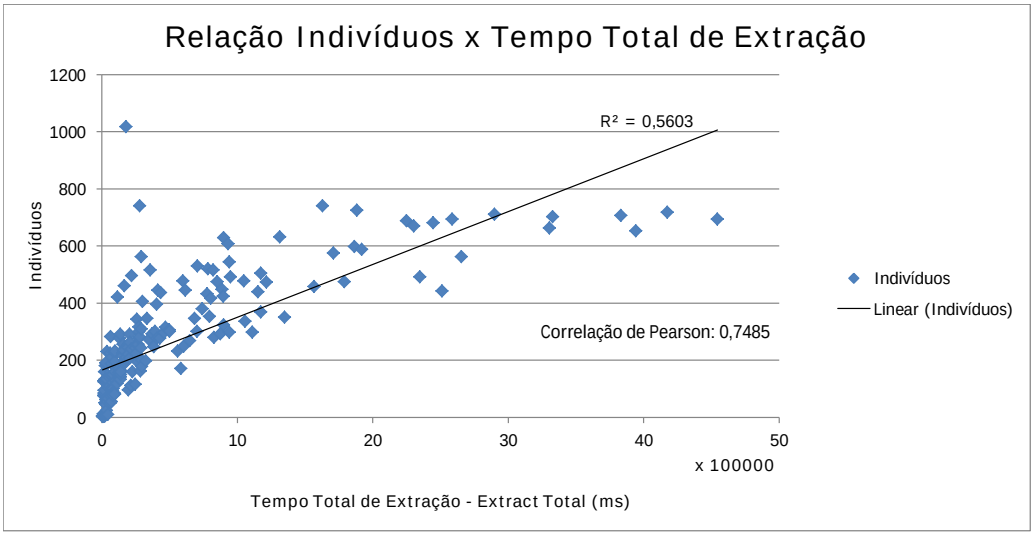


Figura 6.10: *Gráfico de Dispersão considerando todos os quatro critérios Li-Fraumeni.*



Em seguida, apresentaremos os resultados obtidos diretamente dos dados reais de pacientes do *A.C. Camargo Cancer Center*.

6.5 Resultado para o conjunto de casos reais

O processo completo de inferência levou, aproximadamente, 142 horas para concluir e seguiu o mesmo fluxo utilizado nos casos de teste, usando a inferência em duas etapas (Seção 6.4). Foram submetidos ao motor de inferência 172 arquivos, cada um representando uma família. Desses 172 arquivos, 10 apresentaram problemas de processamento (serão discutidos posteriormente na Seção 7.2) sendo 4 deles por inconsistência e 6 por estouro no limite de tempo de processamento (*timeout*),

estipulado, arbitrariamente, em 15 horas. Com isso, apenas 162 arquivos de famílias produziram, efetivamente, algum resultado significativo, que foram divididos em 3 categorias: Tempo de Inferência, Classificação das famílias e Consumo de Memória. Os resultados serão, posteriormente, discutidos e comparados àqueles obtidos com os casos de teste. A Tabela 6.5 lista as famílias utilizadas e a quantidade de indivíduos. O sinal (+) indica que uma família atende a algum dos quatro critérios *Li-Fraumeni*. Essa classificação foi realizada previamente pela equipe do Departamento de Oncogenética do *A.C. Camargo Cancer Center*.

Tabela 6.5: *Tabela resumo dos casos reais de famílias Li-Fraumeni do A.C. Camargo Cancer Center.*

Arquivo de Família	Qtd Indiv	
FAM00999	17	+
FAM01001	39	+
FAM01003	92	+
FAM01004	37	+
FAM01005	23	+
FAM01006	14	+
FAM01007	41	+
FAM01008	142	+
FAM01009	24	+
FAM01010	28	+
FAM01011	37	+
FAM01013	25	+
FAM01014	24	+
FAM01017	19	+
FAM01019	24	+
FAM01020	35	+
FAM01021	28	+
FAM01023	35	+
FAM01024	28	+
FAM01025	46	+
FAM01026	16	+
FAM01027	24	+
FAM01029	103	+
FAM01030	28	+
FAM01031	17	+
FAM01032	25	+
FAM01033	30	+
FAM01035	50	+
FAM01036	41	+
FAM01037	19	+
FAM01038	34	+
FAM01039	32	+
FAM01041	66	+
FAM01042	25	+
FAM01043	59	+
FAM01044	52	+
FAM01045	31	+
FAM01046	147	+
FAM01047	70	+
FAM01048	26	+
FAM01049	42	+
FAM01050	26	+
FAM01051	49	+
FAM01053	28	+
FAM01055	74	+
FAM01056	9	+
FAM01057	23	
FAM01058	23	+
FAM01060	27	+
FAM01064	46	+
FAM01065	31	+
FAM01067	26	+
FAM01068	12	+
FAM01069	42	+
FAM01070	27	+
FAM01071	40	+
FAM01072	42	+
FAM01074	11	+
FAM01075	56	+
FAM01076	31	+
FAM01078	37	+
FAM01081	66	+
FAM01083	127	+
FAM01084	25	+
FAM01085	15	+
FAM01086	62	+
FAM01087	14	+
FAM01088	29	+
FAM01090	10	+
FAM01091	39	+
FAM01092	75	+
FAM01093	56	+
FAM01094	54	+
FAM01097	44	+
FAM01098	18	+
FAM01099	31	+
FAM01100	9	+
FAM01101	21	+
FAM01102	54	+
FAM01103	146	+
FAM01104	20	+
FAM01105	28	+
FAM01106	28	+
FAM01107	44	+
FAM01109	25	+
FAM01110	20	+
FAM01111	23	+
FAM01112	22	
FAM01113	40	+
FAM01114	50	+
FAM01116	32	+
FAM01119	26	+
FAM01120	34	+
FAM01122	31	+
FAM01123	20	+
FAM01126	53	+
FAM01127	51	+
FAM01129	32	+
FAM01131	81	+
FAM01132	35	+
FAM01136	20	+
FAM01137	72	+
FAM01138	39	+
FAM01139	26	+
FAM01140	60	+
FAM01141	37	+
FAM01142	35	+
FAM01143	22	+
FAM01144	61	+
FAM01146	212	+
FAM01147	16	+
FAM01148	24	+
FAM01149	35	+
FAM01150	12	+
FAM01151	24	+
FAM01152	89	+
FAM01153	28	+
FAM01155	12	+
FAM01156	46	+
FAM01157	37	+
FAM01158	21	+
FAM01159	31	+
FAM01160	36	+
FAM01162	35	+
FAM01163	57	+
FAM01164	30	+
FAM01165	25	
FAM01166	29	+
FAM01169	44	+
FAM01172	24	+
FAM01173	25	+
FAM01174	36	+
FAM01175	41	+
FAM01177	42	+
FAM01178	22	+
FAM01179	77	+
FAM01180	20	+
FAM01181	28	+
FAM01182	23	+
FAM01183	25	+
FAM01184	27	+
FAM01186	16	+
FAM01187	50	+
FAM01188	34	+
FAM01189	22	+
FAM01190	37	+
FAM01191	14	+
FAM01193	20	
FAM01194	33	+
FAM01195	54	+
FAM01197	19	+
FAM01199	27	+
FAM01200	19	+
FAM01201	18	+
FAM01203	30	+
FAM01204	36	+
FAM01205	38	+
FAM01206	38	+
FAM01209	31	+
FAM01210	67	+
FAM01211	19	+
FAM01212	15	+

O sinal +, ao lado de cada família, indica que a mesma foi classificada pelo Departamento de Oncogenética do A.C. Camargo Cancer Center e atende a, pelo menos, um dos critérios Li-Fraumeni.

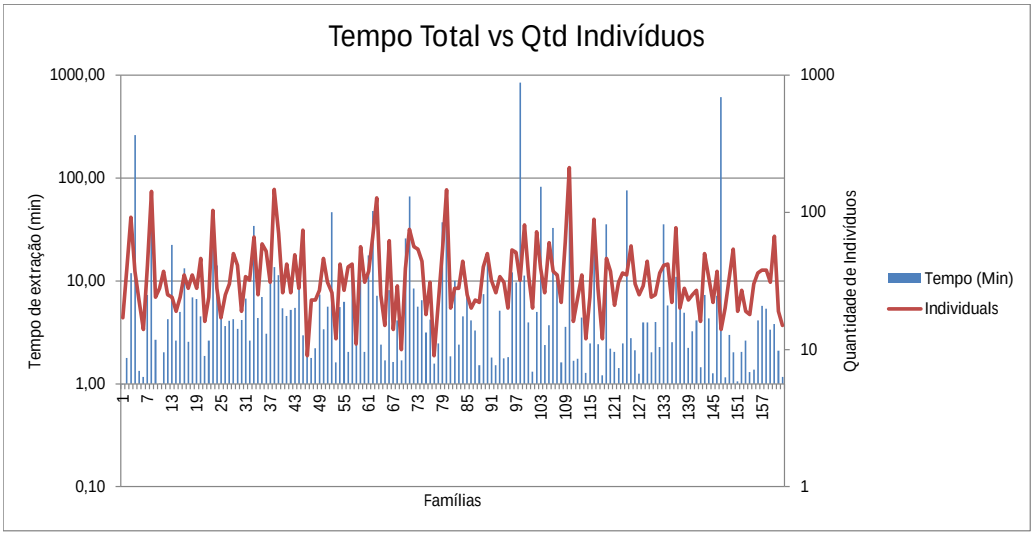
6.5.1 Tempo de Inferência

Observamos a relação existente entre a quantidade de indivíduos em cada família e o tempo de inferência para cada critério, tal qual foram analisados nos casos de teste. O resultado pode ser observado na Tabela 9.2 (Capítulo 9) e no gráfico correspondente 6.11. Os eixos foram plotados

em escala logarítmica para melhor visualização. Nela, apresentamos a variação da quantidade de indivíduos (linha horizontal) para cada uma das famílias (eixo horizontal) e o respectivo tempo total de inferência (em para aquela família (barras verticais).

Uma primeira observação no gráfico nos remete à correlação forte entre o tempo de processamento e a quantidade de indivíduos encontrada nos casos de teste. Entretanto, a análise de correlação das variáveis Quantidade de Indivíduos e Tempo de Inferência mostra que, para os casos reais, essa relação deixa de ser forte e passa a ser quase inexistente (Gráfico 6.12). Discutiremos, posteriormente, alguns fatores que influenciaram esse resultado e quais outras variáveis influenciaram mais fortemente o tempo total de extração.

Figura 6.11: *Tempos totais de inferência em relação à quantidade de indivíduos.*



Apresentamos um detalhamento maior do tempo gasto em cada uma das 6 etapas da ferramenta `Directed-Extract-LiFraumeni` plotado no Gráfico 9.5 (Capítulo 9). Os tempos calculados nesse gráfico representam a soma dos tempos gastos nas duas fases do processo (materialização - 1ª e 2ª etapas e classificação - 3ª e 4ª etapas).

6.5.2 Classificação das famílias

Usamos a Matriz de Confusão para medir o grau de precisão, sensibilidade e acurácia da classificação das famílias (Tabela 7.1b) a fim de comparar os resultados esperados com aqueles atingidos pelo motor de inferência. Também apresentamos um gráfico comparativo resumido das taxas de Acurácia, Prevalência, Sensibilidade e Precisão na Seção 9.1 do Capítulo 9 (Figura 9.2).

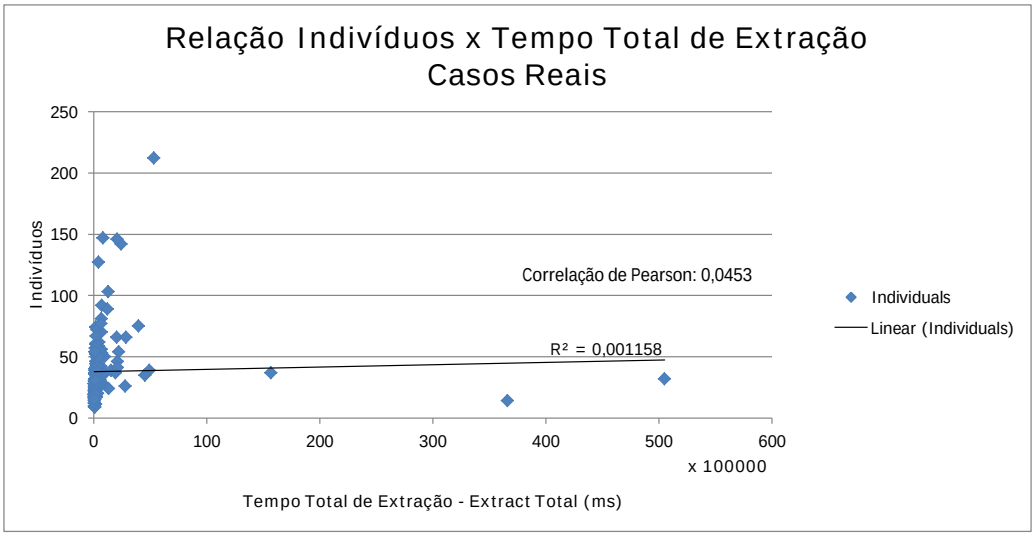
Tabela 6.6: *Matriz de Confusão considerando todas as 162 Famílias do A.C. Camargo Cancer Center.*

		Esperado	
	Casos: 162	Positivo	Negativo
Alcançado	Positivo	134	0
	Negativo	19	9

Acurácia	88,27%	***TPR***	88%
Prevalência	94,44%	***FPR***	0%
Precisão	100,00%	***F-Measure***	0,9338
Sensibilidade	87,58%		

Também foi construído o gráfico de dispersão correlacionando o tempo total de processamento e a quantidade de indivíduos em cada família, juntamente com o teste de correlação de *Pearson*. Ao contrário do que aconteceu nos casos de teste, o gráfico de dispersão para as famílias reais mostrou um resultado diferente, com uma fraca relação entre essas duas variáveis. Os dados de dispersão são apresentados no Gráfico 6.12.

Figura 6.12: *Gráfico de Dispersão das famílias Li-Fraumeni do A.C. Camargo Cancer Center.*



6.5.3 Consumo de Memória

A medição do consumo de memória aconteceu, separadamente, nas duas fases de cada rodada: Materialização e Classificação. Cada etapa é executada independentemente da outra, ou seja, ocupam porções de memória separadas, pois cada fase representa um processo diferente. Quando a primeira fase conclui, a memória utilizada pelo processo de Materialização é desocupada e liberada para a segunda fase, de Classificação. Dessa forma, o total de memória ocupado pela classificação de cada família representa o maior valor alcançado por uma das duas fases. O resultado do consumo de memória durante toda a etapa de classificação está representado no Gráfico 6.13.

Investigamos o consumo de memória da segunda fase em relação ao consumo de memória da primeira etapa, a fim de verificar se houve algum prejuízo ou ganho no consumo de memória com a abordagem de duas fases. O Gráfico 6.14 apresenta os dados comparativos por fases em cada família. O eixo vertical representa cada uma das famílias e o eixo horizontal, o consumo em cada etapa.

Devido à quantidade de famílias analisadas, o Gráfico 6.14 não permite uma comparação mais detalhada dos valores em cada etapa. Apresentamos, então, na Seção 9.2 a Tabela 9.3 que apresenta os números do uso de memória em cada etapa, para cada família. Apresentamos, também, uma relação, expressa em forma de percentual, entre o consumo de memória da segunda etapa em relação ao consumo da primeira etapa. Por fim, relacionamos o consumo de memória de cada etapa à quantidade de indivíduos em cada família *Li-Fraumeni*, conforme está representado no Gráfico 6.15.

No próximo capítulo, apresentaremos uma análise mais detalhada dos dados coletados durante

Figura 6.13: *Gráfico de Consumo de Memória Geral das famílias Li-Fraumeni.*

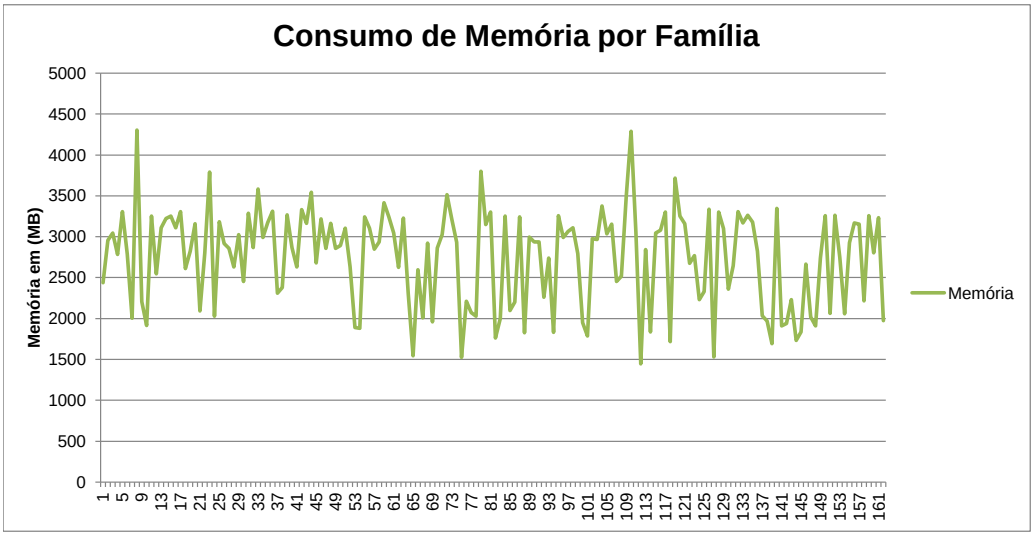
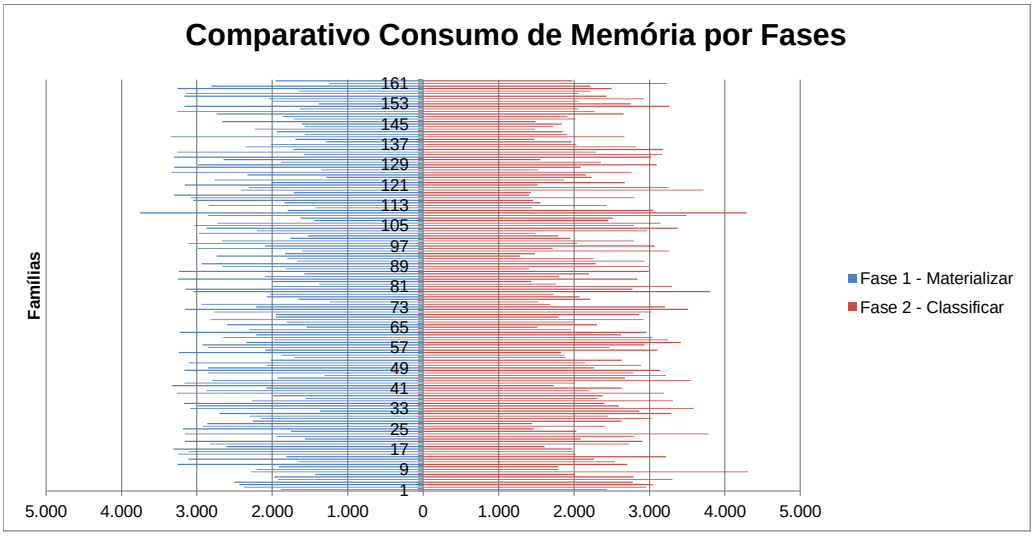
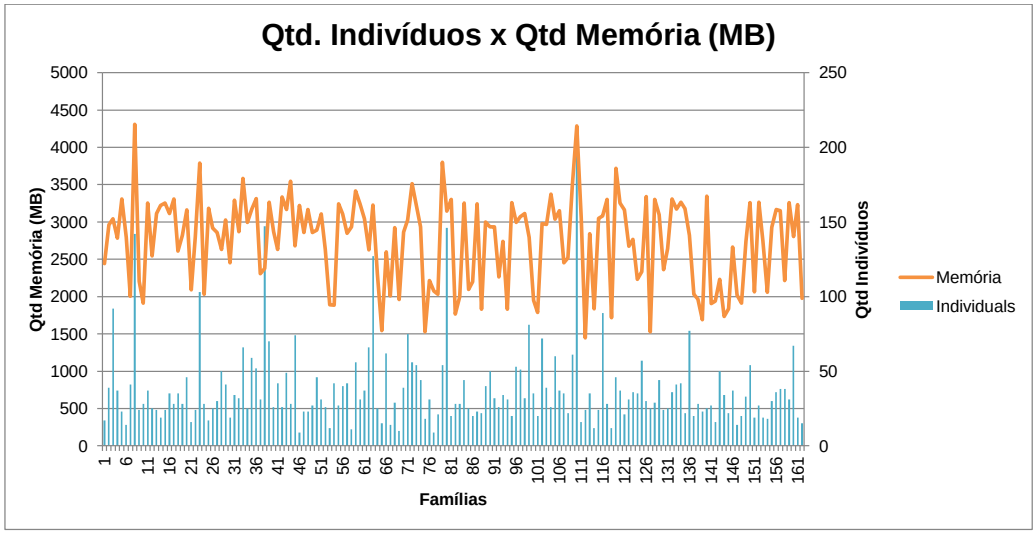


Figura 6.14: *Gráfico de Consumo de Memória Geral das famílias Li-Fraumeni.*



os testes e durante a execução usando os arquivos reais de famílias do *A.C. Camargo Cancer Center*. Lançaremos um olhar mais crítico, principalmente, nos resultados da classificação (Matriz de Confusão) e no gráficos de dispersão, que apontam indícios de como as etapas do processo de inferência influenciam no tempo final de processamento e no consumo de recursos computacionais. Analisaremos, também, os casos que apresentaram algum tipo de problema de processamento, restritos aos problemas de inconsistência.

Figura 6.15: *Gráfico Comparativo do consumo de memória versus quantidade de indivíduos Li-Fraumeni.*



Capítulo 7

Discussão

Apresentaremos, neste capítulo, uma discussão sobre os resultados alcançados e apresentados no Capítulo 6. A discussão está segmentada em duas partes: uma discussão sobre os resultados obtidos nos testes (Seção 7.1) e uma discussão sobre os resultados obtidos nas famílias reais do *A.C. Camargo Cancer Center* (Seção 7.2). Omitimos neste capítulo a repetição de gráficos e tabelas já apresentados no Capítulo 6 e, por isso, talvez seja necessário consultar os resultados apresentados no capítulo anterior.

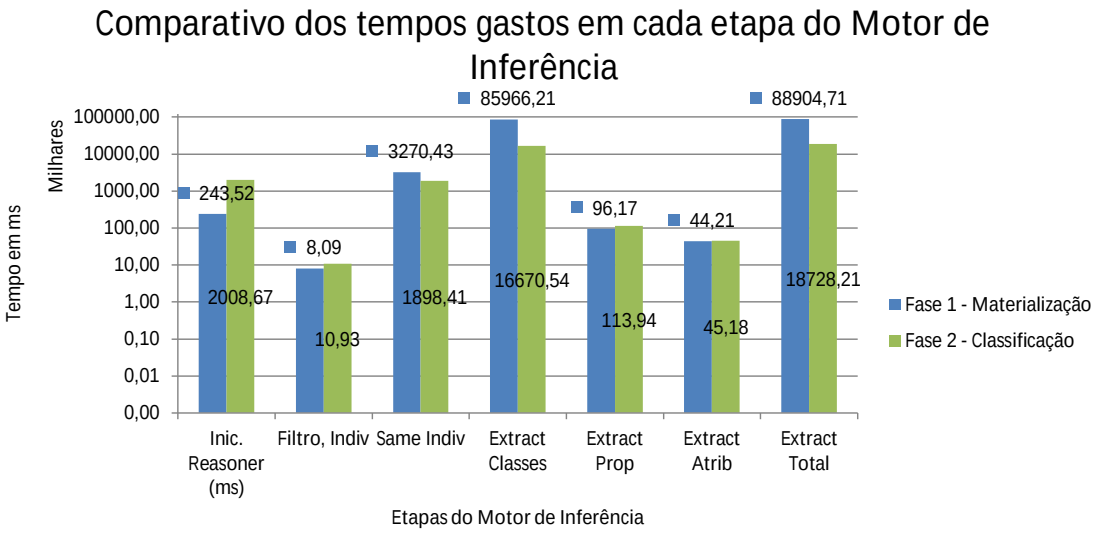
7.1 Discussão dos resultados para o conjunto de testes

Inicialmente, discutiremos sobre tempo gasto pela ferramenta `Directed-Extract-LiFraumeni` para materializar todas as inferências das famílias. Podemos afirmar, após análise dos gráficos de tempo, que a segunda inferência realizada sobre os axiomas materializados (segunda fase), juntamente com o `ABox` e o `TBox`, leva menos tempo para finalizar se comparado à primeira etapa. Apesar disso, o tempo gasto para inicializar o motor de inferência na segunda fase é maior do que na primeira fase, na maioria das vezes. Isso se deve à quantidade de axiomas presentes no arquivo de entrada (resultado da operação de *merge* entre os `ABox`, `TBox` e axiomas materializados da etapa anterior) e que são utilizados para realizar a inferência inicial no momento em que o motor de inferência é inicializado.

Observamos, também, que a etapa de classificação de indivíduos é mais rápida na segunda fase do que na primeira. A explicação para esse fato é que a maioria dos axiomas já foi inferido e materializado na primeira fase, restando, para a segunda, apenas aqueles que não conseguiram ser materializados na primeira fase (em razão do *bug* descrito anteriormente na Seção 6.2.1). Um panorama geral dos tempos comparativos está representado no gráfico da Figura 7.1, que apresenta, também, o tempo total gasto durante as duas fases (*Extract Total*). Essa indicação nos sugere que, mesmo que a segunda inferência tenha uma grande quantidade de axiomas, se as deduções feitas a partir deles já estiverem materializadas, o tempo de inferência será menor. O impacto, portanto, dessa abordagem na nossa solução ocasionou um acréscimo total no tempo de inferência de aproximadamente 5 horas (18728214,25 milissegundos). Entretanto, não é possível afirmar em quais proporções o tempo total de inferência seria mais rápido caso não houvesse necessidade da segunda fase de Materialização.

Uma analise baseada na correlação de *Pearson* nos indica forte relação (considerando a tabela de tamanho de efeito de Cohen [Coh88] - $r > 0.5$) entre os critérios Tempo de Inferência e Quanti-

Figura 7.1: *Gráfico dos tempos gastos em cada etapa da fase de Materialização comparativamente à fase de Classificação para os casos de testes. O gráfico encontra-se formatado em escala logarítmica para melhor visualização dos dados.*



dade de Indivíduos (Gráfico 6.10). É possível concluir, com isso, que o tempo gasto pelo motor de inferência nas duas fases sofre uma influência direta da quantidade de indivíduos em cada família (observar a linha referente ao número de indivíduos acompanhando a variação do tempo). É possível concluir, também, que a etapa mais lenta é a de classificação dos indivíduos por classes (*Extract Classes*) e que também é influenciada diretamente pela quantidade de indivíduos. O mesmo é válido para o tempo de inferência em cada um dos critérios (Figura 6.8), que apresentaram forte relação entre o Tempo de Inferência e a Quantidade de Indivíduos, segundo análise de correlação de *Pearson* individualmente para cada critério (Figura 6.9).

Considerando o tempo separadamente para cada critério, observamos que a diferença entre o critério que consumiu mais tempo (*Classic*) e o que consumiu menor tempo (*Chompret*) foi de três horas. Apesar do critério Chompret possuir regras de classificação maiores do que as dos outros critérios (maior quantidade de regras e regras com mais termos), o que a diferencia das demais é o fato da mesma não ter que selecionar mais do que um parente, em qualquer grau, do probando daquela família (Seção 5.3), dispensando, assim, o uso de termos `owl:differentFrom`. Ou seja, apenas as regras do critério Chompret não possuem átomos `owl:differentFrom` no corpo das regras. O impacto no uso desses átomos em regras SWRL recai no aumento da complexidade computacional (explosão na quantidade de axiomas `a owl:differentFrom b`), que causam um aumento no tempo de verificação de consistência. A ausência de termos `owl:differentFrom` nas regras Chompret, portanto, explicam seu melhor tempo em relação ao desempenho dos demais critérios.

Quanto aos resultados da classificação, a Tabela 6.3a, referente ao critério Eeles, nos mostra que foram geradas 44 famílias portadoras da síndrome e 7 famílias sem a presença da síndrome. A análise dos resultados detectou a presença de um erro na modelagem das regras de classificação Eeles, indicada pela ocorrência de um falso-negativo. Após a correção do erro na regra, a mesma família foi submetida ao classificador, que identificou corretamente a presença da síndrome segundo

os critérios Eeles. Ainda sobre o critério Eeles, apesar de termos obtido uma Precisão de 100%, o motor de inferência não foi 100% correto (Acurácia) em razão do caso falso-positivo. A alta taxa de Sensibilidade indica que o motor de inferência acertou na classificação de grande parte dos casos efetivamente positivos (deixando de fora apenas a família classificada como falso-negativo).

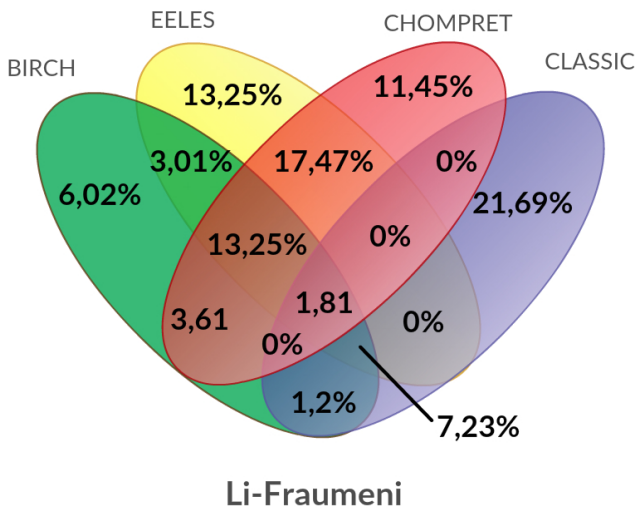
Para os critérios Classic e Birch (Tabelas 6.3c e 6.3b), o motor de inferência obteve os mesmos resultados de Precisão, Acurácia e Sensibilidade. Essas taxas mostram que o motor de inferência teve um bom desempenho ao classificar as famílias corretamente.

Por fim, o critério Chompret (Tabela Tabelas 6.3d) também apresentou um caso falso-negativo, tal qual o critério Eeles. Entretanto, as razões que levaram a isso divergiram do critério anterior. Nesse caso, um erro de processamento causado por uma falha de memória causou um término precipitado da primeira fase do teste. Com isso, o arquivo usado na segunda fase de extração não tinha todos os axiomas materializados e, assim, não conseguiu classificar os indivíduos daquela família corretamente. Para provar, alocamos mais memória para a Máquina Virtual Java e o arquivo da referida família foi submetido novamente ao teste, obtendo, finalmente, o resultado esperado. Um gráfico comparativo resumido das taxas obtidas pode ser observado na Figura 9.1.

Com relação à ocorrência de famílias que atendam a mais de um critério, fizemos uma análise do percentual por critério (Figura 9.3, disponível na Seção 9.1), e outra baseada no conjunto das amostras de teste (Figura 7.2). O resultado nos mostra, empiricamente, que apenas os critérios Birch e Chompret possuem famílias que podem ser classificadas, também, em todos os demais critérios. Nos demais critérios, nas famílias que foram classificadas como Eeles, nenhuma delas também foi classificada com Classic. No caso das famílias Classic, nenhuma delas foi classificada como Chompret. Esse resultado nos sugere que estes sejam critérios disjuntos, em que famílias que atendam ao critério Classic não podem, simultaneamente, atender ao critério Chompret (o mesmo vale, analogamente, para o critério de Eeles). Concluímos, da observação dos gráficos 7.2 e 9.3a, que, aproximadamente 47% das famílias que atenderam aos critérios de Chompret também atenderam aos critérios de Eeles. Já no gráfico 9.3c, de maneira geral, aproximadamente 65% das famílias que atendiam aos critérios Birch também atendem aos critérios Eeles e aproximadamente 53% das famílias que atendiam aos critérios Birch também atendiam aos critérios Chompret. Por fim, observamos que, aproximadamente, metade das famílias que atendiam aos critérios Eeles também atendiam aos critérios Chompret. Vale observar que essas são observações empíricas resultantes das famílias geradas aleatoriamente. Não procedemos com uma análise causa *versus* consequência dessas intersecções (quais os critérios que levaram a uma família ser classificada em dois critérios simultaneamente) por considerarmos esta como uma análise direcionada aos especialistas de domínio e que não se encontra nos objetivos deste trabalho.

Ao analisarmos as amostras conjuntamente, observamos que o maior percentual de famílias que atendem, ao mesmo tempo, dois ou mais critérios *Li-Fraumeni* é pouco mais de 17% (29 famílias), considerando um total de 166 famílias classificadas em pelo menos um dos critérios. Esse valor corresponde às famílias que foram classificadas como atendendo aos critérios Chompret e Eeles, simultaneamente. Apesar do baixo percentual de famílias que atendem a esses critérios simultaneamente, essa correlação sugere uma investigação mais aprofundada pelo Departamento de Oncogenética do *A.C. Camargo Cancer Center*, levando-se em consideração os tipos de tumores envolvidos nos diagnósticos, idade dos pacientes à época dos diagnósticos e graus de parentesco dos familiares acometidos por tumores.

Figura 7.2: *Diagrama de Venn mostrando os percentuais de classificação dos casos de teste para cada um dos quatro critérios da Síndrome de Li-Fraumeni.*



Por fim, uma análise dos resultados das classificações mostrou que a ferramenta pode ser utilizada em casos reais com um alto grau de confiabilidade, pois possui uma alta taxa de Acurácia e Precisão (Tabela 7.1a). Apesar do conjunto amostral utilizado durante essa etapa de testes estar desbalanceado (POS = 81,37%; NEG = 18,63%), a classe de amostras majoritárias foi aquela cujo teste pretendia avaliar (classe das famílias positivas), aumentando, com isso, a confiabilidade nas classificações das famílias positivas para a Síndrome de *Li-Fraumeni*. Além disso, a taxa de erro (*Error Rate*), conhecida também por *Misclassification Rate* e calculada por meio da equação 7.1, possui um valor baixo (0,98%).

$$ER = \frac{FP + FN}{total} \tag{7.1}$$

, em que ER = *Error Rate*, FP = Falso Positivo e FN = Falso Negativo.

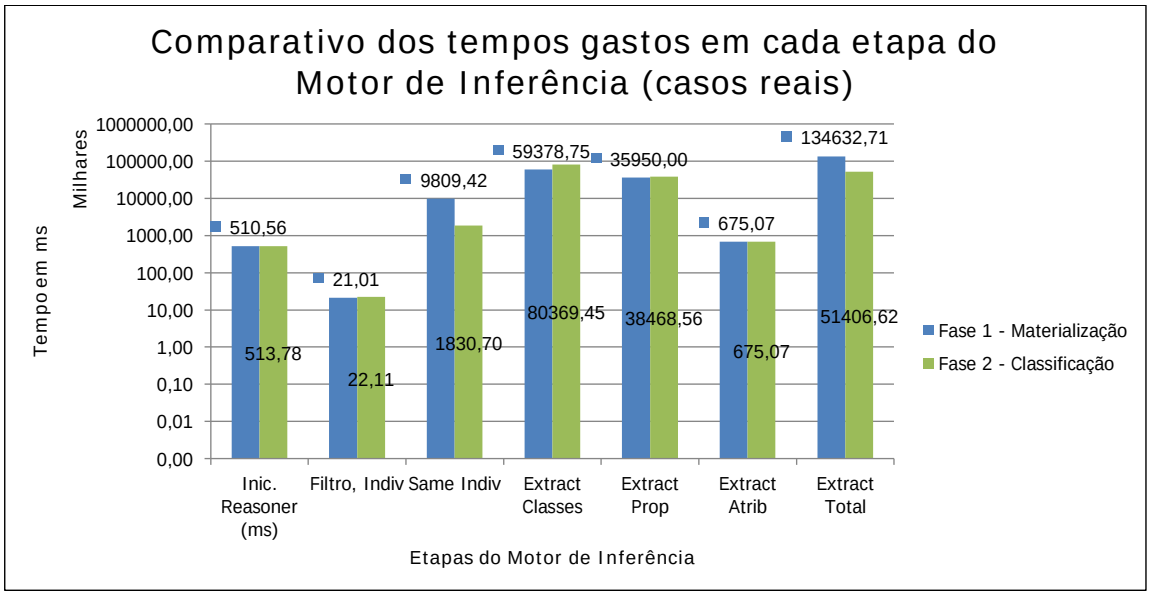
Diante dos resultados promissores que a base de testes nos proporcionou, procedemos com a classificação nos casos reais e os resultados serão discutidos na próxima seção.

7.2 Discussão dos resultados para o conjunto de casos reais

Observamos que o processamento das 172 famílias, apesar de levar muito mais tempo para concluir, manteve a relação de tempo entre a primeira e a segunda fase, observado durante os testes, ou seja, a fase de Classificação foi mais rápida para concluir do que a fase de Materialização. As razões continuam sendo as mesmas observadas nos casos de teste: a maioria dos axiomas a serem inferidos na segunda fase já foram materializados na primeira fase, proporcionando uma economia de tempo. Apesar disso, o tempo de inicialização do motor de inferência continua mais alto na segunda fase, também em razão da quantidade de axiomas que é necessário carregar na memória (maioria dos axiomas já materializados na primeira fase).

Apesar disso, não deixamos de notar que o ganho de tempo na segunda fase foi menor no conjunto de famílias reais (61,8%) se comparado ao ganho de tempo da segunda fase nos arquivos de testes (78,9%) (ver Gráfico 7.1). Essa diminuição é fruto da etapa de extração dos indivíduos

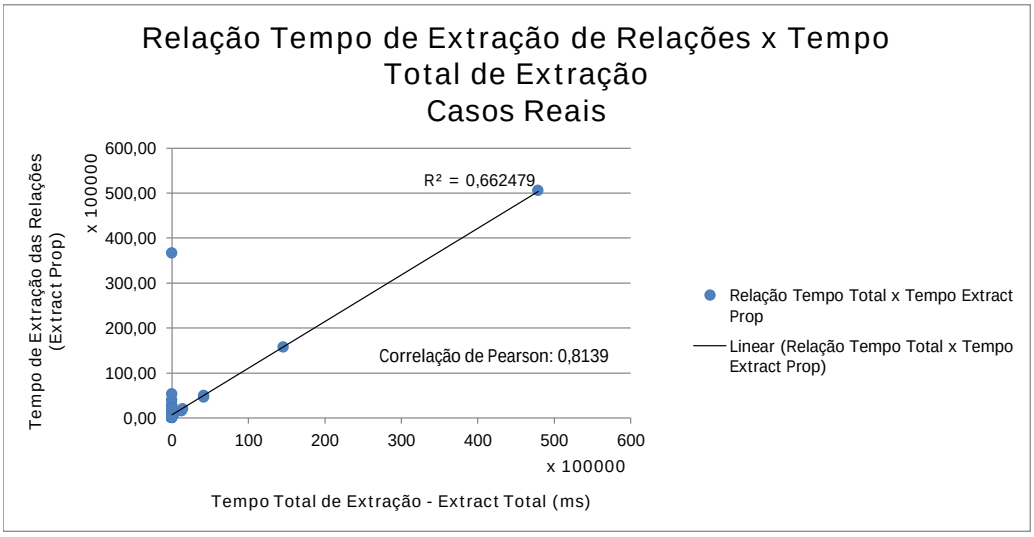
Figura 7.3: *Gráfico dos tempos gastos em cada etapa da fase de Materialização comparativamente à fase de Classificação. O gráfico encontra-se formatado em escala logarítmica para melhor visualização dos dados.*



por classe (*Extract Classes*) ter sido maior na segunda fase (durante os testes, a segunda fase levou menos tempo para concluir a etapa *Extract Classes*), possivelmente em razão de mais axiomas não terem sido materializados durante a primeira fase o que acabou acarretando um tempo maior para extração.

Diferentemente do ocorrido nos casos de testes, entretanto, uma análise da correlação de *Pearson* indicou uma fraca relação entre a quantidade de indivíduos e o tempo de inferência (Gráfico 6.12 para $r < 0.10$). Ao avaliarmos outras variáveis, descobrimos que o Tempo Total de Inferência sofreu maior influência do tempo de extração das relações (*Extract Prop*), conforme pode ser observado no Gráfico 7.4.

Figura 7.4: *Gráfico de Dispersão das famílias Li-Fraumeni do A.C. Camargo Cancer Center em relação ao tempo de Extração das Relações (Extract Prop).*



Essa disparidade ocorreu em razão da diferença na quantidade de propriedades entre os arquivos do caso de teste e os arquivos de famílias reais. Nos arquivos de teste, havia apenas o mínimo de relações para que o motor de inferência identificasse e classificasse o tipo de síndrome. Assim, cada família de teste possuía apenas algumas poucas relações `hasDocument` (3 no máximo, uma para cada documento), `hasDiagnosticCode` (também 3 no máximo, um código para cada diagnóstico) e as relações de parentesco entre os familiares (que dependiam da quantidade de indivíduos). Nas famílias reais, a quantidade de relações `hasDocument` variou de 5 relações (família FAM01100) até 517 relações (família FAM01124), cada uma dessas relações possuía uma relação `hasDiagnosticCode`, pelo menos. Considerando que nas famílias reais apareceram axiomas do tipo `SameAs`, igualando indivíduos e aumentando a quantidade de axiomas `hasDocument` a materializar, então, a extração de relações entre indivíduos `Patient` e indivíduos `Document` foi consideravelmente mais trabalhosa nos casos reais do que nos casos de teste (fato esse que pode ser observado no Gráfico 7.3.

O resultado final da classificação das famílias nos permite concluir que o motor de inferência teve o mesmo rendimento nos arquivos de famílias do *A.C. Camargo Cancer Center* e nos casos de teste. Comparando as duas Matrizes de Confusão (Figura 7.1), percebemos que, em ambos os casos, o motor de inferência obteve Precisão de 100%, indicando que não houve nenhuma família classificada, erradamente, como positiva (FPR). Entretanto, como a taxa de falsos-negativos foi maior nos casos reais do que nos casos de teste, a Acurácia caiu para menos de 90%, mas ainda representa um valor consideravelmente alto, indicando uma boa confiabilidade no uso para classificação de famílias *Li-Fraumeni* (também confirmado pela Sensibilidade dos testes).

Tabela 7.1: *Comparação das duas Matrizes de Confusão: Casos de Teste e Famílias Li-Fraumeni do A.C. Camargo Cancer Center*

(a) *Matriz de Confusão considerando as 204 Famílias de teste.*

		Esperado	
	Casos: 204	Positivo	Negativo
Alcançado	Positivo	164	0
	Negativo	2	38

Acurácia	99,02%	***TPR***	98,80%
Prevalência	81,37%	***FPR***	0%
Precisão	100,00%	***F-Measure***	0,9939
Sensibilidade	98,80%		

(b) *Matriz de Confusão considerando as 162 Famílias Li-Fraumeni do A.C. Camargo Cancer Center.*

		Esperado	
	Casos: 162	Positivo	Negativo
Alcançado	Positivo	134	0
	Negativo	19	9

Acurácia	88,27%	***TPR***	88%
Prevalência	94,44%	***FPR***	0%
Precisão	100,00%	***F-Measure***	0,9338
Sensibilidade	87,58%		

Algumas famílias foram classificadas pelo Departamento de Oncogenética como pertencendo a um único critério, como é o caso da família representada no arquivo `FAM01107`. O mesmo fora classificado de acordo com o critério **Classic** apenas, mas a *LFOnto* também o classificou como atendendo aos critérios Birch, Eeles e Chompret. No computo geral, 2 famílias, classificadas pelo Departamento de Oncogenética como somente **Classic**, foram associadas também a outros critérios; 2 famílias **Birch** também foram associadas a outros critérios; 13 famílias **Eeles** também foram associadas a outros critérios e; 18 famílias **Chompret** também foram associadas a outros critérios. Esse resultado sugere uma maior investigação do Departamento de Oncogenética quanto ao grau de correlação entre os critérios pois, no caso do critério **Classic**, todas as famílias informadas foram associadas a algum outro critério da Síndrome de *Li-Fraumeni*. A Figura 7.5 apresenta um panorama da intersecção dos critérios após a classificação das famílias reais do *A.C. Camargo Cancer Center*.

Nela, podemos observar que nenhuma das famílias foram classificadas exclusivamente como **Classic** ou **Birch**, sugerindo, possivelmente, alguma correlação entre cada um dos critérios e os outros três.

Figura 7.5: *Diagrama de Venn mostrando os percentuais de classificação das famílias reais do A.C. Camargo Cancer Center para cada um dos quatro critérios da Síndrome de Li-Fraumeni.*

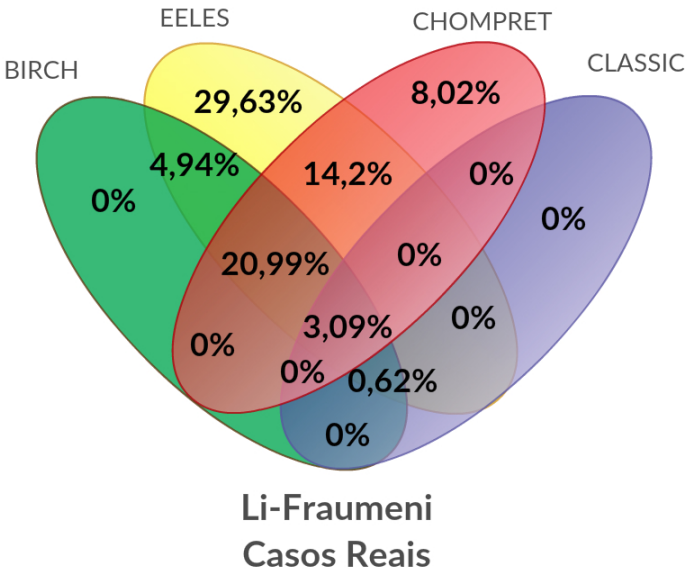
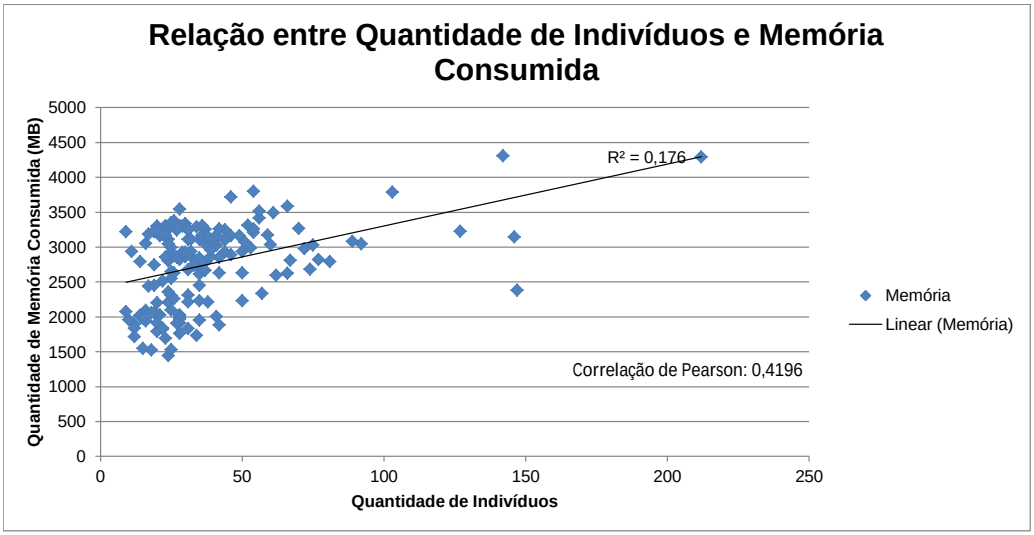


Figura 7.6: *Gráfico de Dispersão das famílias Li-Fraumeni do A.C. Camargo Cancer Center quanto ao número de indivíduos em relação ao consumo de memória.*



Por fim, o consumo de memória avaliado pela ferramenta ficou dentro de um nível aceitável, com extremos de 1,44GB e 4,30GB, para um limite de 10GB alocados no *heapspace* da Máquina Virtual Java (consumo de memória abaixo de 50%). Quanto ao consumo de memória por fases, notamos que a segunda fase (Classificação) teve menor consumo em 56% (91) das famílias classificadas, o que representa um valor sem significância para afirmarmos que a segunda fase tem melhor consumo que a primeira fase (ver o Gráfico 7.3). Também confirmamos a influência mediana da quantidade de indivíduos em cada família no consumo de memória por meio do grau de correlação de *Pearson*, conforme mostra o Gráfico 7.6. Isso ocorre porque o consumo de memória, em um motor de inferên-

cia, não sofre influência unicamente da quantidade de indivíduos presente no `ABox`, mas também da quantidade de relações `ObjectProperty`, dos axiomas do tipo `SameAs`, além de outros.

Os casos julgados como falso-negativos (19 famílias) foram investigados separadamente, a fim de encontrar falhas que pudessem justificar a classificação errônea. Inicialmente, encontramos 2 falhas conceituais na própria ontologia: *(i)* alguns conceitos sobre Neoplasias, oriundos da CID-10, foram desconsiderados na *CDOnto* (capítulo D00_D48 fora desconsiderado, inicialmente), resultando em não classificação de alguns diagnósticos, envolvidos na Síndrome de *Li-Fraumeni*, como sendo `Cancer_Diagnostic`. Com isso, alguns pacientes, que deveriam atender a alguns critérios *Li-Fraumeni*, não o fizeram e, poir isso, foram classificados como negativos; *(ii)* deixamos de considerar familiares de terceiro grau (`has3rdDgRelative`) de pacientes em um dos critérios Chompret. De acordo com a regra 1 do critério Chompret 2009 (ver Seção 3.2.2, Tabela 3.4), pacientes com diagnóstico de tumores pertencentes ao espectro *Li-Fraumeni* antes dos 46 anos e que possuem um parente em qualquer grau com múltiplos tumores primários em qualquer idade, atendem ao critério Chompret. Entretanto, a regra implementada na *LFOnto* considerava apenas parentes de primeiro e segundo graus, deixando de fora os de terceiro grau (Regras *SWRL* 7.1 e 7.2).

Algoritmo 7.1: *Listagem de regra SRWL para definir uma das condições do critério Chompret 2009.*

```
1    Chompret_Spectrum_Diagnosis(?d), DiagsLess46(?d),
2        Multiple_Tumors_Patients(?r), Patient(?p), hasDocument(?p, ?d),
3        hasSome1stOr2ndRelatives(?p, ?r) -> Chompret(?p)
```

Algoritmo 7.2: *Listagem de regra SRWL para definir uma das condições do critério Chompret 2009 considerando todos os graus de parentesco.*

```
1    Chompret_Spectrum_Diagnosis(?d), DiagsLess46(?d),
2        Multiple_Tumors_Patients(?r), Patient(?p), hasDocument(?p, ?d),
3        has3rdDgRelative(?p, ?r) -> Chompret(?p)
```

Após a correção das regras e da taxonomia da *CDOnto*, executamos novamente o motor de inferência nos arquivos de família que resultaram nos casos falso-negativos. Das 19 famílias desse grupo, 5 foram classificadas corretamente após as correções acima (verdadeiro-positivos). As 14 famílias falso-negativas restantes foram investigadas e o resultado está tabulado conforme a Tabela 7.3. Com esse novo resultado, a Matriz de Confusão apresentou novos valores, aumentando a Precisão, Acurácia e *F-Measure* e reduzindo a Taxa de Erro para 8,64% (Tabela 7.2).

Tabela 7.2: *Matriz de Confusão após correção da taxonomia na CDOnto e na LFOnto.*

		Esperado	
	Casos: 162	Positivo	Negativo
Alcançado	Positivo	139	0
	Negativo	14	9

Acurácia	91,36%	TPR	91%
Prevalência	94,44%	FPR	0%
Precisão	100,00%	F-Measure	0,9521
Sensibilidade	90,85%		

No que se refere aos dez casos que não finalizaram sua execução, quatro apresentaram inconsistências quando submetidas ao motor de inferência. A execução da ferramenta **`explain`**[1] em cada

[1] A ferramenta `explain`, disponível no pacote de ferramentas pellet, explica inferências que ocorrem em uma determinada ontologia, inclusive inconsistências. Disponível em: https://github.com/Complexible/pellet

Tabela 7.3: *Tabela com casos falso-negativos.*

Família	Critérios Esperados	Critérios Encontrados	MOTIVO FN
FAM00999	Chompret 2009	Nenhum	1
FAM01021	Classic	Nenhum	1 e 2
FAM01035	Chompret 2009	Nenhum	2
FAM01037	Classic e Chompret	Nenhum	1 e 2
FAM01038	Chompret 2009 e Eeles	Nenhum	2
FAM01056	Classic	Nenhum	1
FAM01060	Chompret 2009 e Eeles	Nenhum	2
FAM01116	Chompret 2009	Nenhum	2
FAM01140	Chompret 2009	Nenhum	2
FAM01148	Chompret 2009	Nenhum	2
FAM01149	Chompret 2009	Nenhum	2
FAM01184	Birch Chompret 2009	Nenhum	2
FAM01195	Eeles e Chompret 2009	Nenhum	2
FAM01204	Chompret 2009	Nenhum	2

Legenda
1 - Informações insuficientes na ontologia. Possivelmente algum dado que classificaria a família no critério esperado não estava presente. Sugere-se revisão nos registros originais para saber se o dado estava presente, mas não foi materializado ou se realmente não estava presente desde a coleta.

2 - Critério informado no campo `hasCriteriaDefinition` diverge daquele encontrado pelo motor de inferência. A classificação da família feita manualmente parece ter sido baseada em outros critérios que não aqueles usados pela LFOnto. Sugere-se revisão da classificação feita manualmente com os registros informados no próprio arquivo de família.

uma das famílias mostrou que todos os arquivos apresentaram axiomas que violaram uma regra existente na *CDOnto*: `Breast_Cancer disjoint Not_Breast_Cancer`. A violação acontece quando um diagnóstico recebe um código que o classificou inicialmente como `Sarcoma` e, simultaneamente, recebeu outro código que o classifica como `Breast_Cancer`. Entretanto, na *CDOnto*, `Sarcoma` é subclasse de `Not_Breast_Cancer`, causando, assim, a inconsistência. Um exemplo dessa análise pode ser verificada na saída do comando `explain` na Figura 7.7.

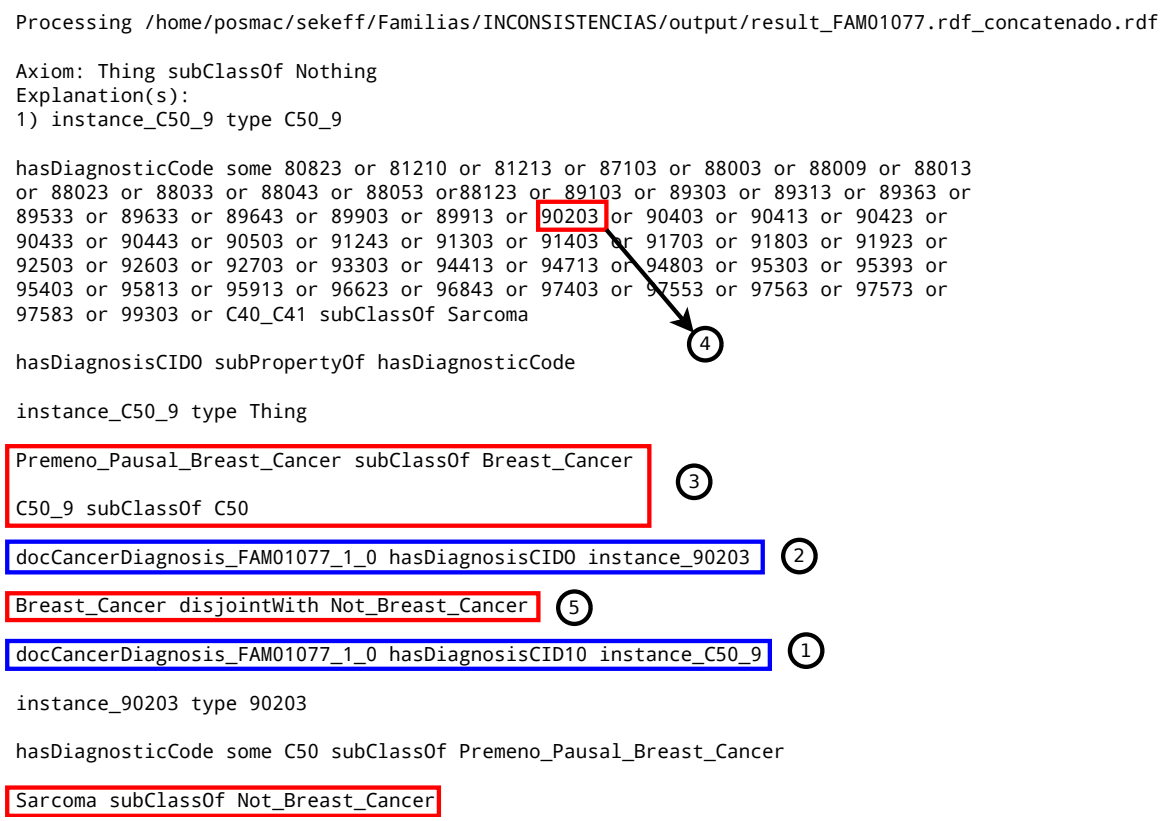
Dessa forma, repassamos a inconsistência ao Departamento de Oncogenética e ao Departamento de Informática Médica para averiguação se este é um erro conceitual, de inconsistência dos dados armazenados, uma falha na materialização dos dados para armazenamento na *triplestore* ou uma falha humana, ocorrida no momento da codificação do diagnóstico.

7.3 Considerações Finais

Apesar da *Li-Fraumeni Ontology* não ter conseguido identificar positivamente os critérios *Li-Fraumeni* em todas as famílias, avaliamos que a solução atendeu satisfatoriamente ao proposto inicialmente, que é se apresentar como um mecanismo de classificação automática de pacientes, segundo os critérios da Síndrome de *Li-Fraumeni*. A alta Acurácia e a baixa Taxa de Erro mostram que a mesma pode ser utiliza de maneira confiável pelo Departamento de Oncogenética do *A.C. Camargo Cancer Center*. A ferramenta também encontrou situações que sugerem uma análise mais detalhada pelo departamento, como famílias diagnosticadas como portadoras da mutação e que não foram classificadas pela ontologia e arquivos de famílias que apresentaram inconsistências quanto aos axiomas codificados na ontologia. Essas situações podem ter sido causadas por erros humanos, sujeira na base de dados ou algum procedimento que falhou no momento da transformação desses dados para a *triplestore*.

A metodologia desenvolvida e utilizada para a construção da *Li-Fraumeni Ontology* foi satisfató-

Figura 7.7: *Saída do comando* `explain`.



Na saída, observamos que o diagnóstico, tanto possui código 9020|3 quando C50.9 (1 e 2). Em seguida, os axiomas 3 e 4 estabelecem que todos os diagnósticos que possuem códigos C50.9 e 9020|3 são diagnósticos, respectivamente, de `Breast_Cancer` *e de* `Sarcoma`*. Por fim, o axioma 5 estabelece que diagnósticos de* `Sarcoma` *e de* `Breast_Cancer` *são disjuntos e, por isso, inconsistentes.*

ria e conseguiu cobrir todos os pontos que julgamos importantes durante o processo, principalmente no quesito **agilidade** e **colaboração**. A modularidade da ontologia exerceu um importante papel durante o desenvolvimento, pois permitiu o rastreio eficiente e a rápida correção de erros, tanto conceituais como sintáticos (regras *SWRL* e linguagem OWL). A decisão de usar a modularização por meio das cláusulas `owl:imports` deixou ainda mais flexível a modularização da *Li-Fraumeni Ontology*, pois, de outra forma, teríamos que manter, além dos módulos já desenvolvidos, as ontologias que representam o mapeamento entre os conceitos de módulos diferentes. O mapeamento dos conceitos da *Li-Fraumeni Ontology* para ULO's é possível, mas pode representar um aumento considerável no tempo levado pelo motor de inferência para extrair os axiomas e concluir o processo de inferência. A *Li-Fraumeni Ontology*, que possui aproximadamente 1800 classes, 1700 indivíduos e 17 relações, classificou uma família com uma média de 38 indivíduos em 20 minutos, em média, atingido picos de até 14 horas para uma família de 32 indivíduos com 34 relações. Ao mapearmos a *Li-Fraumeni Ontology* a uma ULO que, normalmente, possui muito mais do que 1800 classes (a *Gene Ontology* - Seção 2.2.6, por exemplo, possui mais de 44 mil classes e mais de 150 relações) e relações muito complexas, o tempo de extração de inferências poderia ficar impraticável, considerando o tempo de pior caso (2NEXPTIME [DCtTdK11]) para inferências usando o motor de inferência Pellet em ontologias de complexidade $\mathcal{SHROIQ}$. Como foi apresentado na Seção 2.2.4, as ULOs tem por finalidade oferecer uma camada de entendimento entre conceitos para domí-

nios diferentes, objetivando uniformizar o uso das ontologias envolvidas. Baseado nessa afirmação, apresentamos, na Seção 9.5, uma sugestão de mapeamento da *Li-Fraumeni Ontology* para a ULO BioTopLite [BSSH08, SB13] (ver Seção 2.2.6). Escolhemos a BioTopLite por ser uma versão mais leve da BioTop, com menos classes e relações (49 classes e 50 relações contra 390 classes e 82 relações da BioTop), e também por ser uma ULO do domínio específico das ciências biomédicas, além de ter suas origens fundamentadas na BFO (ver Seção 2.2.4), uma ULO madura que formaliza conceitos mais genéricos do mundo real.

Por fim, a abordagem modular da *Li-Fraumeni Ontology* permite que ela seja adaptada, sem muito esforço, para outras síndromes de natureza hereditária. Na Seção 9.4, apresentamos uma proposta de modelagem para a Síndrome de *Lynch* [KW04]. Escolhemos essa síndrome como exemplo porque a mesma é do domínio da Oncogenética e simples de ser modelada. Não aprofundaremos nas questões filosóficas da modelagem nem mostraremos resultados de nenhum dado inferido sobre essa síndrome. O objetivo é apenas mostrar a versatilidade e a modularidade da *Li-Fraumeni Ontology* quando adaptada para outras síndromes.

Capítulo 8

Conclusões

Apesar do muito que se tem feito no combate ao Câncer, a quantidade de pacientes acometidos por novos tumores aumenta a cada ano, sugerindo que ainda há muito o que se fazer nesse campo. A *Li-Fraumeni Ontology* se apresenta, nesse cenário, como uma proposta de ferramenta para auxiliar o médico na descoberta e no diagnóstico da Síndrome de *Li-Fraumeni*, uma síndrome de natureza hereditária que está ligada ao aparecimento incomum de vários tumores no mesmo pacientes.

A *Li-Fraumeni Ontology* é uma ontologia modular que, quando submetida a um motor de inferência, juntamente com um arquivo contendo o histórico familiar de câncer de uma determinada família (`ABox`), é capaz de classificá-la de acordo com os quatro critérios da Síndrome de *Li-Fraumeni*: *Classic*, Birch, Eeles e Chompret. Ela utiliza os dados dos pacientes que foram coletados anteriormente e foram armazenados em sistemas distintos no *A.C. Camargo Cancer Center*. Esses dados são, posteriormente, recuperados por uma ferramenta de integração de dados heterogêneos, desenvolvido pelo Departamento de Informática Médica, processados e, novamente, armazenados, agora em uma *triplestore*, alimentando, assim, a *Li-Fraumeni Ontology*. A fim de uniformizar os conceitos utilizados na *Li-Fraumeni Ontology*, propusemos um mapeamento com uma ontologia *upper-level* (BioTopLite), que serve para nortear o reuso da *Li-Fraumeni Ontology* em outros domínios do conhecimento. Apresentamos, também, uma proposta de modelagem de uma ontologia para a Síndrome de *Lynch*, que também possui características hereditárias e que reúne um conjunto de critérios clínicos que sugerem a presença, ou não, de uma mutação genética. Notamos que é possível o mapeamento da *Li-Fraumeni Ontology* com outras ULOs; entretanto, elas influenciam fortemente o tempo de inferência para extração novos conhecimentos, pois aumentam a complexidade da ontologia.

Quanto ao tempo total de processamento, notamos uma grande influência da quantidade de relações presentes em um `ABox` sobre o tempo de extração dos axiomas, mesmo em ontologias com grandes quantidades de indivíduos. Essa contribuição joga uma luz na discussão sobre a extração de axiomas por inferência em ontologias com grandes `ABox`. Ontologias com muitos indivíduos, em que apenas poucos destes participam de alguma relação, possuem baixo tempo de extração de axiomas por meio de inferência (ontologias dos casos de teste, com muitos indivíduos e poucas relações), enquanto ontologias com menos indivíduos, mas com uma grande quantidade deles participando de algum tipo de relação, possuem maior tempo de extração de axiomas (famílias dos casos reais, com menos indivíduos mas com muitas relações de parentesco e de diagnóstico). Também notamos que axiomas do tipo `SameAs` podem aumentar ainda mais o tempo de processamento, tornando a extração dos axiomas ainda mais lenta.

A metodologia utilizada durante a construção da *Li-Fraumeni Ontology* se mostrou ágil (desburocratizada) e colaborativa, registrando todas as versões criadas e permitindo a colaboração de novas versões/módulos desenvolvidos separadamente pelo Departamento de Informática Médica. A metodologia proposta, apesar de satisfatória, ainda não está madura e carece de diversos testes para que possa ser considerada, definitivamente, como uma metodologia ágil colaborativa. Ela encontra-se, ainda, no seu estágio inicial e nem pode ser equiparada às outras já maduras e bem estáveis, como a METHONTOLOGY, a RapidOWL e a UPON. Entretanto, acreditamos que novos testes podem fazê-la evoluir a um patamar estável, contribuindo mais solidamente para a construção colaborativa de ontologias.

Julgamos, assim, que a *Li-Fraumeni Ontology* teve um rendimento satisfatório no computo geral, principalmente quanto à cobertura do resultado (*Recall*) e a sua Precisão, classificando corretamente $99,02\%$ dos casos de teste e $88,27\%$ das famílias reais. O índice *f-measure* mostrou um equilíbrio entre as taxas de Precisão e de Cobertura, tanto nos casos de teste quanto nos casos reais (com valores, respectivamente, de $0,9939$ e $0,9338$). Esses valores representam que não só o motor de inferência acertou a classificação da maior parte dos casos avaliados (positivos e negativos) em relação à população geral como também acertou na classificação de todos os casos positivos encontrados em relação ao total de casos positivos.

Na área Biomédica, a contribuição deste trabalho é oferecer uma ferramenta importante na classificação de famílias segundo os critérios da Síndrome de *Li-Fraumeni* e na investigação de outros fatores que podem sugerir a presença da mutação do $TP53$. Os casos falso-negativos podem ser analisados pelo Departamento de Oncogenética a fim de encontrar características que levaram à divergência de classificação entre os critérios mapeados pela ontologia e aqueles utilizados pelo profissional que a classificou como um caso positivo.

Apesar da grande contribuição que este trabalho de pesquisa oferece no estudo e desenvolvimento de técnicas computacionais que auxiliem a extração de conhecimentos na área de câncer, reconhecemos que algumas lacunas foram deixadas em aberto durante o seu desenvolvimento, até mesmo por se tratar de uma versão em estágio inicial de maturação. Algumas dessas lacunas podem ser descritas conforme segue:

1. A metodologia proposta neste trabalho foi testada apenas para o domínio da Síndrome de *Li-Fraumeni*. Com isso, algumas particularidades na modelagem de ontologias para outros domínios podem não ter ficado em evidência durante a construção da Síndrome de *Li-Fraumeni*;

2. Não tentamos corrigir o *bug* presente no motor de inferência Pellet porque não se tratava do objetivo principal deste trabalho de pesquisa. Como consequência, utilizamos uma abordagem de extração dos axiomas em duas fases que acabou demorando mais tempo do que a abordagem em uma fase apenas, caso existisse uma versão sem o referido *bug*.

3. Não levamos em conta, durante a modelagem da *Li-Fraumeni Ontology*, nenhum fator ambiental ou epigenético para a classificação das famílias. Esses fatores são considerados, pelos médicos e cientistas, como muito importantes no aparecimento de mutações genéticas. Fatores ambientais como poluição do ar, uso de drogas, estrese constante ou casos de depressão contribuem para o aparecimento de cânceres e de outras doenças complexas [HMC+13].

Acreditamos que esse trabalho de pesquisa seja apenas um primeiro passo na construção de uma ontologia de aplicação mais consistente e sólida e que venha a servir de apoio para novos estudos e

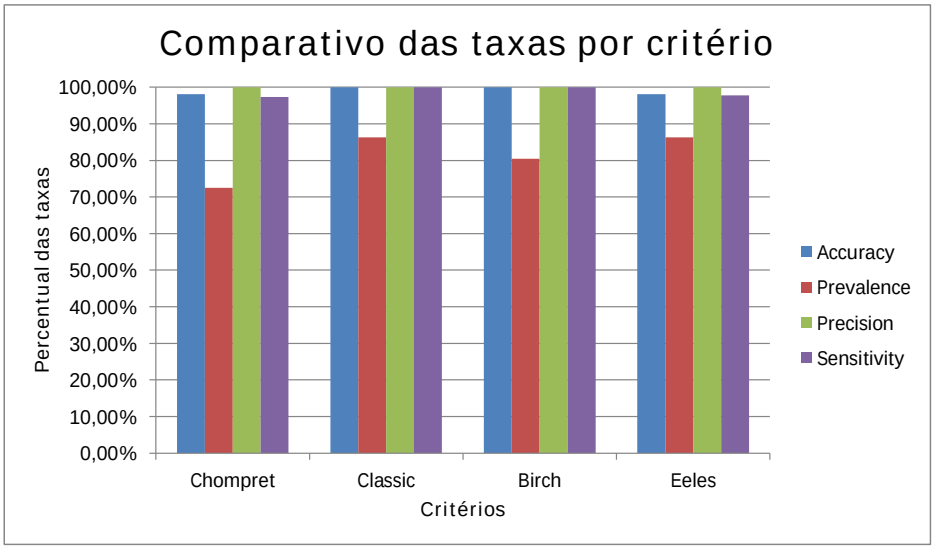
descobertas em outros campos da Biomedicina. Para isso, sugerimos uma continuidade dessa linha de pesquisa propondo, como trabalhos futuros, o amadurecimento da metodologia para a construção da ontologias proposta neste trabalho por meio da modelagem de novas ontologias em diferentes domínios do conhecimento e elaboração de novas métricas para avaliação dos resultados alcançados, o aprimoramento da *Li-Fraumeni Ontology* por meio da inclusão de fatores ambientais e epigenéticos e uma análise dos resultados alcançados pela ontologia sob a ótica da Oncogenética, ou seja, uma análise dos critérios *Li-Fraumeni* em conjunto com outros fatores epigenéticos e ambientais, com o objetivo de encontrar novos padrões que os relacionem ao aparecimento da mutação do $TP53$.

Capítulo 9

Apêndice

9.1 Gráficos Suplementares

Figura 9.1: *Comparativo das taxas de Acurácia, Prevalência, Precisão e Sensibilidade para cada um dos quatro critérios Li-Fraumeni.*



9.2 Tabelas Suplementares

Figura 9.2: *Comparativo das taxas de Acurácia, Prevalência, Precisão e Sensibilidade para os arquivos reais de família Li-Fraumeni.*

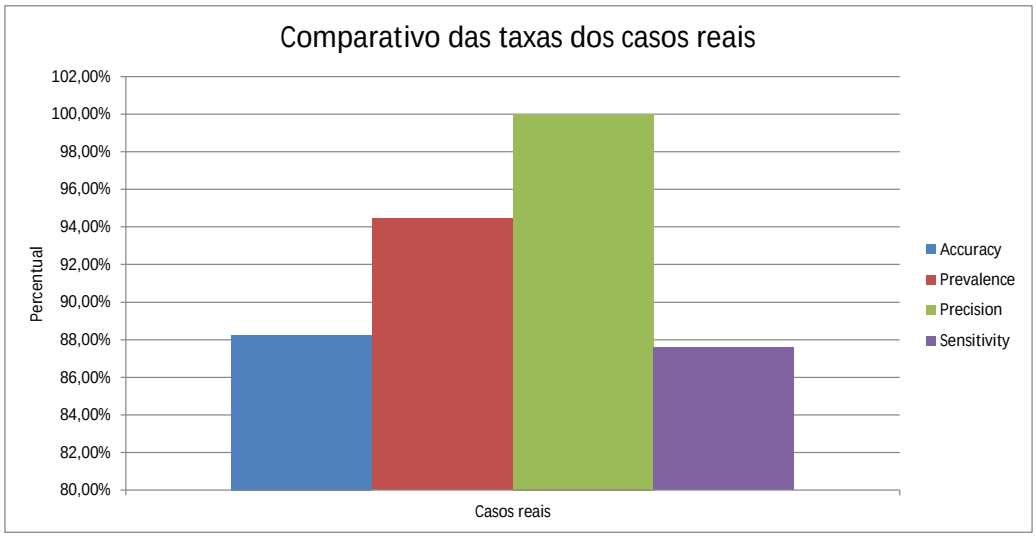
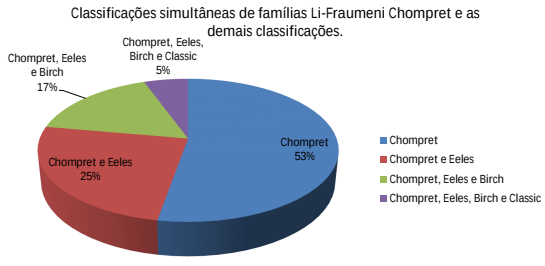
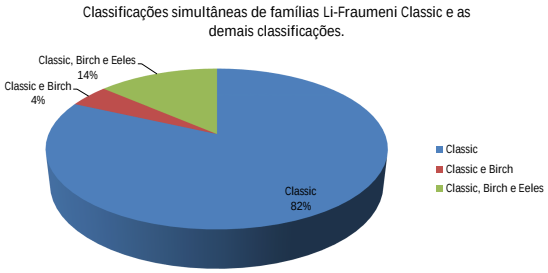


Figura 9.3: *Percentual de famílias que atendem a diferentes critérios concomitantemente.*

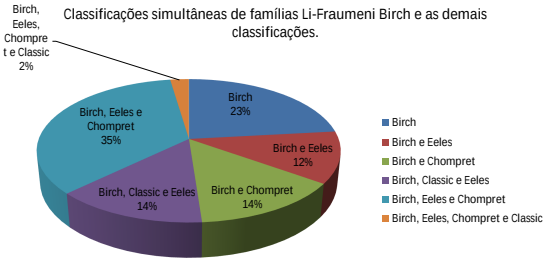
(a) *Percentual de famílias Li-Fraumeni Chompret que também atendem a outros critérios.*



(b) *Percentual de famílias Li-Fraumeni Classic que também atendem a outros critérios.*



(c) *Percentual de famílias Li-Fraumeni Birch que também atendem a outros critérios.*



(d) *Percentual de famílias Li-Fraumeni Eeles que também atendem a outros critérios.*

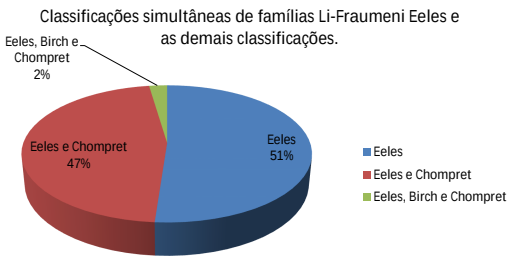


Figura 9.4: *Comparação dos tempos gastos por etapa na ferramenta* `Directed-Extract-LiFraumeni` *para cada um dos critérios LiFraumeni.*

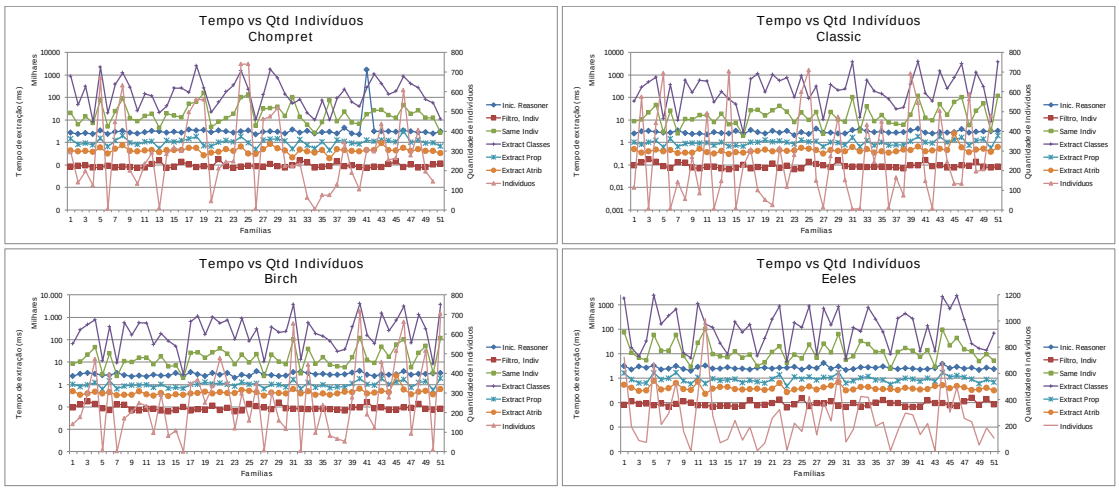


Figura 9.5: *Comparação dos tempos gastos por etapa na ferramenta* `Directed-Extract-LiFraumeni` *para cada um dos arquivos reais Li-Fraumeni.*

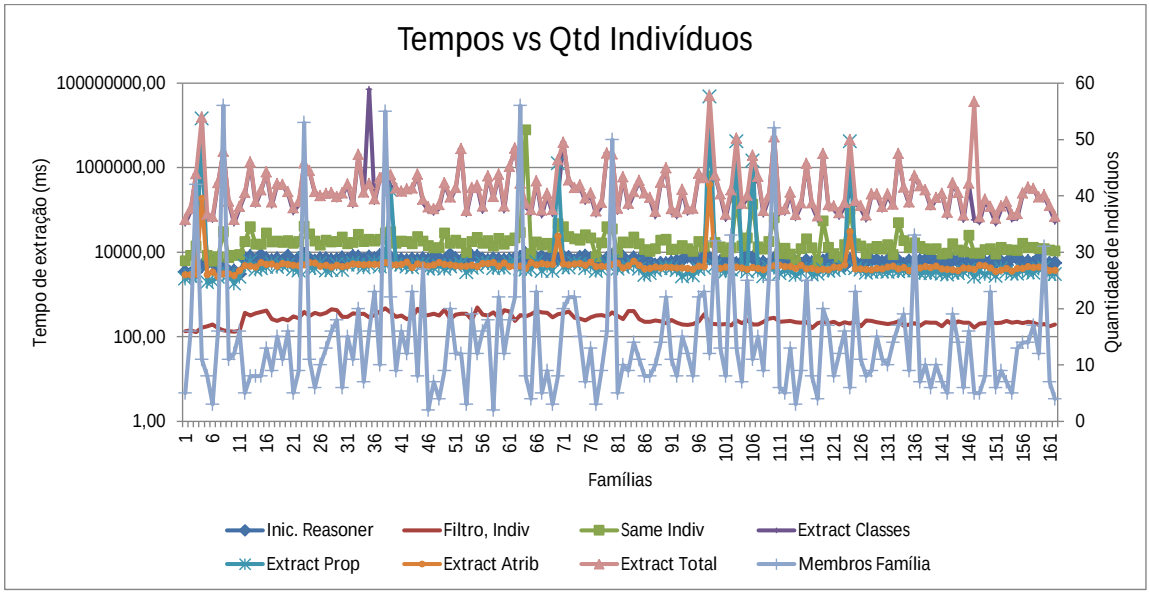


Tabela 9.1: *Tempos gastos em cada etapa do processo de classificação para cada uma das famílias do critério Eeles.*

	Família	Inic. Reasoner	Filt Indiv	Same Indiv	Ext Clas	Ext Prop	Ext Atrib	Ext Total	# Classes	# Indiv
0_Airal.owl	1	3114,97	82,91	73505,56	1807769,41	1673,33	548,65	1883498,21	1864	724
10_Hoipfjig.owl	2	2269,86	116,08	10694,99	17671,55	878,01	402,04	29647,87	1864	190
11_Yzutt.owl	3	2994,19	90,60	6985,94	7734,72	640,94	304,31	15667,19	1864	84
12_Thet.owl	4	2705,55	93,81	5400,31	32321,71	640,03	319,31	38682,61	1864	71
13_Audooz.owl	5	3230,47	77,98	58598,84	2385553,65	1667,48	776,08	2446597,32	1864	681
14_Aud.owl	6	2256,17	94,76	13081,38	165510,51	882,51	346,78	179822,45	1864	209
15_Heithjib.owl	7	2494,91	69,37	13549,93	355484,88	994,32	390,37	370420,76	1864	292
16_Audis.owl	8	3191,98	90,10	57739,87	643674,66	1697,28	644,93	703758,09	1864	530
17_Oissaus.owl	9	2358,19	115,90	8199,82	11040,32	800,00	364,68	20406,04	1864	160
18_Eintainn.owl	10	1984,49	100,60	2884,91	6668,21	585,71	336,56	10476,56	1864	6
19_Eiv.owl	11	2865,72	77,49	27651,34	1122843,31	1109,93	736,61	1152342,47	1864	439
1_Ais.owl	12	3342,72	79,20	105364,04	164272,73	621,24	231,81	177401,61	1864	1017
20_Qyg.owl	13	2425,40	68,83	9573,36	118747,26	961,70	373,44	129657,02	1864	277
21_Ed.owl	14	2510,90	76,05	7757,83	26597,26	818,49	446,64	35621,53	1864	68
22_Oibjin.owl	15	2819,88	74,21	7315,81	9434,73	853,00	447,90	18052,79	1864	95
23_Igh.owl	16	2446,22	69,70	12472,75	201300,52	907,71	362,09	215044,31	1864	239
24_Pout.owl	17	2517,28	74,70	7025,30	74868,36	669,28	345,35	82909,47	1864	87
25_Stadjish.owl	18	2244,15	124,91	9282,62	158753,92	809,60	366,70	169214,09	1864	189
26_Fleiw.owl	19	2656,98	79,65	3139,06	8074,66	720,46	381,77	12317,16	1864	8
27_Toist.owl	20	2633,23	82,30	4331,61	42035,86	643,59	325,58	47337,86	1864	66
28_Echoif.owl	21	2505,60	96,60	11715,19	255448,85	917,72	370,24	268453,27	1864	255
29_Geih.owl	22	2669,81	136,94	20231,67	877619,44	1389,78	638,44	899880,73	1864	324
2_Memib.owl	23	2728,56	64,92	4226,44	4864,03	485,24	273,60	9850,47	1864	12
30_Aufauw.owl	24	2886,85	84,47	9372,30	185132,84	1112,84	349,15	195968,35	1864	220
31_Bjss.owl	25	2258,09	153,00	6372,82	118068,37	940,56	375,93	125758,90	1864	156
32_Ykit.owl	26	2792,49	72,04	23670,48	871817,43	1157,22	462,99	897109,32	1864	423
33_Ov.owl	27	2585,83	96,34	6845,57	9463,48	845,57	343,79	17499,59	1864	129
34_Jss.owl	28	4242,54	93,53	25828,69	714094,50	1096,40	423,07	741444,04	1864	380
35_Aunned.owl	29	2464,95	115,31	11431,00	148467,06	1049,77	492,49	161441,66	1864	234
36_Kym.owl	30	3093,23	75,01	62556,16	834755,75	1695,88	701,71	899710,83	1864	628
37_Haith.owl	31	2215,98	67,51	5705,13	5978,29	639,65	321,13	12645,43	1864	76
38_Pouf.owl	32	2391,21	94,89	7701,34	117280,95	792,37	347,08	126123,10	1864	171
39_Glaisjid.owl	33	2742,55	70,83	32478,22	79623,69	1061,98	446,73	113611,83	1864	421
3_Choob.owl	34	2754,48	79,59	24582,02	778229,38	1194,62	435,37	804442,71	1864	416
40_Aizoim.owl	35	2666,62	99,60	11579,96	251913,69	839,99	362,44	264697,27	1864	194
41_Kjs.owl	36	3050,40	131,13	12259,37	76955,92	846,94	361,06	90424,54	1864	227
42_Yhoup.owl	37	2600,42	96,96	2438,58	9589,40	570,53	359,10	12958,85	1864	8
43_Oigh.owl	38	2342,80	95,37	11834,58	273096,06	791,99	332,77	286056,60	1864	163
44_Jsh.owl	39	2538,03	69,14	17290,85	440273,56	999,64	410,14	458975,41	1864	296
45_Gaugett.owl	40	2455,76	67,48	13115,09	264880,64	931,40	377,58	279305,90	1864	281
46_Mour.owl	41	2239,30	70,02	7612,44	12920,90	743,13	346,96	21624,56	1864	132
47_Aih.owl	42	2334,34	124,90	14128,27	138478,81	988,31	365,49	153962,10	1864	216
48_Oodes.owl	43	2560,46	95,31	2965,23	12955,67	746,40	469,77	17138,40	1864	6
49_Oint.owl	44	3916,11	97,42	94006,63	2151999,87	1693,48	515,66	2248216,93	1864	688
4_Auf.owl	45	2625,19	78,76	21901,23	677913,34	1118,02	393,10	701326,86	1864	301
50_Imauf.owl	46	2872,35	73,99	44510,72	2300086,14	1273,16	477,18	2346348,39	1864	492
5_Vessouss.owl	47	2475,00	114,27	15034,86	242920,32	1069,59	418,46	259444,57	1864	254
6_Prof.owl	48	2639,73	155,90	12028,19	25141,98	912,24	338,11	38421,83	1864	230
7_Stauvath.owl	49	2208,04	81,86	5002,38	16542,23	622,97	341,97	22510,78	1864	51
8_Yp.owl	50	2738,31	137,58	9395,27	14029,72	870,28	412,56	24709,15	1864	181
9_Ooshoosh.owl	51	2331,45	86,87	5217,42	68446,95	692,89	316,03	74674,47	1864	105

Tabela 9.2: *Tempo total de processamento de cada família (em minutos) e a quantidade de indivíduos em cada arquivo (`Patients` e `Documents`).*

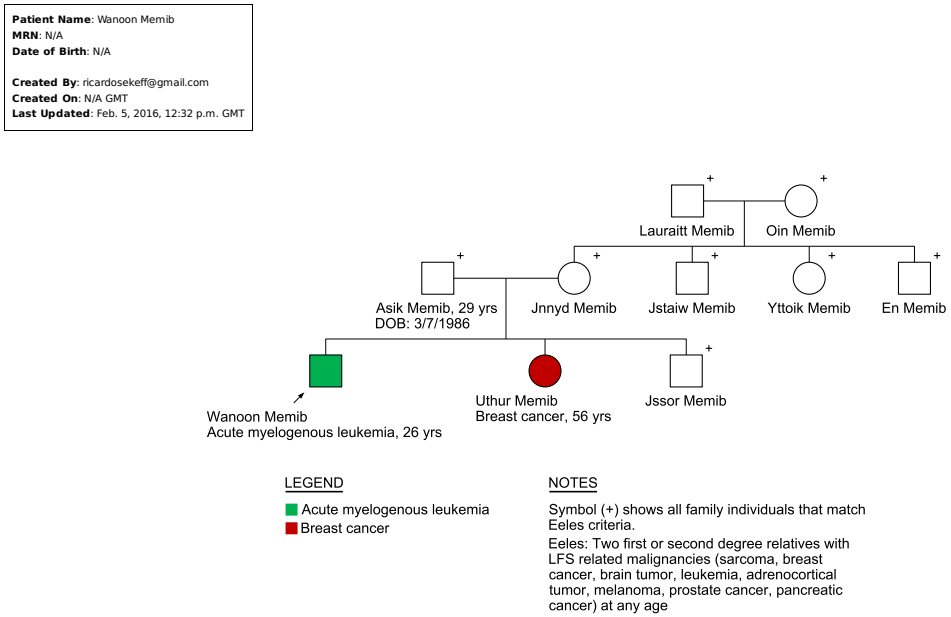
Família	Tempo (Min)	Indiv
FAM00999.rdf	0,99	17
FAM01001.rdf	1,78	39
FAM01003.rdf	11,85	92
FAM01004.rdf	261,67	37
FAM01005.rdf	1,34	23
FAM01006.rdf	1,17	14
FAM01007.rdf	7,33	41
FAM01008.rdf	40,40	142
FAM01009.rdf	2,67	24
FAM01010.rdf	1,00	28
FAM01011.rdf	2,02	37
FAM01013.rdf	4,26	25
FAM01014.rdf	22,34	24
FAM01017.rdf	2,63	19
FAM01019.rdf	5,01	24
FAM01020.rdf	13,20	35
FAM01021.rdf	2,56	28
FAM01023.rdf	6,91	35
FAM01024.rdf	6,67	28
FAM01025.rdf	4,51	46
FAM01026.rdf	1,87	16
FAM01027.rdf	2,63	24
FAM01029.rdf	21,36	103
FAM01030.rdf	14,05	28
FAM01031.rdf	4,53	17
FAM01032.rdf	3,64	25
FAM01033.rdf	4,11	30
FAM01035.rdf	4,26	50
FAM01036.rdf	3,41	41
FAM01037.rdf	4,18	19
FAM01038.rdf	6,75	34
FAM01039.rdf	2,62	32
FAM01041.rdf	34,23	66
FAM01042.rdf	4,35	25
FAM01043.rdf	6,96	59
FAM01044.rdf	3,06	52
FAM01045.rdf	9,65	31
FAM01046.rdf	13,63	147
FAM01047.rdf	11,41	70
FAM01048.rdf	5,42	26
FAM01049.rdf	4,57	42
FAM01050.rdf	5,24	26
FAM01051.rdf	5,48	49
FAM01053.rdf	11,54	28
FAM01055.rdf	2,97	74
FAM01056.rdf	1,95	9
FAM01057.rdf	1,79	23
FAM01058.rdf	2,21	23
FAM01060.rdf	7,36	27
FAM01064.rdf	3,38	46
FAM01065.rdf	5,64	31
FAM01067.rdf	46,65	26
FAM01068.rdf	1,61	12
FAM01069.rdf	5,58	42
FAM01070.rdf	6,28	27
FAM01071.rdf	2,04	40
FAM01072.rdf	10,70	42
FAM01074.rdf	3,44	11
FAM01075.rdf	11,54	56
FAM01076.rdf	2,04	31
FAM01078.rdf	17,73	37
FAM01081.rdf	47,65	66
FAM01083.rdf	7,18	127
FAM01084.rdf	2,40	25
FAM01085.rdf	1,69	15
FAM01086.rdf	8,13	62
FAM01087.rdf	1,63	14
FAM01088.rdf	4,11	29
FAM01090.rdf	1,69	10
FAM01091.rdf	25,77	39
FAM01092.rdf	66,27	75
FAM01093.rdf	8,44	56
FAM01094.rdf	5,61	54
FAM01097.rdf	6,51	44
FAM01098.rdf	3,16	18
FAM01099.rdf	4,19	31
FAM01100.rdf	1,56	9
FAM01101.rdf	2,47	21
FAM01102.rdf	37,13	54
FAM01103.rdf	34,76	146
FAM01104.rdf	1,84	20
FAM01105.rdf	10,00	28
FAM01106.rdf	2,39	28
FAM01107.rdf	4,52	44
FAM01109.rdf	8,31	25
FAM01110.rdf	4,14	20
FAM01111.rdf	3,29	23
FAM01112.rdf	1,52	22
FAM01113.rdf	7,43	40
FAM01114.rdf	16,47	50
FAM01116.rdf	1,79	32
FAM01119.rdf	1,51	26
FAM01120.rdf	5,13	34
FAM01122.rdf	1,76	31
FAM01123.rdf	1,81	20
FAM01126.rdf	12,15	53
FAM01127.rdf	9,65	51
FAM01129.rdf	841,67	32
FAM01131.rdf	11,22	81
FAM01132.rdf	3,95	35
FAM01136.rdf	1,32	20
FAM01137.rdf	4,98	72
FAM01138.rdf	82,31	39
FAM01139.rdf	2,39	26
FAM01140.rdf	3,72	60
FAM01141.rdf	32,57	37
FAM01142.rdf	10,32	35
FAM01143.rdf	1,61	22
FAM01144.rdf	3,57	61
FAM01146.rdf	88,78	212
FAM01147.rdf	1,67	16
FAM01148.rdf	1,76	24
FAM01149.rdf	4,40	35
FAM01150.rdf	1,28	12
FAM01151.rdf	2,47	24
FAM01152.rdf	20,70	89
FAM01153.rdf	2,42	28
FAM01155.rdf	1,21	12
FAM01156.rdf	35,46	46
FAM01157.rdf	2,19	37
FAM01158.rdf	2,04	21
FAM01159.rdf	1,42	31
FAM01160.rdf	2,47	36
FAM01162.rdf	75,94	35
FAM01163.rdf	2,76	57
FAM01164.rdf	2,12	30
FAM01165.rdf	1,25	25
FAM01166.rdf	3,95	29
FAM01169.rdf	3,96	44
FAM01172.rdf	2,02	24
FAM01173.rdf	3,99	25
FAM01174.rdf	2,27	36
FAM01175.rdf	35,58	41
FAM01177.rdf	5,77	42
FAM01178.rdf	2,54	22
FAM01179.rdf	10,93	77
FAM01180.rdf	6,16	20
FAM01181.rdf	4,92	28
FAM01182.rdf	2,24	23
FAM01183.rdf	3,25	25
FAM01184.rdf	4,12	27
FAM01186.rdf	1,45	16
FAM01187.rdf	7,32	50
FAM01188.rdf	4,32	34
FAM01189.rdf	1,27	22
FAM01190.rdf	7,17	37
FAM01191.rdf	610,50	14
FAM01193.rdf	1,15	20
FAM01194.rdf	2,99	33
FAM01195.rdf	2,03	54
FAM01197.rdf	1,06	19
FAM01199.rdf	2,03	27
FAM01200.rdf	2,64	19
FAM01201.rdf	1,30	18
FAM01203.rdf	1,37	30
FAM01204.rdf	4,15	36
FAM01205.rdf	5,72	38
FAM01206.rdf	5,38	38
FAM01209.rdf	3,37	31
FAM01210.rdf	3,82	67
FAM01211.rdf	2,10	19
FAM01212.rdf	1,17	15

Tabela 9.3: *Consumo de Memória (em MB) em cada arquivo de família Li-Fraumeni por fases.*

Família	Fase 1	Fase 2	%
FAM00999	1.886	2440	129,37%
FAM01001	2.376	2959	124,54%
FAM01003	2.438	3044	124,86%
FAM01004	2.500	2782	111,28%
FAM01005	1.923	3307	171,97%
FAM01006	1.972	2791	141,53%
FAM01007	1.433	2002	139,71%
FAM01008	2.285	4306	188,45%
FAM01009	2.205	1798	81,54%
FAM01010	1.914	1788	93,42%
FAM01011	3.255	2703	83,04%
FAM01013	1.647	2546	154,58%
FAM01014	3.110	2267	72,89%
FAM01017	1.814	3221	177,56%
FAM01019	3.252	2026	62,30%
FAM01020	3.111	1994	64,10%
FAM01021	3.308	1979	59,82%
FAM01023	2.610	1608	61,61%
FAM01024	2.822	2731	96,78%
FAM01025	3.160	2910	92,09%
FAM01026	1.565	2092	133,67%
FAM01027	1.939	2801	144,46%
FAM01029	3.161	3789	119,87%
FAM01030	1.751	2029	115,88%
FAM01031	3.185	1464	45,97%
FAM01032	2.919	2413	82,67%
FAM01033	2.859	1444	50,51%
FAM01035	2.258	2631	116,52%
FAM01036	2.146	3025	140,96%
FAM01037	2.300	2452	106,61%
FAM01038	2.700	3289	121,81%
FAM01039	1.370	2868	209,34%
FAM01041	3.087	3583	116,07%
FAM01042	2.992	2591	86,60%
FAM01043	3.172	2399	75,63%
FAM01044	2.265	3312	146,23%
FAM01045	1.556	2309	148,39%
FAM01046	1.986	2379	119,79%
FAM01047	3.266	3191	97,70%
FAM01048	2.870	2197	76,55%
FAM01049	2.080	2629	126,39%
FAM01050	3.333	1730	51,91%
FAM01051	3.164	1999	63,18%
FAM01053	2.798	3546	126,73%
FAM01055	1.928	2679	138,95%
FAM01056	1.308	3219	246,10%
FAM01057	2.857	2795	97,83%
FAM01058	3.166	3143	99,27%
FAM01060	2.857	2267	79,35%
FAM01064	2.075	2892	139,37%
FAM01065	3.107	2155	69,36%
FAM01067	2.019	2629	130,21%
FAM01068	1.708	1892	110,77%
FAM01069	1.882	1870	99,36%
FAM01070	3.241	1830	56,46%
FAM01071	2.095	3106	148,26%
FAM01072	2.849	2475	86,87%
FAM01074	2.924	2935	100,38%
FAM01075	2.342	3417	145,90%
FAM01076	1.979	3244	163,92%
FAM01078	2.652	3044	114,78%
FAM01081	2.213	2625	118,62%
FAM01083	3.228	2957	91,60%
FAM01084	2.305	1960	85,03%
FAM01085	1.546	1515	97,99%
FAM01086	2.597	2302	88,64%
FAM01087	1.810	2009	110,99%
FAM01088	2.820	2922	103,62%
FAM01090	1.958	1790	91,42%
FAM01091	1.952	2862	146,62%
FAM01092	2.769	3028	109,35%
FAM01093	3.157	3512	111,24%
FAM01094	2.211	3211	145,23%
FAM01097	2.933	1685	57,45%
FAM01098	1.241	1526	122,97%
FAM01099	1.652	2213	133,96%
FAM01100	2.073	2074	100,05%
FAM01101	2.028	1733	85,45%
FAM01102	3.040	3801	125,03%
FAM01103	3.147	2775	88,18%
FAM01104	1.998	3302	165,27%
FAM01105	1.376	1764	128,20%
FAM01106	1.999	1433	71,69%
FAM01107	3.251	2841	87,39%
FAM01109	2.096	1801	85,93%
FAM01110	1.577	2199	139,44%
FAM01111	3.241	2990	92,26%
FAM01112	1.828	1412	77,24%
FAM01113	2.660	2998	112,71%
FAM01114	2.936	2294	78,13%
FAM01116	1.674	2936	175,39%
FAM01119	1.802	2261	125,47%
FAM01120	2.739	1288	47,02%
FAM01122	1.831	1480	80,83%
FAM01123	1.607	3256	202,61%
FAM01126	2.991	1713	57,27%
FAM01127	2.096	3067	146,33%
FAM01129	3.112	2044	65,68%
FAM01131	2.669	2794	104,68%
FAM01132	1.759	1951	110,92%
FAM01136	1.527	1789	117,16%
FAM01137	2.975	1486	49,95%
FAM01138	2.202	2965	134,65%
FAM01139	2.871	3374	117,52%
FAM01140	3.037	2799	92,16%
FAM01141	2.728	3152	115,54%
FAM01142	1.442	2452	170,04%
FAM01143	1.625	2515	154,77%
FAM01144	2.855	3490	122,24%
FAM01146	3.754	4289	114,25%
FAM01147	1.795	3051	169,97%
FAM01148	1.431	1444	100,91%
FAM01149	2.842	2431	85,54%
FAM01150	1.836	1554	84,64%
FAM01151	3.046	1458	47,87%
FAM01152	3.082	2802	90,91%
FAM01153	3.304	1404	42,49%
FAM01155	1.717	1424	82,94%
FAM01156	2.414	3719	154,06%
FAM01157	2.307	3255	141,09%
FAM01158	3.161	1517	47,99%
FAM01159	2.008	2674	133,17%
FAM01160	2.768	1868	67,49%
FAM01162	1.281	2231	174,16%
FAM01163	2.332	2158	92,54%
FAM01164	3.338	2762	82,74%
FAM01165	1.350	1529	113,26%
FAM01166	3.300	2092	63,39%
FAM01169	2.988	3098	103,68%
FAM01172	1.887	2358	124,96%
FAM01173	2.644	1554	58,77%
FAM01174	3.307	3019	91,29%
FAM01175	1.584	3170	200,13%
FAM01177	3.263	2287	70,09%
FAM01178	1.723	3179	184,50%
FAM01179	2.360	2822	119,58%
FAM01180	2.022	2032	100,49%
FAM01181	1.293	1968	152,20%
FAM01182	1.692	1468	86,76%
FAM01183	3.347	2669	79,74%
FAM01184	1.572	1908	121,37%
FAM01186	1.938	1849	95,41%
FAM01187	2.233	1484	66,46%
FAM01188	1.571	1731	110,18%
FAM01189	1.607	1836	114,25%
FAM01190	2.665	1492	55,98%
FAM01191	1.718	2020	117,58%
FAM01193	1.866	1911	102,41%
FAM01194	2.738	2659	97,11%
FAM01195	3.258	2271	69,71%
FAM01197	1.635	2065	126,30%
FAM01199	3.164	3264	103,16%
FAM01200	1.381	2747	198,91%
FAM01201	2.004	2058	102,69%
FAM01203	2.046	2927	143,06%
FAM01204	3.167	2433	76,82%
FAM01205	3.152	2062	65,42%
FAM01206	1.647	2215	134,49%
FAM01209	3.258	2495	76,58%
FAM01210	2.806	2221	79,15%
FAM01211	1.239	3232	260,86%
FAM01212	1.954	1975	101,07%

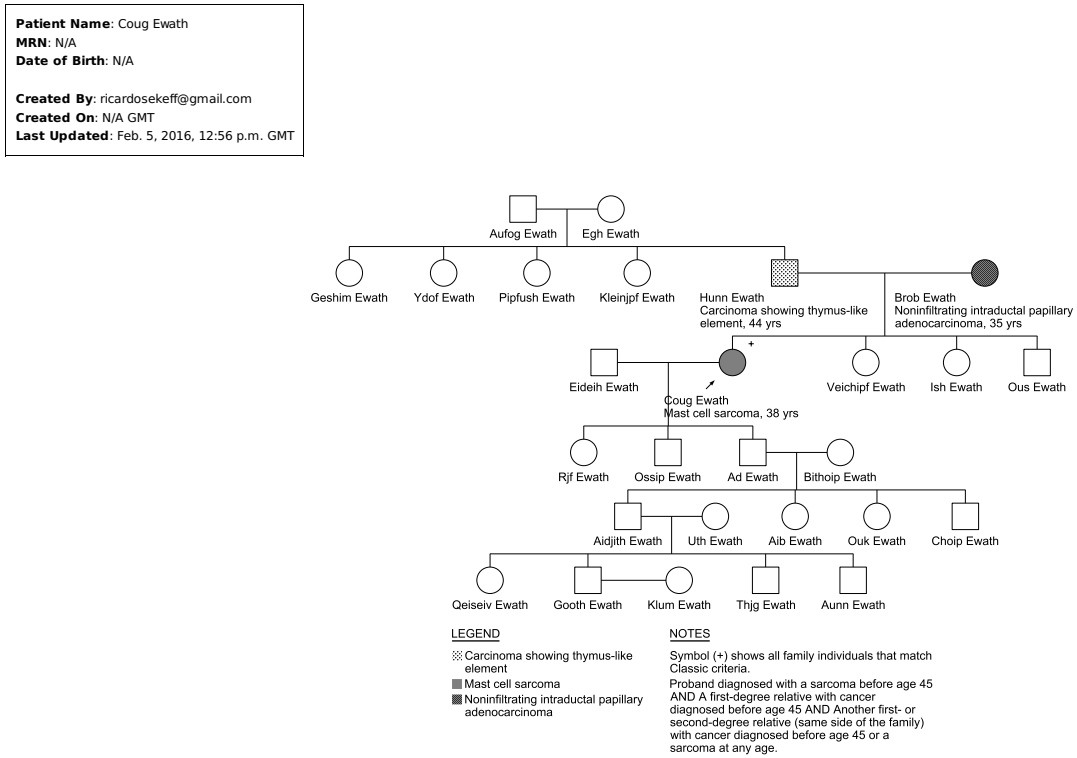
9.3 Heredogramas

Figura 9.6: *Família 02 Memib.*



Família gerada pela ferramenta OpenGLIFS segundo os critérios de Eeles para a Síndrome de Li-Fraumeni. O heredograma foi gerado pela ferramenta on-line Invitae(https://familyhistory.invitae.com). O indivíduo marcado com uma flecha indica o probando.

Figura 9.7: *Família 27 Ewath.*



Família gerada pela ferramenta OpenGLIFS segundo os critérios de Classic para a Síndrome de Li-Fraumeni. O heredograma foi gerado pela ferramenta on-line Invitae(https://familyhistory.invitae.com). O indivíduo marcado com uma flecha indica o probando.

9.4 Lynch Syndrome Ontology

Apresentamos, nessa seção, uma proposta para a Síndrome de *Lynch* a ser integrada com a CDOnto e a GenOnto.

Figura 9.8: *Ontologia para a Síndrome de Lynch apresentada no Protegé.*

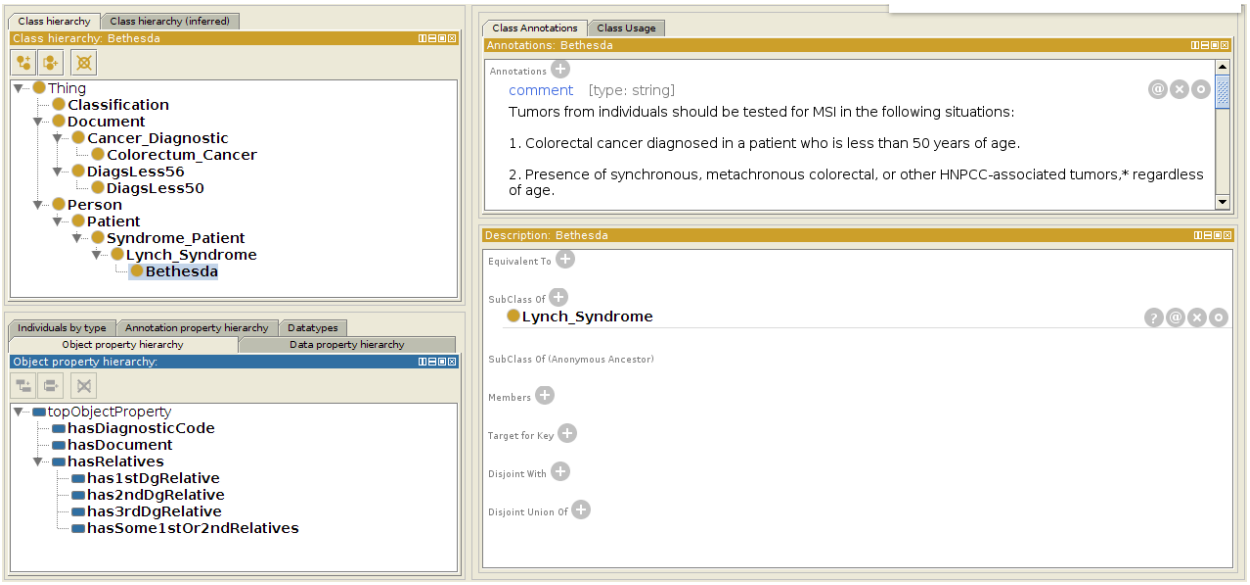
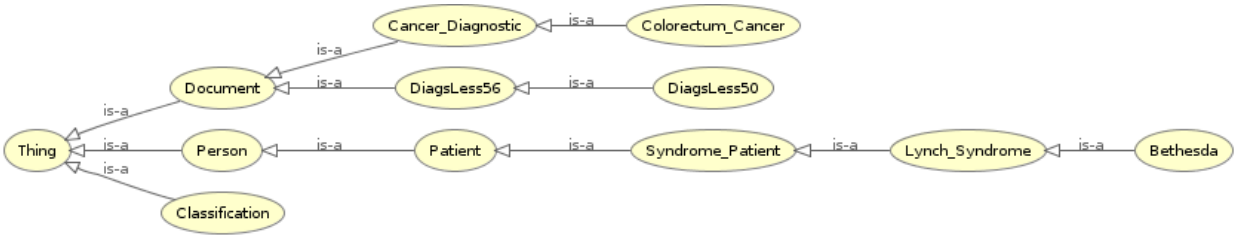
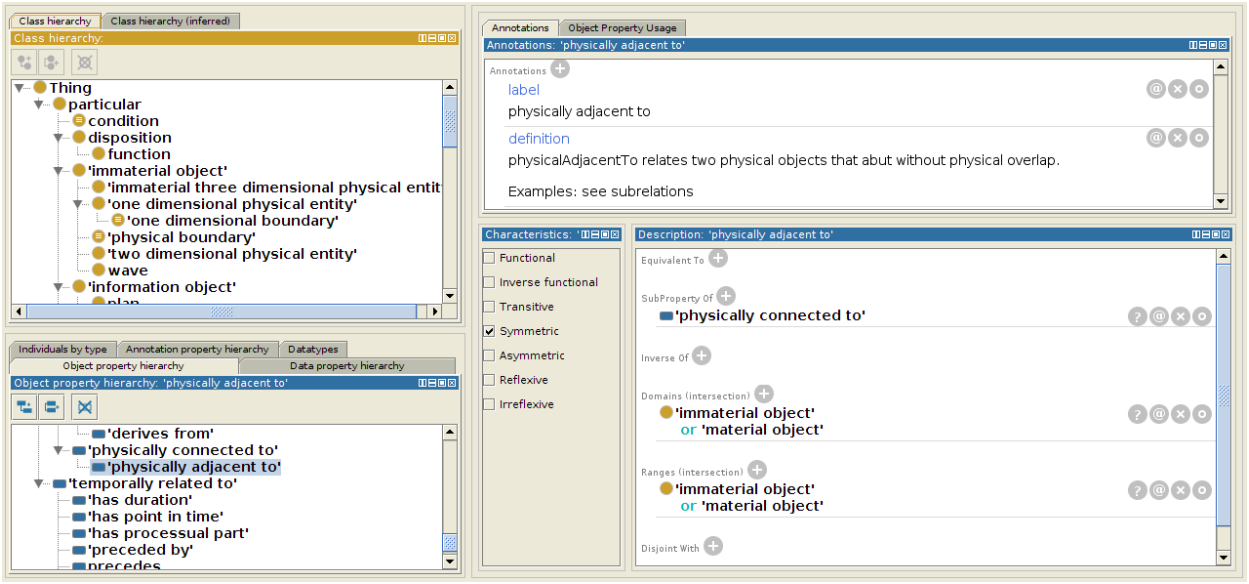


Figura 9.9: *Grafo da Ontologia Síndrome de Lynch.*



9.5 Mapeamento para a BioTopLite

Figura 9.10: *Sugestão de mapeamento da Li-Fraumeni Ontology para a BioTopLite.*



Referências Bibliográficas

[ABB+00] Robert Charles Anderson, Paul Barkley, Robert Booth, Birdie Holsclaw, Robert Velke, John Vincent Wylie et al. Gentech genealogical data model. http://members.ngsgenealogy.org/GENTECH_Data_Model/Description_GENTECH_Data_Model_1.1.doc, 2000. 26

[Ach08] Maria Isabel Alves de Souza Waddington Achatz. *Modificadores de penetrância de mutações germinativas no gene TP53 em famílias brasileiras com diagnóstico clínico da síndrome de Li-Fraumeni e Li-Fraumeni Like: impacto dos polimorfismos intragênicos do TP53 e de genes que regulam a atividade da p53.* Tese de Doutorado, Faculdade de Medicina da Universidade de São Paulo, 2008. 46

[ACM12] Joan Campanyà Artés, Jordi Conesa Caralt e Enric Mayol. Modeling Genealogical Domain - An Open Problem. Em *KEOD*, páginas 202–207, 2012. 26, 68

[AH06] Sören Auer e Heinrich Herre. The RapidOWL Methodology–Towards Agile Knowledge Engineering. Em *15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'06)*, páginas 352–357, 2006. xiii, 17, 44, 57, 58

[AMA14] American Medical Association AMA. Preparing for the ICD-10 Code Set: October 1, 2014 Compliance Date. Relatório técnico, 2014. Disponível em https://www.unitypoint.org/waterloo/filesimages/ford10-differences.pdf. Acessado em 26-11-2015. xvii, 32, 61

[AOC+07] Maria Isabel Waddington Achatz, Magali Olivier, Florence Le Calvez, Ghyslaine Martel-Planche, Ademar Lopes, Benedito Mauro Rossi, Patricia Ashton-Prolla, Roberto Giugliani, Edenir Inez Palmero, Fernando Regla Vargas, José Claudio Casali Da Rocha, Andre Luiz Vettore e Pierre Hainaut. The TP53 mutation, R337H, is associated with Li-Fraumeni and Li-Fraumeni-like syndromes in Brazilian families. *Cancer Letters*, 245(1-2):96–102, 2007. 45, 46

[BCH06a] Jie Bao, Doina Caragea e Vasant G Honavar. Modular Ontologies - A Formal Investigation of Semantics and Expressivity. *Order A Journal On The Theory Of Ordered Sets And Its Applications*, 4185:616–631, 2006. 36

[BCH06b] Jie Bao, Doina Caragea e Vasant G. Honavar. Towards collaborative environments for ontology construction and sharing. Em *Collaborative Technologies and Systems, 2006. CTS 2006. International Symposium on*, páginas 99–108, May 2006. 36

[BCK03] Maria Mitzi Brentani, Francisco Ricardo Gualda Coelho e Luiz Paulo Kowalski. *Bases Da Oncologia.* Lemar, 2003. 10

[BCM+03] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi e Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, New York, NY, USA, 2003. 20

[BFO] BFO – Basic Formal Ontology. Disponível em: http://ontology.buffalo.edu/bfo/. Acessado em 26-11-2015. 35

[BHTP94] JM Birch, AL Hartley, KJ Tricker e J Prosser. Prevalence and diversity of constitutional mutations in the p53 gene among 21 Li-Fraumeni families. *Cancer research*, 54(5):1298–304, Março 1994. 46, 49

[BJSSA05] Andrew Burton-Jones, Veda C. Storey, Vijayan Sugumaran e Punit Ahluwalia. A semiotic metrics suite for assessing the quality of ontologies. *Data and Knowledge Engineering*, 55(1):84–102, 2005. 44

[BL96] Tim Berners-Lee. The world wide web: Past, present and future, 1996. Disponível em https://www.w3.org/People/Berners-Lee/1996/ppf.html. Acessado em 19-10-2015. 21

[BL04] Ronald J. Brachman e Hector Levesque. *Knowledge representation and reasoning*. 2004. 76

[BLHL01] T. Berners-Lee, J. Hendler e O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001. 17

[BMB+02] Robin L. Bennett, Arno G. Motulsky, Alan Bittles, Louanne Hudgins, Stefanie Uhrich, Debra Lochner Doyle, Kerry Silvey, C. Ronald Scott, Edith Cheng, Barbara McGillivray, Robert D. Steiner e Debra Olson. Genetic Counseling and Screening of Consanguineous Couples and Their Offspring: Recommendations of the National Society of Genetic Counselors. *Journal of Genetic Counseling*, 11(2):97–119, 2002. 69

[Bor97] Willem Nico Borst. *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. Tese de Doutorado, Universiteit Twente, Enschede, Setembro 1997. Disponível em http://doc.utwente.nl/17864/. 12

[BS85] Ronald J. Brachman e James G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive science*, 9(2):171–216, Junho 1985. 41

[BS02] Alex Borgida e Luciano Serafini. Distributed description logics: Directed domain correspondences in federated information sources. Em *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, páginas 36–53. Springer, 2002. 36

[BSSH08] Elena Beisswanger, Stefan Schulz, Holger Stenzhorn e Udo Hahn. BioTop: An Upper Domain Ontology for the Life Sciences: A Description of Its Current Structure, Contents and Interfaces to OBO Ontologies. *Applied Ontolologies*, 3(4):205–212, Dezembro 2008. 35, 117

[CEST08] Elena Cardillo, Claudio Eccher, Luciano Serafini e Andrei Tamilin. Logical analysis of mappings between medical classification systems. Em *Artificial Intelligence: Methodology, Systems, and Applications*, páginas 311–321. Springer, 2008. 31

[CFLGP03] Oscar Corcho, Mariano Fernández-López e Asunción Gómez-Pérez. Methodologies, tools and languages for building ontologies. Where is their meeting point? *Data & Knowledge Engineering*, 46(1):41–64, 2003. 21

[CG05] Diego Calvanese e Giuseppe De Giacomo. Data integration: A logic-based perspective. *Artificial Intelligence Magazine*, 126(1):1–18, 2005. 37

[CGC] Consanguinity Fact Sheet – Debunking Common Myths. http://www.larasig.com/node/2020. Acessado: 21-05-2014. xvii, 70

[CGL11] Diego Calvanese, Giuseppe De Giacomo e Domenico Lembo. The Mastro system for ontology-based data access. *Semantic Web*, 2:43–53, 2011. xiii, 40, 41

[CGPFL+12] Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, Mariano Fernández-López, M C Suárez-Figueroa, @bullet A Gómez-Pérez e Mariano Fernández-López. The NeOn methodology for ontology engineering. Em Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, Enrico Motta e Aldo Gangemi, editors, *Ontology engineering in a networked world*, páginas 9–34. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. xiii, 17, 18

[Coh88] Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. L. Erlbaum Associates, 1988. 107

[Con01] T. G. O. Consortium. Creating the Gene Ontology Resource: Design and Implementation. *Genome Research*, 11(8):1425–1433, 2001. 29

[DCHW08] Tharam Dillon, Elizabeth Chang, Maja Hadzic e Pornpit Wongthongtham. Differentiating conceptual modelling from data modelling, knowledge modelling and ontology modelling and a notation for ontology modelling. Em *Proceedings of the Fifth Asia-Pacific Conference on Conceptual Modelling - Volume 79*, APCCM '08, páginas 7–17, Darlinghurst, Australia, Australia, 2008. Australian Computer Society, Inc. 19

[DCtTdK11] Kathrin Dentler, Ronald Cornet, Annette ten Teije e Nicolette de Keizer. Comparison of reasoners for large ontologies in the owl 2 el profile. *Semantic Web*, 2(2):71–87, 2011. 97, 116

[DMN05] Antonio De Nicola, Michele Missikoff e Roberto Navigli. A Proposal for a Unified Process for Ontology Building: UPON. *Database and Expert Systems Applications*, 3588:655–664, 2005. xiii, 17, 18, 56

[DNMN09] Antonio De Nicola, Michele Missikoff e Roberto Navigli. A software engineering approach to ontology building. *Information Systems*, 34(2):258–275, 2009. 55, 56

[DSdC+09] Élida Lívia Rafael Dantas, Fernando Henrique de Lima Sá, Sionara Melo de Figueiredo de Carvalho, Anderson Pontes Arruda, Evelane Marques Ribeiro e Erlane Marques Ribeiro. Genética do câncer hereditário. *Revista Brasileira de Cancerologia*, 55(3):263–9, 2009. 10

[ED13] Faezeh Ensan e Weichang Du. A semantic metrics suite for evaluating modular ontologies. *Information Systems*, 38(5):745–770, 2013. 2, 44

[Eel95] Ra Eeles. Germline mutations in the TP53 gene. *Cancer Survey*, páginas 101–124, 1995. 46, 50

[EN10] Ramez Elmasri e Shamkant Navathe. *Fundamentals of Database Systems*. Addison-Wesley Publishing Company, USA, 6th edição, 2010. 38

[Euz96] Jérôme Euzenat. Corporate Memory Through Cooperative Creation of Knowledge Bases and Hyper-Documents. Em *Proceedings of the 10th Knowledge Acquisition, Modeling and Management for Knowledge-based Systems Workshop (KAW'96)*, páginas 1–18, 1996. 18

[FABP+01] Thierry Frebourg, Anne Abel, Catherine Bonaiti-Pellie, Laurence Brugières, Pascaline Berthet, Brigitte Bressac-de Paillerets, Annie Chevrier, Agnès Chompret, Odile Cohen-Haguenauer, Olivier Delattre, Josué Feingold, Jean Feunteun, Didier Frappaz, Jean-Paul Fricker, Paul Gesta, Philippe Jonveaux, Chantal Kalifa, Catherine Lasset,

Bruno Leheup, Jean-Marc Limacher, Michel Longy, Catherine Nogues, Daniel Oppenheim, Danièle Sommelet, Florent Soubrier, Claude Stoll, Dominique Stoppa-Lyonnet e Henri Tristant. Le syndrome de Li-Fraumeni : mise au point, données nouvelles et recommandations pour la prise en charge. *Bulletin du Cancer.*, 88(6):581–7, Junho 2001. 46, 49

[FL99] Mariano Fernández-López. Overview of Methodologies for Building Ontologies. Em *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem Solving Methods (KRR5), August 2*, 1999. 16, 17, 55

[FLGPJ97] Mariano Fernández-López, Asuncion Gómez-Pérez e Natalia Juristo. Methontology: from ontological art towards ontological engineering. *AAAI-97 Spring Symposium Series*, (Proceedings of the Ontological Engineering AAAI-97 Spring Symposium Series), 1997. xiii, 17, 44, 55

[FPJ+00] April Fritz, Constance Percy, Andrew Jack, Kanagaratnam Shanmugaratnam, Leslie Sobin, D. Max Parkin e Sharon Whelan. *International Classification of Diseases for Oncology. Third Revision.* World Health Organization, Geneva, 2000. 83

[FSM+05] David Fenstermacher, Craig Street, Tara McSherry, Vishal Nayak, Casey Overby e Michael Feldman. The Cancer Biomedical Informatics Grid (caBIGTM). *Conference proceedings : Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 1:743–746, 2005. 32

[FVH+01] Dieter Fensel, Frank Van Harmelen, Ian Horrocks, Deborah L. McGuinness e Peter F. Patel-Schneider. OIL: An ontology infrastructure for the semantic web. *IEEE Intelligent Systems and Their Applications*, 16(2):38–45, 2001. 21

[Gar87] David A Garvin. Competing on the 8 dimensions of quality. *Harvard Business Review*, 65(6):101–109, 1987. 44

[GF95] M. Grüninger e M. Fox. Methodology for the Design and Evaluation of Ontologies. Em *IJCAI'95, Workshop on Basic Ontological Issues in Knowledge Sharing, Abril 13, 1995*, 1995. 17, 44, 59

[GFB+92] Michael Genesereth, Richard E. Fikes, Ronald Brachman, Thomas Gruber, Patrick Hayes, Reed Letsinger, Vladimir Lifschitz, Robert Macgregor, John Mccarthy, Peter Norvig e Ramesh Patil. Knowledge interchange format version 3.0 reference manual, 1992. 41

[GHA07] Stephan Grimm, Pascal Hitzler e Andreas Abecker. Knowledge Representation and Ontologies Logic, Ontologies and Semantic Web Languages. *Semantic Web Services*, páginas 51–105, 2007. 20, 22

[GHM+08] Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, Bijan Parsia, Peter Patel-Schneider e Ulrike Sattler. OWL 2: The next step for OWL. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):309–322, 2008. 22, 43

[GN87] Michael R. Genesereth e Nils J. Nilsson. *Logical Foundations of Artificial Intelligence.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987. 12

[GNB+09] Kelly D. Gonzalez, Katie A. Noltner, Carolyn H. Buzin, Dongqing Gu, Cindy Y. Wen-Fong, Vu Q. Nguyen, Jennifer H. Han, Katrina Lowstuter, Jeffrey Longmate, Steve S. Sommer e Jeffrey N. Weitzel. Beyond Li Fraumeni Syndrome: Clinical Characteristics of Families With p53 Germline Mutations. *Journal of Clinical Oncology*, 27(8):1250–1256, 2009. xvii, 47

[GOS09] Nicola Guarino, Daniel Oberle e Steffen Staab. What is an Ontology? Em *Handbook on Ontologies*. Springer, 2ª edição, 2009. 11, 12

[GPFLD96] Assunción Gómez-Pérez, Mariano Fernández-López e A.J. De Vicente. Towards a method to conceptualize domain ontologies. *Workshop on Ontological Engineering, ECAI'96*, páginas 41–51, 1996. 24

[Gru93] Thomas R. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2):199–220, Junho 1993. 1

[Gru95] Thomas R. Gruber. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human-Computer Studies*, 43(5-6):907–928, Dezembro 1995. 12

[Gua97] Nicola Guarino. Understanding, building and using ontologies. *International Journal of Human-Computer Studies*, 46(2-3):293–310, 1997. 12

[Gua98] Nicola Guarino. *Formal Ontology in Information Systems: Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy.* IOS Press, Amsterdam, The Netherlands, The Netherlands, 1 edição, 1998. xiii, 2, 12, 14, 15, 16, 44

[Gui05] Giancarlo Guizzardi. *Ontological foundations for structural conceptual models.* CTIT PhD Thesis Series, Centre for Telematics and Information Technology, Enschede, The Netherlands, 2005. 1, 11, 12, 15, 23, 25, 67

[GW02] Nicola Guarino e Christopher Welty. Evaluating ontological decisions with OntoClean. *Communications ACM*, 45(2):61–65, 2002. 2

[HDG12] Robert Hoehndorf, Michel Dumontier e Georgios V. Gkoutos. Evaluation of research in biomedical ontologies. *Briefings in bioinformatics*, 14(6):696–712, Setembro 2012. 2, 29

[HL7] About Health Level Seven International. http://www.hl7.org/about/index.cfm?ref=nav. Acessado em 22-11-2015. 32

[HMC+13] Carolyn M. Hutter, Leah E. Mechanic, Nilanjan Chatterjee, Peter Kraft, Elizabeth M. Gillanders, Christian C. Abnet, Christopher Amos, David Balshaw, Heike Bickeböller, Laura Jean Bierut, Paolo Boffetta, Melissa Bondy, Stephen Chanock, Huann Sheng Chen, Nancy Cox, Immaculata De Vivo, Rao Divi, Josee Dupuis, Gary Ellison, Margaret Daniele Fallin, W. James Gauderman, Elizabeth Gillanders, Christopher Haiman, Carolyn Hutter, Naoko Ishibe Simonds, Edwin Iversen, Muin J. Khoury, Loic Le Marchand, Kimberly McAllister, Leah Mechanic, Ulrike Peters, Ross Prentice, Timothy Rebbeck, Jill Reedy, Nathaniel Rothman, Sheri Schully, Daniela Seminara, Daniel Shaughnessy, Sanjay Shete, Donna Spiegelman, Daniel O. Stram, Duncan Thomas, Molin Wang, Wendy Wang, Clarice Weinberg, Deborah M. Winn e John S. Witte. Gene-environment interactions in cancer epidemiology: A national cancer institute think tank report. *Genetic Epidemiology*, 37(7):643–657, 2013. 69, 120

[HNOS+07] Melissa a. Haendel, Fabian Neuhaus, David Osumi-Sutherland, Paula M. Mabee, José L. V. Mejino Jr., Chris J. Mungall e Barry Smith. CARO - The Common Anatomy Reference Ontology. Em *Anatomy Ontologies for Bioinformatics: Principles and Practice*, páginas 311–333. 2007. 33

[Hoe10] Robert Hoehndorf. What is an upper level ontology? http://ontogenesis.knowledgeblog.org/740, 2010. Acessado em 08-10-2015. 23, 24

[Hor51] Alfred Horn. On sentences which are true of direct unions of algebras. *The Journal of Symbolic Logic*, 16(1):14–21, 1951. 71

[HSV08] Gergely Héja, György Surján e Péter Varga. Ontological analysis of SNOMED CT. *BMC medical informatics and decision making*, 8(Suppl 1):S8, 2008. 30

[IHT] IHTSDO - International Health Terminology Stantards Development Organization. http://www.ihtsdo.org/snomed-ct/what-is-snomed-ct/history-of-snomed-ct. Acessado em 8-11-2015. 30

[Jor04] Lynn B. Jorde. *Genética Médica*. Elsevier (Medicina), Rio de Janeiro, 3 edição, 2004. 9, 10, 11, 46

[Kaz08] Yevgeny Kazakov. SRIQ and SROIQ are Harder than SHOIQ. Em *Description Logics*, 2008. 43

[KLWZ03] Oliver Kutz, Carsten Lutz, Frank Wolter e Michael Zakharyaschev. E-connections of Description Logics. Em *Proceedings of 2003 International Workshop on Description Logics*, páginas 178–187, Rome, Italy, 2003. 36

[Kon15] Bogumil M. Konopka. Biomedical ontologies - A review. *Biocybernetics and Biomedical Engineering*, 35(2):75–86, 2015. 29, 30

[KSD01] Atanas Kiryakov, Kiril Iv. Simov e Marin Dimitrov. OntoMap: portal for upper-level ontologies. páginas 47–58, 2001. 23

[KW04] Gruber SB. Kohlmann W. Lynch Syndrome. Disponível em http://www.ncbi.nlm.nih.gov/books/NBK1211/, 2004. Atualizado em 22-05-2014. 117

[Len01] Maurizio Lenzerini. Data integration is harder than you thought. *Cooperative Information Systems*, 2172:22–26, Setembro 2001. 41

[Len02] Maurizio Lenzerini. Data Integration: A Theoretical Perspective. Em *Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '02, páginas 233–246, New York, NY, USA, 2002. ACM. xiii, 37, 43

[LeP06] Paea LePendu. Integrating Databases into the Semantic Web through an Ontology-Based Framework. *22nd International Conference on Data Engineering Workshops (ICDEW'06)*, páginas 54–54, 2006. 40

[LFM+88] Federick Pei Li, Joseph F Fraumeni, John J Mulvihill, William A. Blattner, Margaret G Dreyfurs, Margaret A. Tucker e Robert W. Miller. A cancer family syndrome in twenty-four kindreds. *Cancer Research*, páginas 5358–5362, Setembro 1988. 48, 49

[LG90] Douglas B. Lenat e Ramanathan. V. Guha. *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1 edição, 1990. 17

[Lib03] Leonid Libkin. Expressive power of SQL. *Lecture Notes in Computer Science*, 1973, Outubro 2003. 39

[LJ69] Federick Pei Li e Joseph F Fraumeni Jr. Soft-tissue sarcomas, breast cancer, and other neoplasms. A familial syndrome?. *Annals Of Internal Medicine*, 71(4):747–752, 1969. 47, 49

[LRH09] Markus Luczak-Rösch e Ralf Heese. *Networked Knowledge - Networked Media: Integrating Knowledge Management, New Media Technologies and Semantic Systems*, chapter Managing Ontology Lifecycles in Corporate Settings, páginas 235–248. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. 17

[LSR96] Sean Luke, Lee Spector e David Rager. Ontology-based knowledge discovery on the world-wide web. Em *Working Notes of the Workshop on Internet-Based Information Systems at the 13th National Conference on Artificial Intelligence (AAAI96)*, páginas 96–102, 1996. 21

[LTGP04] Adolfo Lozano-Tello e Asunción Gómez-Pérez. Ontometric: A method to choose the appropriate ontology. *Journal of Database Management*, 2(15):1–18, 2004. 44

[LZTJ10] Yongquan Liang, Hongmei Zhu, Qijia Tian e Shujuan Ji. A method for OWL ontology module partition. *Proceedings - 2010 IEEE 2nd Symposium on Web Society, SWS 2010*, páginas 372–377, 2010. 36

[Mal11] David Malkin. Li-fraumeni syndrome. *Genes & cancer*, 2(4):475–84, Abril 2011. 46, 48

[MCBV12] Carmen Martinez-Cruz, Ignacio J. Blanco e M. Amparo Vila. Ontologies Versus Relational Databases: Are They So Different? A Comparison. *Artificial Intelligence Review*, 38(4):271–290, Dezembro 2012. xiii, 2, 15, 16, 19

[MM09] Manuel Möller e Saikat Mukherjee. Context-Driven Ontological Annotations in DICOM Images-Towards Semantic Pacs. Em *HEALTHINF*, páginas 294–299, 2009. 31

[MRW77] Jim A McCall, Paul K Richards e Gene F Walters. Factors in software quality. Concepts and definitions of software quality. Relatório técnico, DTIC Document, 1977. 44

[MSE13] Manuel Möller, Daniel Sonntag e Patrick Ernst. Modeling the international classification of diseases (ICD-10) in OWL. Em *Knowledge Discovery, Knowledge Engineering and Knowledge Management*, volume 272 CCIS, páginas 226–240. Springer, 2013. 31

[NCI] National Cancer Institute - Cancer Staging Fact Sheets. http://www.cancer.gov/about-cancer/diagnosis-staging/staging/staging-fact-sheet. Acessado em 27-09-2015. 11

[NM01] Natalya F. Noy e Deborah L. Mcguinness. Ontology Development 101: A Guide to Creating Your First Ontology. http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html, 2001. Acessado em 7-10-2015. xiii, 12, 13, 14, 17, 44, 58, 59, 67, 70

[OBO] OBO Foundry Principles. http://obofoundry.org/principles/fp-001-open.html. Acessado em 22-11-2015. 33

[OTK02] Tomoko Ohta, Yuka Tateisi e Jin-Dong Kim. The genia corpus: An annotated research abstract corpus in molecular biology domain. Em *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, páginas 82–86, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. 35

[Pas13] Adrian Paschke. Ontomaven: Maven-based ontology development and management of distributed ontology repositories. Em *Proceedings of the 9th International Workshop on Semantic Web Enabled Software Engineering (SWESE 2013)*, 2013. 18

[PB07] Jane Peace e PF Brennan. Ontological representation of family and family history. *AMIA - American Medical Informatics Association. Annual Symposium proceedings*, página 1072, Janeiro 2007. 26, 27

[PL08] Antonella Poggi e Domenico Lembo. Linking data to ontologies. *Journal on Data Semantics X*, páginas 133–173, 2008. 40

[Pre11] Roger S Pressman. *Engenharia de software*. McGraw Hill Brasil, 2011. 44

[RCCG14] Mariela Rico, María Laura Caliusco, Omar Chiotti e María Rosa Galli. OntoQualitas: A framework for ontology quality assessment in information interchanges between heterogeneous systems. *Computers in Industry*, 65(9):1291–1300, 2014. 44

[Rec03] Alan L. Rector. Modularisation of domain ontologies implemented in description logics and related formalisms including OWL. *Proceedings of the international conference on Knowledge capture - K-CAP '03*, página 121, 2003. 36

[RN09] Stuart Russell e Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3 edição, 2009. 1, 20

[RRP96] Alan L. Rector, J. E. Rogers e P. Pole. The GALEN high level ontology. *Studies in Health Technology and Informatics*, 34:174–178, 1996. 29

[SAR+07] Barry Smith, Michael Ashburner, Cornelius Rosse, Jonathan Bard, William Bug, Werner Ceusters, Louis J Goldberg, Karen Eilbeck, Amelia Ireland, Christopher J Mungall, Neocles Leontis, Philippe Rocca-Serra, Alan Ruttenberg, Susanna-Assunta Sansone, Richard H Scheuermann, Nigam Shah, Patricia L Whetzel e Suzanna Lewis. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature biotechnology*, 25(11):1251–1255, 2007. 32, 33

[SB13] Stefan Schulz e Martin Boeker. BioTopLite: An Upper Level Ontology for the Life Sciences. Evolution, Design and Application. Em *Informatik 2013*, páginas 1889–1899, 2013. 35, 117

[SCC97] Kent A Spackman, Keith E Campbell e Roger A Côté. SNOMED RT: a reference terminology for health care. Em *Proceedings of the AMIA annual fall symposium*, página 640. American Medical Informatics Association, 1997. 29, 30

[SCK+05] Barry Smith, Werner Ceusters, Bert Klagges, Jacob Köhler, Anand Kumar, Jane Lomax, Chris Mungall, Fabian Neuhaus, Alan L Rector e Cornelius Rosse. Relations in biomedical ontologies. *Genome biology*, 6(5):R46, 2005. xvii, 33, 34

[SHN] Semantic Health Network. Disponível em: http://www.semantichealthnet.eu/index.cfm/news/. Acessado em 28-11-2015. 35

[SJ02] Kiril Simov e Stanislav Jordanov. BOR: a pragmatic DAML+ OIL reasoner. *On-To-Knowledge Project*, 2002. Disponível em http://iswc2002.semanticweb.org/posters/simov-jordanov.pdf. Acessado em 19-10-2015. 21

[Soc14] American Cancer Society. Cancer Facts and Figures. Special Section: Cancer in Children & Adolescents. *American Cancer Society*, páginas 25–42, 2014. 83

[SRGBSA12] Miguel-Angel Sicilia, Daniel Rodríguez, Elena García-Barriocanal e Salvador Sánchez-Alonso. Empirical findings on ontology metrics. *Expert Systems with Applications*, 39(8):6706–6711, 2012. 2

[SS09a] Steffen Staab e Rudi Studer. *Handbook on Ontologies*. Springer Publishing Company, Incorporated, 2 edição, 2009. 20

[SS09b] Robert Stevens e Margaret Stevens. A family history knowledge base using OWL 2. Em *CEUR Workshop Proceedings*, volume 432, 2009. 26

[SSMJ13] Robert Stevens, Margaret Stevens, Nicolas Matentzoglu e Simon Jupp. *Manchester Family History Advanced OWL Tutorial*. University of Manchester, 1.0 edição, 2013. 71

[SSSS01] Steffen Staab, Rudi Studer, Hans Peter Schnurr e York Sure. Knowledge processes and ontologies. *IEEE Intelligent Systems and Their Applications*, 1:26–34, 2001. 17

[Suj01] Walter Sujansky. Heterogeneous database integration in biomedicine. *Journal of biomedical informatics*, 34(4):285–98, Agosto 2001. 1, 37

[TAM+05] Samir Tartir, I. Budak Arpinar, Michael Moore, Amit P. Sheth e Boanerges Aleman-Meza. OntoQA: Metrics-Based Ontology Quality Analysis. 2005. 44

[TBBD+09] Julie Tinat, Gaelle Bougeard, Stéphanie Baert-Desurmont, Stéphanie Vasseur, Cosette Martin, Emilie Bouvignies, Olivier Caron, Brigitte Bressac-de Paillerets, Pascaline Berthet, Catherine Dugast, Catherine Bonaïti-Pellié, Dominique Stoppa-Lyonnet e Thierry Frébourg. 2009 version of the Chompret criteria for Li Fraumeni syndrome. *Journal of clinical oncology : official journal of the American Society of Clinical Oncology*, 27(26):e108–9; author reply e110, Setembro 2009. 46, 49

[TBS+15] Lindsey A. Torre, Freddie Bray, Rebecca L. Siegel, Jacques Ferlay, Joannie Lortet-Tieulent e Ahmedin Jemal. Global cancer statistics, 2012. *CA: A Cancer Journal for Clinicians*, 65(2):87–108, 2015. 7

[Tsu03] Jun'ichi Tsujii. The genia project website. http://www.geniaproject.org/home, 2003. Acessado em 22-11-2015. 35

[UG96] Mike Uschold e Michael Gruninger. ONTOLOGIES: Principles, Methods and Applications. Em *The Knowledge Engineering Review*, volume 11. June 1996. 15, 44, 73

[UK95] Mike Uschold e Martin King. Towards a Methodology for Building Ontologies. Em *Workshop on Basic Ontological Issues in Knowledge Sharing (IJCAI)*, volume 80, páginas 275–280, 1995. 17, 44

[vEB06] Andrew C von Eschenbach e Kenneth Buetow. Cancer Informatics Vision: caBIG™. *Cancer Informatics*, 2:22–24, 2006. 32

[VSD+04] Jean-luc Verschelde, Dos Santos, Tom Deray, Barry Smith e Werner Ceusters. Ontology-Assisted Database Integration to Support Natural Language Processing and Biomedical Data Mining. Em *Journal of Integrative Bioinformatics*, volume 1, 2004. 37

[W3Ca] OWL 2 Web Ontology Language Document Overview. http://www.w3.org/TR/owl2-overview/. Acessado em 8-10-2015. 20, 22, 41

[W3Cb] OWL 2 Web Ontology Language Document Overview. http://www.w3.org/TR/owl-ref/. Acessado em 19-10-2015. 21, 22

[W3Cc] OWL 2 Web Ontology Language Document Overview - Class Expression. http://www.w3.org/TR/owl2-syntax/#Class_Expressions. Acessado em 19-10-2015. 79

[WBHQ07] Yimin Wang, Jie Bao, Peter Haase e Guilin Qi. Evaluating Formalisms for Modular Ontologies in Distributed Information Systems. Em Massimo Marchiori, JeffZ. Pan e ChristiandeSainte Marie, editors, *Web Reasoning and Rule Systems*, volume 4524, chapter Evaluating, páginas 178–193. Springer Berlin Heidelberg, 2007. 36, 44

[Wei13] François Weil. *Family Trees*. Harvard University Press, 2013. 25

[WHO92] World Health Organization WHO. *International Classification of Diseases and Related Health Problems, Tenth Revision*. World Health Organization, Geneva, 1992. 31, 83

[WHO15a] World Health Organization WHO. Initial WHO response to the report of the external review of the ICD-11 revision. http://www.who.int/classifications/icd/whoresponseicd11.pdf, 2015. Acessado em 23-11-2015. 31

[WHO15b] World Health Organization WHO. Report of ICD-11 Revision Review. http://www.who.int/classifications/icd/reportoftheicd11review14april2015.pdf, 2015. Acessado em 23-11-2015. 31

[Woo10] Charla Woodbury. Automatic extraction from and reasoning about genealogical records: A prototype. http://dagwood.cs.byu.edu/deg/papers/Charla.Thesis.pdf, 2010. 26

[WVV+01] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann e S. Hübner. Ontology-based integration of information – a survey of existing approaches. Em *IJCAI–01 Workshop: Ontologies and Information Sharing*, páginas 108–117, 2001. 40

[Yu06] Alexander C Yu. Methods in biomedical ontology. *Journal of biomedical informatics*, 39(3):252–66, 2006. 30

[Zan05] Ivo Zandhuis. Towards a Genealogical Ontology for the Semantic Web. Em *XVIth International Conference of the Association for History and Computing*, páginas 1–8, 2005. 26

Índice Remissivo