

OntoForInfoScience e Onto4ALLEditor: metodologia e editor de ontologias como facilitadores na construção de ontologias por especialistas do domínio e cientistas da informação

Fabício Martins Mendonça¹

Lucas Piazzi de Castro²

Resumo: Atualmente, ontologias são usadas com frequência em ciências como a Linguística, Ciência da Informação e Ciência da Computação, especialmente, nas áreas de Modelagem Conceitual e Representação do Conhecimento. Um dos grandes desafios em ontologias é seu processo de construção, oneroso e complexo, que pode levar a erros de representação do conhecimento e recuperação da informação. O uso de metodologias e editores de ontologia ainda representa um desafio aos especialistas do domínio e profissionais da informação, cuja formação na área não privilegia requisitos para tais atividades. Diante desse cenário, o presente artigo apresenta uma metodologia própria para construção de ontologias - *OntoForInfoScience* - e um software editor de ontologias - *Onto4ALLEditor* -, cujo propósito principal é fornecer uma base teórica e funcionalidades da ferramenta para a fim de facilitar o desenvolvimento de ontologias por diferentes tipos de usuários, principalmente especialistas do domínio e cientistas da informação. A pesquisa inclui também uma avaliação dos editores de ontologia, disponíveis atualmente, e uma breve discussão sobre a utilização e possibilidades da *OntoForInfoScience* e do *Onto4ALLEditor* nesse contexto. Como resultados esperados da abordagem proposta, pretende-se contribuir com a popularização do uso de ontologias em Ciência da Informação e áreas afins.

Palavras-chave: Construção de ontologia. Editor de ontologia. Engenharia ontológica. Representação do Conhecimento. Ciência da Informação.

¹ Professor. Doutor em Ciência da Informação. Universidade Federal de Juiz de Fora. Email: fabricao.mendonca@ice.ufjf.br

² Graduando. Ciência da Computação. Universidade Federal de Juiz de Fora. Email: lpiazzi@ice.ufjf.br

*OntoForInfoScience and Onto4ALLEditor:
methodology and ontology editor as facilitators in ontology building
by experts domain and information scientists*

Abstract: Currently, ontologies are frequently used in sciences such as Linguistics, Information Science and Computer Science, especially in the areas of Conceptual Modeling and Knowledge Representation. One of the great challenges in ontologies is their costly and complex construction process, which can lead to errors in knowledge representation and information retrieval. The use of ontology methodologies and editors still represents a challenge to domain experts and information professionals, whose formation in the area does not privilege requirements for such activities. In this scenario, this paper presents a building methodology - OntoForInfoScience - and an ontology editor software - Onto4ALLEditor - whose main purpose is to provide a theoretical basis and functionalities of the tool in order to facilitate the development of ontologies by different types of users, mainly domain experts and information scientists. The research also includes an evaluation of ontology editors, currently available, and a brief discussion on the use and possibilities of OntoForInfoScience and Onto4ALLEditor in this context. As expected results of the proposed approach, it is intended to contribute to the popularization of the use of ontologies in Information Science and related areas.

Keywords: Ontology building. Ontology editor. Ontology engineering. Knowledge representation. Information Science.



1 INTRODUÇÃO

A partir dos anos 1990, verificou-se um grande número de iniciativas para o uso de ontologias em sistemas de informação. Apontava-se a inconsistência das práticas nos primeiros anos de modelagem como a principal causa dos problemas de falta de integração entre sistemas (Smith e Welty, 2001). Ainda que massivamente adotadas nos últimos 15 anos, ontologias já eram citadas nos anos 1960 (Mealy, 1967), além de iniciativas significativas nos anos 1980 (Wand, Storey e Weber, 1999) e nos anos 1990 (Gruber, 1992) (Guarino, 1998), até sua popularização nos anos 2000. Ao longo de todos esses anos, a criação de modelos conceituais foi motivada pela busca de melhorias na representação em sistemas de informação.

Por todo esse histórico, o uso de ontologias se concentra nas áreas da Modelagem Conceitual e de Representação do Conhecimento. Entretanto, outras áreas científicas fazem uso de ontologias, por exemplo, a Ciência da Informação (Almeida 2013) e a Linguística (Schalley 2019). A Ciência da Informação (CI) herdou métodos de classificação e indexação, atividades que são parte do núcleo duro da área. Por ser Ciência Social Aplicada, a CI se aproximou de teorias da Filosofia, por exemplo (Smith 2004) resultando deficiência prática como a construção de ontologias. Além disso, a formação da CI não privilegia requisitos para construir ontologias, como lógica e modelagem, dentre outros.

De fato, a construção de ontologias é um processo bastante complexo e oneroso, principalmente para desenvolvedores de ontologia, que não estão tão familiarizados com o jargão técnico da área, além de conceitos de lógica formal e filosóficos necessários nesta atividade. Na computação, conceitos comuns na modelagem orientada a objetos, tais como classes abstratas e derivadas, herança, disjunção e sobreposição, cardinalidades, além do conhecimento em formalismos e linguagens lógicas, como a linguagem *Ontology Web Language* (OWL), são exemplos de conteúdo técnico nem sempre familiares a estudantes e profissionais da área de CI, bem como a especialistas de diferentes domínios para os quais se busca a representação do conhecimento através de ontologias.

A partir da década de 1990, foi possível notar o surgimento de diferentes metodologias para construção de ontologias, tais como: metodologia de Gruninger e Fox - TOVE (Gruninger e Fox, 1995), metodologia de Uschold e King - Enterprise (Uschold e King, 1995), Methontology (Gómez-Perez, Fernandez-Lopes e Vicente 1996), método 101 (Noy e McGuinness 2001), metodologia NeOn (Suaréz-Figueroa 2008); Up for ONtology (UPON) (De Nicola, Missikoff e

Navigli 2009). Mais recentemente, foi proposta a metodologia OntoForInfoScience (Mendonça 2015), que tem como característica principal uma abordagem mais detalhada das etapas e tarefas do ciclo de desenvolvimento ontológico, de forma a possibilitar uma melhor compreensão das atividades necessárias à construção de ontologias para diferentes tipos de desenvolvedores, dos menos aos mais experientes. No presente artigo, faz-se uma breve apresentação de tal metodologia, correlacionando-a com as tarefas realizadas no editor de ontologias Onto4ALLEditor, o qual foi concebido com base na fundamentação teórica da OntoForInfoScience (Mendonça et al. 2020).

Assim como as metodologias de construção, foram desenvolvidos, também a partir da década de 1990, editores de ontologia com as mais diversas características e cujo objetivo geral é fornecer mais agilidade e facilidade aos seus usuários na modelagem ontológica, inclusão dos elementos necessários às ontologias, verificação e validação dos erros de modelagem, além de formalização do conteúdo em alguma linguagem lógica. O número de editores é bastante extenso, para citar alguns tem-se: Protégé (Musen 2015), Neon Toolkit (Michael e Waterfeld 2012), OLED (Guerson et al 2015), Top Braid Composer (Top Quadrant, 2019), OBOEdit (Day-Richter, Harris e Haendel 2007) e o próprio Onto4ALLEditor (Mendonça et al. 2020) (Mendonça et al. 2021).

O uso de editores de ontologia populares como o Protégé (Munsen 2015) ainda é um desafio em Ciência da Informação e entre especialistas do domínio, que podem envolver médicos, engenheiros, químicos, biólogos, linguistas, entre outros. Apesar de ser uma ferramenta com mais de duas décadas de utilização e a melhor avaliada entre seus usuários (Malik, 2017) (Warren, 2013), estes ainda encontram dificuldades no uso de recursos que o Protégé oferece. Alguns problemas comuns, apontados em entrevistas com usuários do Protégé, Web Protégé, TopBraid e SWOOP são: i) necessidade de um ambiente mais colaborativo entre especialistas do domínio para modelagem da ontologia; ii) falta de suporte para visualização de ontologias em ambientes preferidos dos usuários; iii) feedback ruim na depuração de erros da ontologia; iv) falta de suporte para pesquisa de termos em ontologias externas para reuso; v) interface pouco amigável na navegação pela hierarquia de classes; entre outros (Vigo 2014).

Diante do cenário apresentado, justifica-se a discussão do tema construção de ontologias e um olhar mais detalhado para a metodologia OntoForInfoScience e o software Onto4ALLEditor a fim de verificar se tal projeto fornece facilidades aos profissionais de informação e especialistas



do domínio na construção de ontologias, além de possíveis contribuições na área. Cabe destacar que embora o editor se oriente pela OntoForInfoScience, trata-se de software independente e, por isso, pode ser utilizado com qualquer outra metodologia ou mesmo sem seguir nenhuma (Mendonça et al. 2021). As etapas da OntoForInfoScience, no escopo editor, servem para orientar os usuários nas etapas necessárias de construção, mesmo que segui-las não seja obrigatório.

O presente artigo está estruturado nas seguintes seções: a seção 1 de introdução, apresentou o contexto e problema de pesquisa, além de justificativas e objetivos; a seção 2 cita metodologias para construção de ontologias (subseção 2.1) e destaca a OntoForInfoScience (subseção 2.2); a seção 3 discorre sobre editores de ontologia, incluindo uma visão geral sobre os editores populares e aqueles recomendados em padrões como o W3C (subseção 3.1); e a apresentação dos recursos e funcionalidades do Onto4ALLEditor (subseção 3.2). A seção 4 apresenta a discussão do artigo, destacando as limitações atuais das metodologias e editores de ontologia, bem como as possíveis contribuições da metodologia e editor apresentados aos desenvolvedores de ontologia. Por fim, a Seção 5 traz considerações finais deste artigo.

2 METODOLOGIAS DE CONSTRUÇÃO E A ONTOFORINFOSCIENCE

Esta seção do artigo faz referência às metodologias de construção de ontologias mais citadas na literatura da área e destaca também, de forma mais detalhada, a metodologia OntoForInfoScience e suas etapas.

2.1 Uma visão geral sobre metodologias para ontologias

Devido ao uso e aplicações de ontologias a partir dos anos de 1990 para representação do conhecimento, foram desenvolvidas a partir de então uma série de metodologias, métodos e propostas de como conduzir a construção de uma ontologia. Através da literatura da área, é possível perceber que houve um aumento significativo de metodologias e métodos de construção entre meados da década de 90 até o início dos anos 2000, algumas das mais citadas são: metodologia TOVE (Gruninger e Fox, 1995), metodologia de Uschold e King - Enterprise (Uschold e King, 1995), Methontology (Gómez-Perez, Lopes, Vicente, 1996) e o método 101 (Noy e McGuinness, 2001). De meados dos anos 2000 até hoje, surgiram novas metodologias de construção em um número menor se comparado ao período anterior. Algumas delas são: metodologia UP for ONtology (UPON) (De Nicola, Missikoff e Navigli 2009), metodologia NeOn (Suárez-Figueroa 2010) e metodologia OntoForInfoScience (Mendonça 2015).



De maneira geral, tais metodologias dão suporte aos desenvolvedores de ontologia nas tarefas que devem ser executadas no processo de construção, elucidando etapas deste processo e como devem ser realizadas. A existência de diferentes metodologias deve-se, em grande parte, a não existência de um padrão metodológico comum e unificado para área, muito em razão de ser ainda um ramo de pesquisa recente se comparado às metodologias tradicionais da Ciência da Informação ou às metodologias de processo de software da Ciência da Computação.

Trabalhos de revisão de literatura na área (Jones, Bench-Capon e Visser, 1998) (Fernandéz-López 1999) (Fernandéz-López e Corcho, 2004) (Silva, Souza e Almeida, 2008) (Santos, 2014) (Mendonça, 2015) descrevem as metodologias para ontologias mais conhecidas e citadas, as quais estão apresentadas no Quadro 1, incluindo seu nome, descrição breve e etapas previstas.

Quadro 1 – Metodologias para construção de ontologias

METODOLOGIA	DESCRIÇÃO (objetivos e domínio de aplicação)	ETAPAS (processos e atividades)
Metodologia de Gruninger e Fox – TOVE (GRUNINGER e FOX, 1995)	Criada no projeto <i>Toronto Virtual Enterprise (TOVE)</i> , cujo objetivo era criar um modelo de senso comum compartilhado corporativo. Domínio de aplicação: Negócios (empresarial)	<ol style="list-style-type: none"> 1. Elaboração de cenários de motivação 2. Questões de competência informal 3. Concepção da terminologia formal 4. Questões de competência formal 5. Especificação de axiomas formais; 6. Verificação de teoremas completos.
Metodologia de Uschold e King ENTERPRISE (USCHOLD e KING, 1995)	Desenvolvido com base na prática da construção da ontologia <i>Enterprise Ontology</i> , que descreve conhecimento no domínio corporativos. Domínio de aplicação: Negócios (empresarial)	<ol style="list-style-type: none"> 1. Propósito, grau de formalismo e usuário 2. Construção da ontologia em três etapas: <ol style="list-style-type: none"> a) conceitualização b) implementação c) integração com ontologias já existentes; 3. Avaliação 4. Documentação
<i>Methontology</i> (GÓMEZ-PEREZ, FERNANDEZ-LOPES e VICENTE, 1996)	Possibilita a construção de ontologias por reengenharia, utilizando-se do conhecimento de domínio. Domínio de aplicação: Diversos.	<ol style="list-style-type: none"> 1. Especificação: 2. Aquisição de Conhecimento 3. Conceitualização 4. Integração 5. Implementação 6. Avaliação 7. Documentação
Método <i>Kactus</i>	Método derivado do projeto Kactus que permite reutilização de conhecimento	<ol style="list-style-type: none"> 1. Lista de necessidades ou requisitos

(BERNARAS, LARESGOTTI, CORERA, 1996)	em sistemas técnicos, tal como redes elétricas. Domínio de aplicação: Sistemas de complexidade técnica.	<ol style="list-style-type: none"> 2. Identificação de termos relevantes 3. Criação de modelo preliminar 4. Estruturação e refinamento 5. Integração e reutilização.
Método <i>Sensus</i> (SWARTOUT et al., 1996).	Método derivado da ontologia <i>Sensus</i> , desenvolvida para o processamento de linguagem natural. Domínio de aplicação: Diversos.	<ol style="list-style-type: none"> 1. Identificar termos-chave do domínio; 2. Ligar os termos-chave à ontologia <i>Sensus</i>; 3. Adicionar novos termos para o domínio; 4. Adicionar subárvores completas.
Método 101 (NOY e GUINNESS, 2001)	Método usado no desenvolvimento de exemplos práticos utilizando o editor de ontologias Protégé. Domínio de aplicação: Diversos.	<ol style="list-style-type: none"> 1. Determinar escopo 2. Considerar o reuso de termos 3. Enumerar termos 4. Definir classes 5. Organizar as classes em uma taxonomia 6. Definir propriedades e suas restrições 7. Adicionar valores de instâncias
Método CYC (REED e LENAT, 2002)	Método usado na construção da ontologia CYC, que busca abranger o considera o conhecimento consensual do mundo. Domínio de aplicação: Diversos.	<ol style="list-style-type: none"> 1. Extração manual do conhecimento 2. Extração auxiliada por computador 3. Desenvolvimento de representação 4. Representação do conhecimento de diferentes domínios usando primitivas.
<i>On-to-Knowledge Methodology (OTKM)</i> (SURE, STAAB e STUBER, 2003)	Metodologia desenvolvida para a construção de ontologias para aplicações de gestão do conhecimento Domínio de aplicação: Gestão do conhecimento empresarial.	<ol style="list-style-type: none"> 1. Estudo de viabilidade 2. <i>Kickoff</i> 3. Refinamento 4. Avaliação 5. Aplicação e evolução 6. Geração de conhecimento 7. Captura de conhecimento 8. Recuperação e acesso
Metodologia <i>UP for ONtology (UPON)</i> (DE NICOLA, MISSIKOFF e NAVIGLI, 2009)	Metodologia de construção de ontologias derivada em padrão de engenharia de software Processo Unificado Domínio de aplicação: Negócios (e-bussiness).	<ol style="list-style-type: none"> 1. Workflow de Requisitos 2. Workflow de Análise 3. Workflow de Desenvolvimento 4. Workflow de Implementação 5. Workflow de Teste
Metodologia <i>NeON</i> (SUARÉZ-FIGUEROA, 2010)	Metodologia para construção de redes ontológicas baseado em um desenvolvimento colaborativo Domínio de aplicação: Diversos.	<ol style="list-style-type: none"> 1. Gerenciamento de processos e atividades 2. Desenvolvimento orientado de processos e atividades 3. Suporte aos processos e atividades

<p>Metodologia <i>MFPFO</i> (LIM, LIU e LEE, 2011)</p>	<p>Metodologia de construção de ontologia multi-facetada, anotada semanticamente, para a modelagem de uma família de produtos. Domínio de Aplicação: Domínios que possuem uma família de produtos</p>	<ol style="list-style-type: none"> 1. Construção de uma taxonomia da família de produtos; 2. Extração de entidades; 3. Identificação do conceito e geração da unidade facetada; 4. Anotação semântica e modelagem faceta; 5. Construção de uma ontologia de família de produtos multi-facetada e anotada semanticamente; 6. Avaliação e validação da ontologia.
<p><i>OntoForInfoScience</i> (MENDONÇA, 2015)</p>	<p>Metodologia teve sua origem nas dificuldades conhecidas da área de CI para construção de ontologias. Seu objetivo principal e diferencial é detalhar os passos metodológicos, explicando termos técnicos, questões lógicas e filosóficas envolvidas, que raramente são de conhecimento ou domínio dos interessados. Domínio de Aplicação: Diversos.</p>	<ol style="list-style-type: none"> 0. Avaliação da necessidade da ontologia 1. Especificação 2. Aquisição e Extração de Conhecimento 3. Conceitualização 4. Fundamentação Ontológica 5. Formalização da Ontologia 6. Avaliação 7. Documentação 8. Disponibilização

Fonte: adaptado de Mendonça (2015)

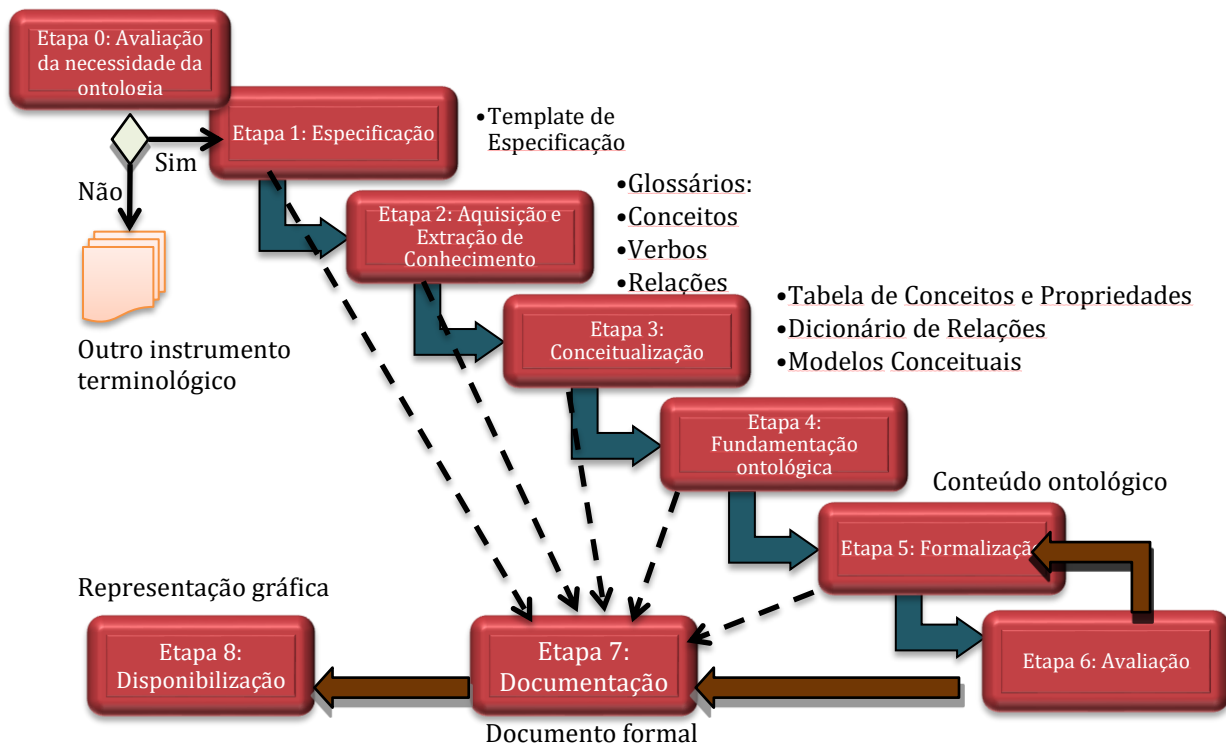
2.2 Metodologia OntoForInfoScience

A metodologia OntoForInfoScience foi criada com o propósito de possibilitar que especialistas em Organização do Conhecimento, incluindo cientistas da informação e outros desenvolvedores, possam superar problemas relativos ao jargão técnico e às questões lógicas e filosóficas no desenvolvimento de ontologias (Mendonça, 2015). Para tanto, a OntoForInfoScience detalha os passos necessários do processo de construção de ontologias em uma linguagem simples e adequada a profissionais diversos, característica que a diferencia das atuais metodologias existentes (Mendonça e Soares, 2017).

Ainda que apresente uma característica particular, a metodologia OntoForInfoScience também se baseou em algumas etapas de metodologias já existentes, especificamente a Methontology, o método 101 e a metodologia NeOn, de forma a reutilizar algumas etapas e suprir as limitações presentes em cada uma delas: detalham-se as etapas e passos reutilizados, além de realizar adaptações necessárias para obter uma linguagem apropriada a todos desenvolvedores de ontologia.

A metodologia OntoForInfoScience engloba, ao todo, uma pré-etapa e oito etapas, conforme apresentado na Figura 1. No restante da presente seção apresenta-se uma breve descrição dessas etapas. A descrição completa está disponível em Mendonça (2015).

Figura 1: Etapas da metodologia OntoForInfoScience



Fonte: Mendonça e Soares (2017)

Na pré-etapa ou etapa 0 (zero), o desenvolvedor realiza uma avaliação prévia sobre a real necessidade da construção de uma ontologia para o problema analisado. Se o objetivo desse desenvolvimento consiste, por exemplo, unicamente na indexação e recuperação de informações em um domínio específico, a construção de um tesouro é suficiente. Entretanto, se o objetivo consiste na representação de aspectos e objetos do mundo real e também a necessidade do uso de relações mais extensíveis do que aquelas contidas em vocabulários controlados, a construção de uma ontologia é fundamental.

Decidida a necessidade de desenvolver a ontologia inicia-se a sequência de etapas do processo de construção. Na etapa 1 realiza-se a especificação da ontologia através do template de especificação, o qual deve conter informações sobre: o domínio e escopo, propósito geral, classes de usuários que representam o público-alvo da ontologia, cenários de aplicação para uso da ontologia e o grau de formalidade. Estabelece-se também o tipo da ontologia: ontologia de

alto ou médio nível, de domínio ou de tarefa; ontologia leve ou pesada. Por fim, delimita-se do escopo de cobertura da ontologia descrevendo-se o ponto de partida, o limite do domínio coberto e questões de competência.

A etapa 2 consiste na aquisição e extração de conhecimento, que engloba as atividades de seleção dos materiais de referência do domínio e a adoção de métodos de extração do conhecimento. Na OntoForInfoScience, essas atividades são conduzidas de forma a combinar diferentes métodos de extração, como: análise textual e manual de textos, extração terminológica automática, método semiautomático de identificação de conceitos do domínio. Além disso, busca-se colaboração, isto é, a aquisição de conhecimento envolvendo diferentes visões do domínio a partir de consulta a especialistas diversos. Os artefatos resultantes da etapa 2 são glossários de conceitos, de verbos e de relações, os quais são utilizados como parâmetros de entrada na etapa 3.

Na etapa 3, de conceitualização, são realizadas atividades como identificação e análise dos conceitos do domínio que serão incluídos como classes da ontologia. Além disso, realiza-se a organização e a estruturação do conhecimento de forma a se obter relações, propriedades e restrições da ontologia. Na OntoForInfoScience esse processo ocorre a partir da transformação dos glossários construídos na etapa anterior em artefatos conceituais da etapa 3, os quais correspondem à: i) tabela de conceitos e propriedades, que contém os conceitos identificados e validados, suas definições, seus termos sinônimos, seus valores possíveis e suas propriedades (restrições) a nível conceitual; ii) dicionário de verbos, que contém os verbos identificados e validados como candidatos à relação na ontologia e suas definições textuais; e iii) os modelos conceituais gráficos, que representam as relações conceituais entre os conceitos identificados através de grafos ou estruturas similares.

A etapa 4 corresponde à fundamentação da ontologia desenvolvida, isto é, o desenvolvedor deve pesquisar por ontologias de fundamentação que atuem como ponto de partida e escolher aquela mais apropriada ao objetivo proposto. Ao fazer essa escolha, o desenvolvedor deve considerar a abordagem filosófica subjacente a ontologia de fundamentação, de forma a justificar decisões ontológicas na modelagem do domínio. A ontologia de fundamentação selecionada deve ser importada para um editor de ontologias para início do desenvolvimento ontológico.

Na etapa 5 realiza-se a representação formal da ontologia através de uma linguagem lógica. Nessa etapa produz-se a descrição formal do domínio a partir da conceitualização anterior



(etapa 3). Portanto, o conhecimento do domínio, tratado anteriormente apenas a nível conceitual, passa a ser tratado a nível ontológico-formal, o que implica em uma série de adaptações nas estruturas conceituais de forma a atender as restrições ontológicas e formais. Estão incluídas como atividades: i) construir a taxonomia geral da ontologia, baseada na estrutura da taxonomia da ontologia de fundamentação escolhida; ii) definir propriedades descritivas das classes, as quais envolvem apenas atributos textuais, tais como nomes, sinônimos, definições textuais e anotações; iii) criar definições formais para cada uma das classes da ontologia usando uma linguagem lógica, de forma que a definição formal seja derivada da definição textual elaborada no passo anterior; iv) definir propriedades de dados das classes, as quais envolvem atributos como tipos de dados, cardinalidade e uso dos quantificadores universais ou existenciais; v) criar instâncias das classes da ontologia, de forma a testar o funcionamento da ontologia; vi) especificar relações ontológicas, o que consiste na aplicação de um conjunto definido de regras e princípios para a transição das relações conceituais para o nível formal-ontológico. Dentre essas regras, tem-se como exemplos: a restrição à inclusão de relações a apenas aquelas identificadas como entidades da realidade, a caracterização dos tipos de uma relação ontológica, tais como tipos *is_a* e tipos *part_of*, dentre outras; vii) definir propriedades das relações ontológicas, as quais incluem o nome da relação, a definição semi-formal, a propriedades lógicas básicas, o domínio e a faixa, dentre outras.

A etapa 6 envolve a avaliação da ontologia que, no caso da OntoForInfoScience, corresponde a um conjunto de critérios que permitem realizar tanto a validação ontológica (adequação da ontologia ao domínio – mundo real) quanto a verificação ontológica (análise da ontologia quando a corretude de sua construção). Exemplos de critérios de validação são: natureza ontológica, universalidade de classes e relações e grau de representatividade do domínio. Exemplos de critérios de verificação são: a não recursividade nas definições, a especificação dos diferentes tipos da relação *part_of*, a definição de relações inversas e criação de cardinalidades.

Na etapa 7, de documentação, produz-se o documento formal da ontologia. A elaboração da documentação é realizada, praticamente, ao longo de todo o processo de construção. O conteúdo do documento formal da ontologia engloba o documento de especificação (etapa 1), materiais ou documentos de referência do domínio (etapa 2), conjunto de modelos conceituais (etapa 3), ontologias reutilizadas (etapas 4 e 5), conteúdo ontológico-formal (etapa 5), documento de avaliação da ontologia (etapa 6).

Por fim, a etapa 8 cuida para que a ontologia desenvolvida seja disponibilizada em meio eletrônico de fácil acesso e visualização aos usuários. Como passos desta etapa, tem-se: i) a exportação do conteúdo formal da ontologia (desenvolvido na etapa 5), através de algum editor de ontologias para alguma linguagem lógica, em geral a *Ontology Web Language (OWL)*, para disponibilização na web; ii) a apresentação da ontologia em formato gráfico para mais fácil entendimento pelos usuários, o que pode incluir tanto um documento estático, quanto uma interface de busca ao conteúdo da ontologia.

3 EDITORES DE ONTOLOGIA

A presente seção lista alguns dos principais editores de ontologia com base em referências atualizadas da *World Wide Web Consortium (W3C)* e da *Industrial Ontologies Foundry³ (IOF)* (subseção 3.1); destaca mais detalhadamente o editor de ontologias *Onto4ALLEditor*, através de seus recursos e funcionalidades (subseção 3.2); e apresenta alguns trabalhos que discorrem sobre a utilização e avaliação dos editores por seus usuários e com base critérios específicos (subseção 3.3).

3.1 Uma visão geral sobre editores de ontologia

Essa seção lista, de forma não exaustiva, editores citados em três referências atualizadas (Quadro 2), à saber: (W3C, 2020) (IOF, 2020) (Braun, Estevez e Fillottrani, 2019).

Quadro 2 – Lista de editores de ontologia (não exaustiva, ordem alfabética)

EDITOR	CARACTERÍSTICAS PRINCIPAIS	REFERÊNCIA
<i>CmapTools</i>	Editor de mapeamento conceitual. Permite representar nodos e mapas conceituais em ambiente gráfico.	https://cmap.ihmc.us/cmaptools/
<i>Eddy for Graphol</i>	Editor que usa a linguagem gráfica Graphol. Desenvolve grafos compostos por nós e conexões evitando textos.	https://www.obdasystems.com/eddy
<i>Graffo</i>	Editor de código fonte aberto para a apresentação de classes, propriedades e restrições em OWL.	https://essepuntato.it/graffoo/
<i>Hozo</i>	Editor, servidor e gerenciador de ontologias em ambiente distribuído;	http://www.hozo.jp/

³ Disponível em: <https://www.industrialontologies.org/#> Acesso em 22 de Julho de 2021.



	baseado na Web para colaboração.	
<i>ICOM</i>	Permite manipular múltiplas ontologias; interface gráfica com suporte à digrama de classes e raciocinador.	https://www.inf.unibz.it/~franconi/icom/
<i>Menthor</i>	Editor de código aberto, permite integração com a OntoUML, estereótipos e importação do <i>Ent. Architect</i> .	https://ontouml.org/ontouml/tooling/
<i>NeOn Toolkit</i>	Inclui metodologia para desenvolvimento de ontologias em larga escala em ambiente distribuído.	http://neon-project.org/
<i>OBO-Edit</i>	Editor de código aberto, codificado em Java. Otimizado para o formato ontológico OBO.	http://oboedit.org/
<i>OLED</i>	Editor baseado na linguagem OntoUML, fundamentada na UFO; oferece validação, simulação e detecção de padrões.	https://ontouml.org/ontouml/tooling/
<i>OWLGrEd</i>	Interface gráfica para editar e visualizar ontologias, extensão por plugins inclusive para modelagem UML.	http://owlgred.lumii.lv/
<i>Protégé</i>	Editor de código aberto com interface para criação e edição; extensão por plugins mais processador de consultas.	https://protege.stanford.edu/
<i>SWOOP</i>	Editor de código aberto com suporte a OWL de arquitetura baseada em Web. Suporte à raciocínio.	http://www.mindswap.org/
<i>TopBraid Composer</i>	IDE comercial para modelagem RDF em interface gráfica. Tem capacidade de inferência, e consultas.	https://www.topquadrant.com/products/topbraid-composer/
<i>yEd Graph Editor</i>	Editor múltiplo comercial, permite trabalhar com grafos de redes semânticas e OWL.	https://www.yworks.com/products/yed
<i>WebProtégé</i>	Versão colaborativa na Web do Protégé. Suporta edições em OWL e OBO); importa e exporta formatos.	https://protege.stanford.edu/

Fonte: adaptado de Mendonça et al (2020)



3.2 Onto4ALLEditor

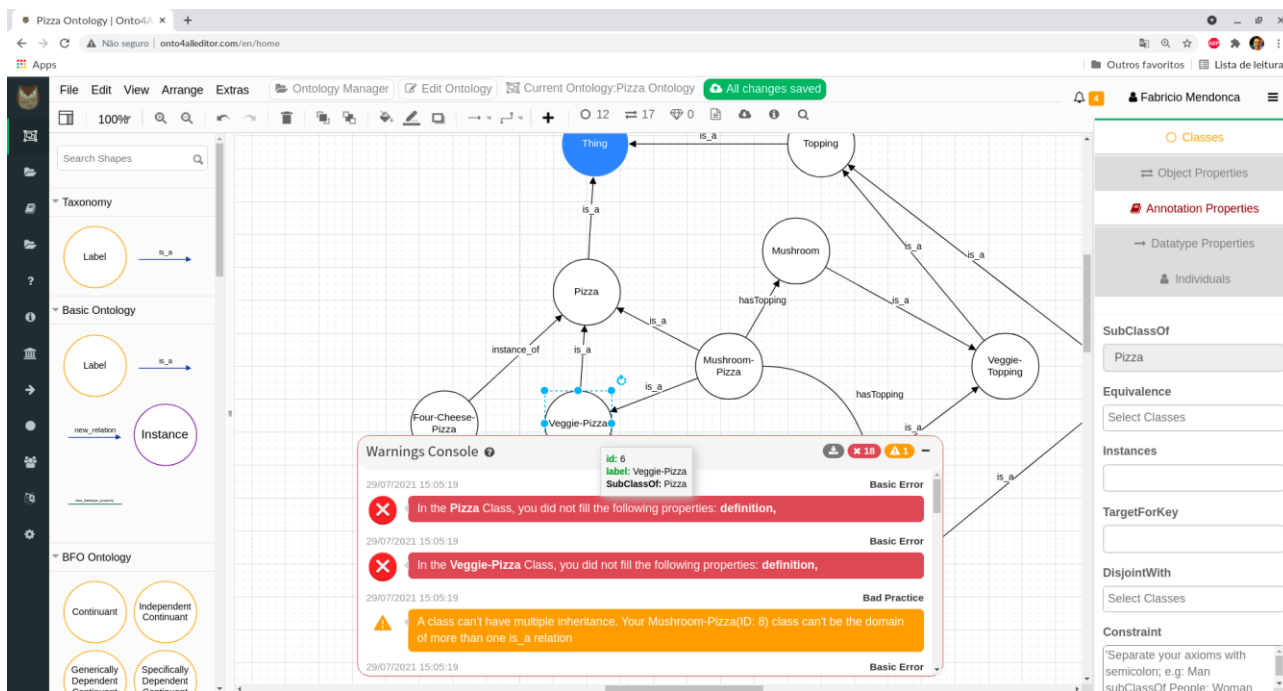
Esta subseção descreve o software Onto4ALLEditor, em sua versão atual 3.0, apresentando suas características, funcionalidades e etapas para criação de ontologias no editor. Como já mencionado, a teoria que fundamenta o Onto4ALLEditor é a metodologia OntoForInfoScience, cuja característica principal é conter uma abordagem detalhada das etapas e tarefas do ciclo de desenvolvimento ontológico, de forma a possibilitar melhor compreensão das atividades necessárias à construção de ontologias para diferentes tipos de desenvolvedores, dos menos aos mais experientes (Mendonça e Soares, 2017). Esse também é um dos propósitos do Onto4ALLEditor, que pode ser usado em conjunto com tal metodologia, outra metodologia ou nenhuma, já que é um software independente.

O diferencial do Onto4ALLEditor em relação aos demais editores avaliados corresponde às facilidades oferecidas no desenvolvimento por usuários sem experiência. Nesse sentido, foram incorporadas as seguintes funcionalidades principais: a) interface gráfica e colaborativa, disponível na web; b) suporte metodológico para tarefas executadas no editor; c) reuso de ontologias de alto nível, além de classes e relações individuais; d) console de avisos e erros ontológicos exibido em tempo real na ferramenta; e) relatórios da ontologia; f) editor de tesouros; dentre outros recursos.

A tela principal do editor é intuitiva e inclui uma área de desenho gráfico para modelagem da ontologia no ambiente web (vide Figura 2). As funcionalidades do Onto4ALLEditor são acessadas através de menu dinâmico, à esquerda da tela, com um ícone específico para cada uma. Também à esquerda da interface está a paleta de elementos gráficos (classes, relações e instâncias), a qual foi separada em 3 (três) grupos: a) *Taxonomy*: elementos para construção de taxonomias – classes nomeadas por “label” e relação “is_a”; b) *Basic Ontology*: elementos básicos para construção de ontologias – classes (círculo), instâncias (losango) e relações “is_a” e “new_relation” (para relações do usuário); c) *BFO Ontology*: constructos (elementos) para reuso de classes e relações das ontologias de alto nível Basic Formal Ontology e a Relation Ontology.



Figura 2 – Ambiente de modelagem gráfica do Onto4ALLEditor

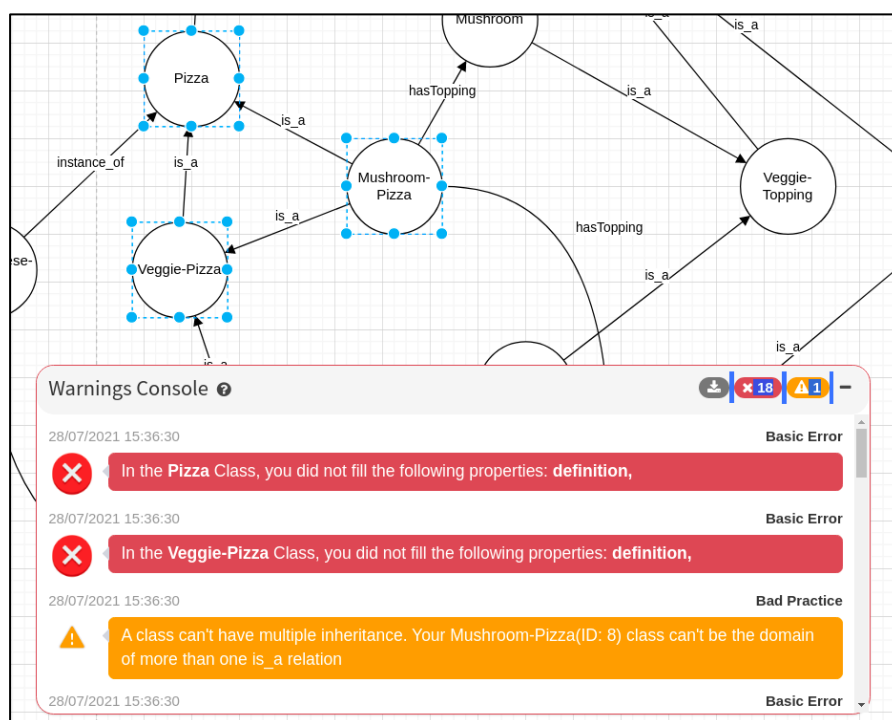


Fonte: Elaborado pelos autores.

Ainda na Figura 2, a paleta à direita inclui os elementos textuais (propriedades) necessários para construção de ontologias separados em 5 (cinco) grupos: i) **Classes**: representam propriedades das classes, tais como: *subClassOf*, *equivalence*, *instances*, *targetForKey*, *disjointWith* e *Constraint* (local do editor usado para inserção e validação dos axiomas da ontologia); ii) **Object Properties**: representam as propriedades das relações da ontologia, tais como: *domain e range* (preenchidos automaticamente pelo editor), *equivalentTo*, *subpropertyOf*, *inverseOf*, *disjointWith*; iii) **Annotation Properties**: são propriedades de anotação de uma classe ou relação da ontologia, tais como: *label*, *comments*, *isDefinedBy*, *seeAlso*, *incompatibleWith*; iv) **Datatype Properties**: representam as propriedades dos tipos de dados da ontologia, tais como: *value*, *domain*, *range*, *equivalentTo*, *subpropertyOf*; v) **Individuals**: são as propriedades das instâncias da ontologia, tais como: *types*, *sameAs*, *differentAs*, *objectProperties*, *dataProperties*. É importante destacar que cada grupo de propriedades é habilitado de acordo com o elemento gráfico selecionado na área de desenho. Assim, por exemplo, na Figura 2 estão habilitados os grupos *Classes* (laranja) e *Annotation Properties* (vermelho), porque a classe “Veggie-Pizza” está selecionada, e desabilitados (em cinza escuro) os outros 3 (três) grupos de propriedades).

Um dos recursos mais relevantes implementados no Onto4ALLEditor é seu console de avisos e erros de modelagem ontológica em tempo real. Entre os avisos e erros já disponibilizados na ferramenta estão: a) classes duplicadas; b) uso incorreto da relação “instance_of”; c) herança múltipla; d) circularidade; e) uso indevido de relações inversas; f) falta de anotações e metadados; g) erros no uso do ambiente gráfico. A Figura 3, a seguir, destaca o console de avisos e erros do editor, os quais não são de correção obrigatória por parte do usuário, mas o alerta sobre eles, e também podem ser exportados para arquivo-texto.

Figura 3 – Console de avisos e erros do editor

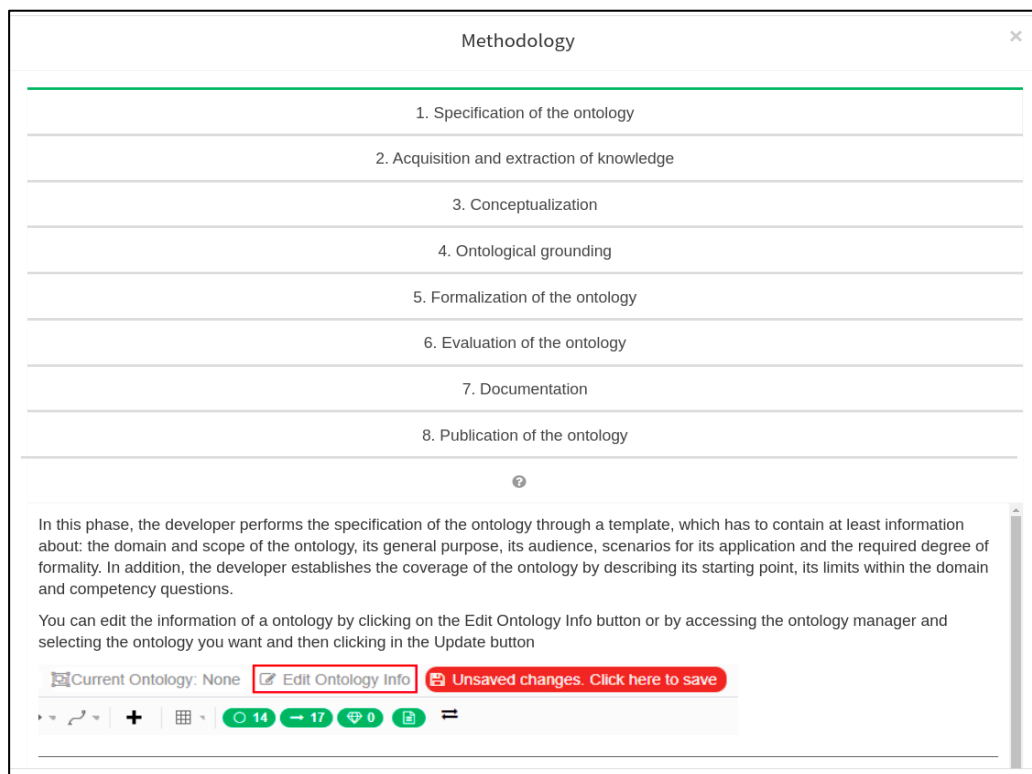


A Figura 3 ilustra avisos no console para uma ontologia de testes, sobre pizzas, onde o usuário recebeu três avisos (em destaque): a) herança múltipla da classe Mushroom-Pizza, que é simultaneamente subclasse de Pizza e de Veggie-Pizza; b) e c) não preenchimento da propriedade obrigatória “definition” para as classes Pizza e Veggie-Pizza. Em ambos os casos, o console detecta dinamicamente os problemas, informando o horário, a classe ou relação com erro, além de emitir um identificador do aviso.

Sobre o **suporte metodológico**, o Onto4ALLEditor é capaz de realizar sete etapas das oito previstas na OntoForInfoScience, a saber: 1) Especificação; 3) Conceitualização; 4) Fundamentação; 5) Formalização; 6) Avaliação; 7) Documentação; 8) Publicação. A segunda etapa da metodologia (Aquisição e Extração do Conhecimento) é a única não possível através

do editor, uma vez que a aquisição do conhecimento do domínio a ser modelado é feita de maneira independente pelos desenvolvedores da ontologia. De forma a orientar o desenvolvedor, as etapas da metodologia são exibidas em uma caixa de diálogo, contendo a descrição da etapa e as tarefas a executar (vide Figura 4). A Figura 4 ilustra a primeira etapa – especificação da ontologia – e como realizá-la a partir de um *template* de especificação.

Figura 4 – Etapas da OntoForInfoScience no software editor



Outras duas funcionalidades que merecem ser destacadas do editor são: i) o processo de armazenamento e documentação da ontologia, realizado através de um repositório específico para o usuário gerenciar suas próprias ontologias, além de poder exportar o conteúdo da ontologia para um arquivo-texto como parte da documentação; ii) a exportação da ontologia modelada para um dos três formatos atuais disponíveis no editor: imagem (.SVG), texto semi-estruturado (.XML) e lógica (.OWL/XML ou .OWX). A exportação para lógica permite que a ontologia possa ser manuseada em outros editores, por exemplo, no popular Protégé.

Por fim, quanto às tecnologias utilizadas, o Onto4ALLEditor foi desenvolvido no framework Laravel versão 6.x, utilizando arquitetura Model View Controller (MVC) e como linguagens de programação PHP (back-end), HTML, CSS e Javascript (front-end) com o pacote Laravel-AdminLTE, e banco de dados MySQL. Em JavaScript, adotaram-se duas bibliotecas específicas:

i) o mxGraph3, modificada para permitir a criação de um componente de diagramação interativo; ii) a API JQuery, utilizada nas funcionalidades dinâmicas do editor, tais como a geração de alertas. O gerenciamento e controle de versão do código-fonte é feito através do GitHub.

3.3 Sobre a utilização, usabilidade e avaliação dos editores

Algumas pesquisas na literatura da área abordam aspectos interessantes sobre a utilização dos editores de ontologia citados, as experiências dos usuários (*User eXperience* – UX) com tais ferramentas, considerando aspectos de usabilidade, e avaliações dos editores sob um conjunto de critérios estabelecidos. Esta subseção do artigo destaca esses aspectos, apresentando tais pesquisas.

Uma pesquisa quantitativa com 65 profissionais que trabalham com ontologias identificou, dentre outras questões, quais os **editores eram os mais usados** (Warren, 2013). O Protégé foi identificado como o mais utilizado com 50% dos entrevistados; o TopBraid Composer, um editor comercial, apareceu em segundo lugar com 14%; o NeOn Toolkit (Michael e Waterfeld, 2012), CmapTools (Cañas et al, 2004) e SWOOP (Kalyanpur et al, 2006) apareceram, respectivamente, com 6%, 5% e 4%; além dos editores voltados para Biomedicina, o OBO Edit (Day-Richter, Harris e Haendel, 2007) com 3%.

Outra pesquisa sobre os aspectos da **usabilidade** de editores (Malik, 2017) apresentou aos participantes uma sequência de tarefas em cinco editores. Em seguida, aplicou-se aos participantes um questionário de usabilidade. Dentre os resultados, o Protégé apareceu como o mais fácil de usar e que exigiu menos tempo para as tarefas; o IHMC CmapTool apresentou os piores resultados. O tempo para as tarefas com o Protégé foi cerca da metade em comparação com outros editores como TopBraid, NeOn e SWOOP.

Outro trabalho conduziu uma entrevista com cerca de 30 profissionais de ontologias, questionando qual o melhor editor para iniciantes (Siricharoen, 2018). O Protégé foi o indicado como o mais adequado e, dentre os outros, apenas o TopBraid Composer teve mais de uma menção. Os resultados mostraram que cerca de 70% dos profissionais recomendam o Protégé e apenas 10% indicaram o TopBraid Composer. As demais ferramentas envolvidas foram NeOn Toolkit, SWOOP, OntoStudio. Um dos aspectos mencionados por aqueles que escolheram o Protégé é a existência de um tutorial de uma ontologia de pizza mostrando o uso de tal editor.

Outros motivos pela escolha do Protégé incluem sua grande comunidade de usuários, o fato de ser gratuito e ter código aberto.

Outro aspecto relevante sobre editores de ontologia, abordado na presente pesquisa, refere-se a uma avaliação geral de tais softwares quanto aos recursos e funcionalidades disponíveis aos seus usuários no processo de construção de ontologias. Com base em trabalhos sobre o tema na literatura da área ((Alatrish, 2013) (Vigo et al, 2014) (Warren, 2013) (Malik, 2017) (Siricharoen, 2018) (Mendonça et al., 2021)) foi possível apontar as principais limitações dos softwares editores atualmente disponíveis. São elas:

- Problemas com a visualização gráfica de ontologias e com a navegação pouco amigável nos editores que oferecem tal recurso. Poucas ferramentas permitem que o usuário interaja dinamicamente com o modelo gráfico da ontologia, especificando classes, relações e propriedades;
- Falta de suporte para visualização de ontologias em ambientes preferidos dos usuários;
- Interface pouco amigável na navegação pela hierarquia de classes;
- Necessidade de um ambiente mais colaborativo entre especialistas do domínio para modelagem da ontologia. Um fator que prejudica a colaboração no desenvolvimento é o fato de que muitos editores de ontologia não estão disponíveis na web;
- Falta de suporte metodológico no desenvolvimento da ontologia dentro do editor, por exemplo, com ações que orientem os usuários nas tarefas de construção;
- O tempo gasto para executar tarefas de construção da ontologia em alguns editores é alto;
- Feedback ruim na depuração de erros da ontologia;
- Falta de suporte para pesquisa de termos em ontologias externas para reuso;
- Existe uma grande variedade de formatos de importação e exportação do conteúdo da ontologia para linguagens formais, o que indica uma ausência de padronização nesses formatos que pode prejudicar a interoperabilidade semântica;
- Falta de suporte a reasoner em alguns editores;
- A extensibilidade dos editores é assegurada, em grande parte apenas, pela inclusão de plug-ins ao editor principal;
- Muitos editores de ontologias foram descontinuados ou já não estão mais disponíveis para acesso por seus usuários;
- Editores de ontologia comerciais, em geral, cobram um valor alto pela licença do software;

- Falta de suporte à inserção de axiomas;
- Não há um padrão para armazenamento das ontologias construídas nos editores, que podem ser armazenadas em arquivos simples, formato HTML ou em bancos de dados relacionais;
- Falta de suporte e gerenciamento de backup.

Essa análise mais geral das limitações dos editores de ontologia é mais detalhada e explicada no trabalho de Mendonça et al. (2021), no qual os autores conduziram uma avaliação individual de cada editor de ontologia explorado, tendo como base um conjunto de critérios básicos do processo de construção de ontologias. Entre tais critérios e sua avaliação individual em cada editor analisado (CmapTools, Eddy for Graphol, Graffo, Hozo, ICOM, Menthor, NeOn Toolkit, OBO-Edit, OLED, Onto4ALLEditor, OWLGrEd, Protégé, SWOOP, TopBraid Composer, yEd Graph Editor, WebProtégé) pode-se destacar os seguintes:

1. **Visão gráfica da ontologia:** todos editores analisados fornecem tal recurso.
2. **Editor disponível on-line na web:** dentre os editores analisados apenas yEd Graph Editor, WebProtégé e Onto4ALLEditor estão disponíveis para uso na web.
3. **Suporte metodológico às etapas de construção da ontologia:** esse critério refere-se ao uso de uma metodologia de construção dentro do software editor. Nenhum dos editores avaliados oferece tal recurso, com exceção do Onto4ALLEditor através da OntoForInfoScience. Ainda assim, o suporte fornecido por este editor precisa ser melhor vinculado e detalhando à metodologia usada.
4. **Importação e exportação de ontologias para linguagens formais:** muitos formatos diferentes estão disponíveis para importação e exportação de ontologias em tais editores, como por exemplo: RDF, XML, OWL, OBO, TTL, UML, HTML, XMI, dos quais alguns não são formalizações de ontologia, como HTML, XML e UML.
5. **Suporte à inserção de axiomas:** tal recurso está presente na maior parte dos editores. São eles: Eddy for Graphol, Neon Toolkit, OBO-Edit, OLED, Onto4ALLEditor, OWLGrEd, Protégé, SWOOP, TopBraid Composer e WebProtégé.
6. **Suporte a *reasoners*:** tal funcionalidade está presente em cerca de metade (ou sete) editores analisados: Eddy for Graphol, ICOM, Neon Toolkit, OLED, Protégé, SWOOP e TopBraid Composer.
7. **Tipo de ambiente (desktop, web ou outro) de funcionamento do editor:** neste critério avaliativo a grande parte dos editores estão disponíveis apenas para instalação e execução via desktop, exceto os editores Onto4ALLEditor e WebProtégé – disponíveis somente em ambiente web – e yEd Graph Editor e TopBraid Composer – disponíveis tanto em ambiente desktop quanto web.

8. **Armazenamento das ontologias:** nesse item grande parte dos editores armazena as ontologias construídas em um arquivo local da máquina do usuário, porque em sua maioria funcionam em ambiente desktop. Os editores CMapsTools, yEd Graph Editor e WebProtégé armazenam as ontologias no servidor com opção de download para máquina do usuário, enquanto os editores Onto4ALLEditor e TopBraid Composer fornecem recurso de armazenamento das ontologias em banco de dados, além de arquivo local.
9. **Suporte e gerenciamento de backup das ontologias:** esse é um recurso que a maioria dos editores não disponibiliza aos usuários. De certa forma, os editores Onto4ALLEditor e TopBraid Composer fornecem tal recurso ao permitir que seus usuários salvem as ontologias em bancos de dados usados em tais ferramentas;
10. **Plataformas e sistemas operacionais:** quase todos editores analisados estão disponíveis para Windows, Linux e MacOS, com exceção dos editores Hozo (apenas Windows e MacOS) e OWLGrEd (apenas Windows);
11. **Status atual de funcionamento:** corresponde ao fato do editor estar em funcionamento ou não na atualidade, uma vez que alguns editores de ontologia foram descontinuados e não estão mais disponíveis. Dentre esses indisponíveis atualmente tem-se: ICOM, Neon Toolkit, OBO-Edit e SWOOP. Os demais editores estão disponíveis para uso.

4 DISCUSSÃO

Como mencionado anteriormente, o uso de editores de ontologia ainda é um desafio em Ciência da Informação e entre especialistas do domínio, mesmo com um número significativo de editores disponíveis atualmente e a utilização dos editores mais populares, como o Protégé, ocorrer há mais de duas décadas na área. Esta seção do artigo apresenta uma breve discussão sobre as possibilidades de uso do Onto4ALLEditor e sua metodologia associada – OntoForInfoScience – entre usuários da informação (profissionais, pesquisadores e estudantes da área) e especialistas do domínio.

Um resumo das avaliações apresentadas por usuários dos editores de ontologia e encontrada na literatura da área, conforme descrito na seção 3, aponta entre as limitações: i) necessidade de um ambiente mais colaborativo entre especialistas do domínio para modelagem da ontologia; ii) falta de suporte metodológico na construção das ontologias; iii) falta de suporte para visualização de ontologias em ambientes preferidos dos usuários; iv) feedback ruim na depuração de erros da ontologia; v) ausência de recursos para backup e gerenciamento das ontologias desenvolvidas; vi) inexistência de um padrão de linguagem formal para exportação e importação de ontologias; vii) falta de suporte para pesquisa de termos em ontologias externas para reuso; entre outras.



Diante do cenário apresentado, iniciativas e projetos que tem por objetivo permitir acesso ao tema e mesmo disseminar a construção de ontologias na área tornam-se relevantes e podem representar um passo importante na popularização do uso de ontologias em Ciência da Informação e em outras áreas não tão familiarizadas com conceitos técnicos de modelagem conceitual e lógica formal. Esse é o propósito principal da abordagem ontológica que inclui a metodologia OntoForInfoScience e o software Onto4ALLEditor.

Publicada em 2015, a OntoForInfoScience surgiu da necessidade de uma metodologia mais detalhada das atividades envolvidas na construção de ontologias direcionada, inicialmente, para cientistas da informação, como forma de auxiliá-los a superar problemas relativos ao jargão técnico e às questões lógicas e filosóficas necessários para o desenvolvimento ontológico (Mendonça, 2015). Entretanto, percebeu-se que tal metodologia pode ser útil não apenas na área de Ciência da Informação, mas também a qualquer desenvolvedor de ontologia, incluindo especialistas do domínio, linguistas e mesmo cientistas da computação, que precisam adquirir maior conhecimento nas questões técnicas relacionadas à construção de ontologias.

Os resultados práticos da OntoForInfoScience ainda precisam ser melhor avaliados do ponto de vista de ontologias construídas a partir de tal metodologia e se de fato o propósito principal de disseminar e popularizar o conhecimento no processo de construção de ontologias é atingido. Ainda assim, é possível encontrar exemplos na literatura da área de ontologias e métodos desenvolvidos a partir da OntoForInfoScience:

- *Hemonto*: uma ontologia biomédica de domínio que representa o conhecimento relativo aos componentes do sangue humano (hemocomponentes e hemoderivados) usados para fins terapêuticos (Mendonça e Almeida, 2016) (Mendonça, 2015);
- *Ontolegis*: uma ontologia de domínio jurídico que representa o conhecimento relativo à informação legislativa brasileira sobre o Direito Médico (Torres, 2017);
- Modelagem de ontologia no *domínio do futebol* para fins de recuperação da informação em acervos digitais de jornais impressos (Santos e Bräscher, 2015);
- *Onto4M4IS*: método de reuso ontológico para Ciência da Informação (Firmino e Lima, 2018);
- *OntoMEDTRAD*: uma ontologia da medicina tradicional africana (Kouame, 2019);
- Ontologia de domínio voltada para *cibersegurança* que representa o conhecimento no domínio automotivo e segurança cibernética envolvidos para suportar os desenvolvedores de softwares embarcados no processo de avaliação da cibersegurança de CPS (Rezende, 2020);

- Ontologias para organização do conhecimento no *setor de energia elétrica* do Brasil (Teixeira et al, 2021).

Além desses trabalhos concluídos, existem trabalhos em andamento (ainda não publicados) que utilizam a abordagem OntoForInfoScience e o software Onto4ALLEditor no desenvolvimento de seus projetos ontológicos: i) a *ReVera Ontology*: uma ontologia de tarefa para rastreabilidade de informações quanto à sua veracidade na área de *fact checking*; ii) *Onto Hate*: uma ontologia de domínio para representar e identificar termos em língua portuguesa usados em discursos de ódio no ambiente web; iii) ontologias para representar o conhecimento do *patrimônio cultural relativo ao acervo dos museus brasileiros* integrados ao Instituto Brasileiro de Museus (IBRAM).

Embora o número de projetos ontológicos que utilizam ou utilizaram a OntoForInfoScience não seja tão expressivo quanto às metodologias tradicionais e mais antigas da área, tais como a Methontology e a NeOn, os trabalhos aqui citados mostram um crescimento em sua aplicação nos contextos acadêmico e profissional. Uma característica que tem contribuído para o uso da OntoForInfoScience é seu uso associado ao software Onto4ALLEditor, o qual teve seu desenvolvimento iniciado em 2018 e cuja primeira versão oficial foi disponibilizada em 2020 a usuários finais e publicado no trabalho “Onto4ALLEditor: um Editor Web Gráfico de Ontologias Direcionado a Diferentes Tipos de Desenvolvedores de Ontologias” (Mendonça et al., 2020).

Apesar de ser uma ferramenta ainda em processo de desenvolvimento e evolução, o Onto4ALLEditor inclui recursos e funcionalidades atuais como uma alternativa às limitações encontradas nos editores de ontologias disponíveis. Os parágrafos seguintes destacam características deste editor que fornecem facilidades e suporte à construção de ontologias por usuários da informação, especialistas de domínio e outros usuários menos experientes com o tema, com base nas avaliações dos editores de ontologia, apresentada na Seção 3.

Uma estratégia defendida e praticada no Onto4ALLEditor é o uso de uma interface gráfica para construção da ontologia como primeira atividade nesse processo. Essa característica beneficia usuários com menos experiência e contribui para melhor visualização do modelo ontológico, de forma que o usuário interaja dinamicamente com o modelo gráfico da ontologia. Nesse sentido, não procede comparar o Onto4ALLEditor ao Protégé, cuja abordagem envolve como primeira atividade a elaboração da hierarquia de classes da ontologia e a geração do modelo gráfico ao final do processo com uso de plug-ins da ferramenta. Além disso, é importante destacar que essas 2 (duas) ferramentas encontram-se em estágios de desenvolvimento completamente



diferentes e foram criadas em contextos e com recursos bastante díspares. O desenvolvedor de ontologia pode no seu projeto optar por uma desses dois editores ou outro, ou ainda utilizá-los simultaneamente sem prejuízos ao seu projeto ontológico.

No que se refere à disponibilização de editores de ontologia na web, nota-se que poucas ferramentas oferecem este recurso e algumas que possuem (por exemplo, TopBraid Composer) tem alto custo de licença do software. A modelagem gráfica no ambiente web é uma característica essencial para construção de ontologias por especialistas no domínio, pelo fato de possibilitar o trabalho colaborativo, simultâneo e on-line. Esses recursos são nativos do Onto4ALLEditor.

O suporte metodológico é outra funcionalidade de relevância para usuários menos experientes. Editores com essa característica, rara entre os avaliados, conseguem orientar seus usuários nas tarefas necessárias. Como citado antes, o editor Onto4ALLEditor fornece suporte metodológico com base na OntoForInfoScience.

Outros recursos importantes são o suporte à inserção de axiomas, como como faz o Protégé através de um validador de expressões lógicas em Manchester Syntax, e a inclusão de reasoners para identificação de erros e inconsistências. O Onto4ALLEditor permite, em sua versão atual, a inserção de axiomas, validados através do atributo “constraint” que faz uma requisição a OWL-API a fim de validar as expressões lógicas contidas no axioma. A inclusão de reasoners como plug-ins do Onto4ALLEditor está prevista para as próximas versões da ferramenta, já que eles são abordagens complementares ao console de erros e avisos do editor.

Quanto ao armazenamento de ontologias, a maioria dos editores permite salvá-las em um arquivo local na máquina do usuário. O Onto4ALLEditor dispõe desse tipo de armazenamento, mas também pode salvar o trabalho em um banco de dados no servidor web, recurso que permite o gerenciamento de ontologias desenvolvidas por cada usuário.

Cabe ainda mencionar que alguém pode sugerir como limitação o fato de que os problemas apontados em outros editores, ao longo do texto, são de mais de cinco anos atrás. Contudo, as referências aqui apresentadas são as semanais da literatura e não se identificou nenhum estudo mais recente no processo de revisão. Ainda que o número de plug-ins de algumas ferramentas tenha aumentado nesse período, acredita-se que tal aumento não reflete necessariamente em melhorias no uso das ferramentas por profissionais e pesquisadores da Ciência da Informação.



5 CONSIDERAÇÕES FINAIS

Este artigo apresentou a metodologia OntoForInfoScience e o editor web gráfico de ontologias, Onto4AllEditor, como alternativa para modelagem ontológica, principalmente para não especialistas em ontologias. A metodologia e o editor fazem parte de um projeto em andamento que objetiva popularizar o desenvolvimento de ontologias, fortalecendo assim a pesquisa na área de ontologias na Ciência da Informação.

Nesse estágio, a pesquisa contém limitações, sendo a principal delas uma avaliação de usabilidade mais robusta que respalde o que a prática já tem mostrado. Nesse sentido, o editor tem sido utilizado por professores e alunos de graduação, mestrado e doutorado na Universidade Federal de Minas Gerais (UFMG) e na Universidade Federal de Juiz de Fora (UFJF). Além disso, o Onto4ALLEditor tem sido utilizado em projetos ontológicos acadêmicos e comerciais em andamento, por exemplo: i) ontologias no domínio de energia elétrica do Brasil; ii) ontologias no domínio de patrimônio cultural relativo ao acervo dos museus brasileiros do IBRAM; iii) ontologia de tarefa para rastreabilidade de informações quanto à sua veracidade na área de fact checking; e iv) ontologia de domínio para termos em língua portuguesa sobre discursos de ódio no ambiente web. Espera-se, em breve, um retorno empírico de tais projetos no sentido de respaldar as aplicações da metodologia e editor.

Paralelamente a realização dos projetos ontológicos mencionados, trabalhos futuros estão previstos para a metodologia e o editor, aqui apresentados. Será gerada uma nova versão (2.0) da OntoForInfoScience, que prevê uma simplificação das etapas e atividades incluídas na metodologia, além de uma associação mais forte dessas etapas com sua realização no Onto4ALLEditor - desenvolvido após a metodologia. Quanto ao editor, novos recursos e funcionalidades estão sendo adicionados: i) importação e exportação de ontologias para os seguintes formatos: RDF/XML, OWL 2.0, Turtle, Manchester Syntax. Atualmente o formato suportado no editor é o OWL/XML; ii) edição da ontologia em tempo real por 2 ou mais usuários - na versão atual uma ontologia pode ser compartilhada por 2 ou mais usuários, mas apenas 1 (um) pode editá-la em tempo real; iii) inclusão de uma ferramenta de *chat* entre usuários que compartilham a mesma ontologia, para que possam discutir questões da modelagem ontológica em tempo real; iv) criação de um tutorial passo a passo de utilização do editor por seus usuários; v) aplicação de questionários de avaliação da usabilidade e experiência do usuário com o editor.

Por fim, a metodologia OntoForInfoScience e o software Onto4ALLEditor se justificam como forma de permitir acesso ao tema e mesmo disseminar a construção de ontologias na área. Também se justificam como uma abordagem mais simples e detalhada para a construção de ontologias entre desenvolvedores menos experientes e especialistas de domínio, e também no âmbito de áreas científicas que não englobam ou não privilegiam o ensino de modelagem conceitual e lógica formal, tais como a Ciência da Informação e a Linguística.

REFERÊNCIAS

- ALATRISH, E. Comparison Some of Ontology Editors. *Management Information Systems*, [s. 1], Vol. 8, No. 2, pp. 018-024, 2013.
- ALMEIDA, M.B. Revisiting ontologies: A necessary clarification. *Journal of the American Society for Information Science and Technology*, [s. 1], v. 64, n. 8, p. 1682-93, 2013.
- BERNARAS, A., LARESGOITI, I., & CORERA, J. Building and Reusing Ontologies for Electrical Network Applications. In *European Conference on Artificial Intelligence – ECAI 1996*, p. 298-302.
- BRAUN G., ESTEVEZ E. and FILLOTRANI, P. A Reference Architecture for Ontology Engineering Web Environments. *Journal of Computer Science & Technology*, vol. 17, no. 2, pp. 1-3, 2018.
- CAÑAS, A. J., HILL, G., CARFF, R., SURI, LOTT, J., GÓMEZ, G., ESKRIDGE, T. C., ARROYO, M., CARVAJAL, R. CMapTools: a Knowledge Modeling and Sharing Environment. In *First Int. Conference on Concept Mapping*, Spain, 2004.
- DAY-RICHTER J., HARRIS M. A., HAENDEL M. The Gene Ontology OBO-Edit Working Group, Suzanna Lewis, OBO-Edit—an ontology editor for biologists. In *Bioinformatics*, Volume 23, Issue 16, 15 August 2007, Pages 2198–2200, 2007.
- DE NICOLA, A.; MISSIKOFF, M.; e NAVIGLI, R. A software engineering approach to ontology building. *Information Systems* 34, p. 258–275, 2009.
- FERNÁNDEZ, M. et al. Building a chemical ontology using methontology and the ontology design environment. *Intelligent Systems*, v. 14, n. 1, p. 37-46, Jan./Feb.1999.
- FERNÁNDEZ, M.; CORCHO, O. Methodologies and methods for building ontologies. In: FERNÁNDEZ, M.; CORCHO, O. *Ontological engineering*. London: Springer, 2004. p. 107-153.
- FIRMINO, H. N. M.; LIMA, G. N. B. O. Ontom4is+: ontology reuse method for information science. *Pesquisa Brasileira em Ciência da Informação e Biblioteconomia*, v. 13, n. 2, 2018. Acesso em: 28 jul. 2021.



GÓMEZ-PÉREZ, A.; FERNÁNDEZ, M.; VICENTE, A. J. Towards a method to conceptualize domain ontologies. In: *European Conference on Artificial Intelligence – ECAI*, 1996, Budapest, Hungary.

GRÜBER, T. (1992). What is an Ontology? Disponível em: <http://www.ksl.stanford.edu/kst/what-is-an-ontology.html> Acesso: 20 maio de 2020.

GRUNINGER, M.; FOX, M. S. “Methodology for the Design and Evaluation of Ontologies”. In: *Workshop on Basic Ontological Issues in Knowledge Sharing (IJCAI)* [S.l.; s.n.], 1995.

GUARINO, N. Formal Ontology in Information Systems (FOIS), Trento, Itália, 1998. Disponível em: <http://citeseer.ist.psu.edu/guarino98formal.html>. Acesso em: 2 jul. 2020.

GUERSON, J.; SALES, T. P.; GUIZZARDI, G., et al. (2015). “OntoUML Lightweight Editor: A Model-Based Environment to Build, Evaluate and Implement Reference Ontologies,” In: *IEEE 19th International Enterprise Distributed Object Computing Workshop*, pp. 144-147, 2015.

INDUSTRIAL ONTOLOGIES FOUNDRY (IOF). *Helpful materials on ontologies*, 2020. Disponível em: <https://www.industrialontologies.org/>. Acesso em: 20 março 2020.

JONES, D.; BENCH-CAPON, T.; VISSER, P. Methodologies for ontology development. 1998. Disponível em: <http://cweb.inria.fr/Resources/ONTOLOGIES/methodo-for-ontodev.pdf>. Acesso em: 13 de março de 2014.

KALYANPUR, A., PARSIA B., SIRIN E., GRAU B. C.; HENDLER, J. Swoop: A web ontology editing browser. *Web Semantics: Science, Services and Agents on the World Wide Web*, [s. 1], n. 4.2, pp. 144-153, 2006.

KOUAME, A. Système de gestion de la médecine traditionnelle dans une plateforme web social et sémantique : une approche basée sur une ontologie visuelle. Intelligence artificielle [cs.AI]. Université Gaston Berger de Saint-Louis (Sénégal), 2018. Français.

LIM, S. C. J.; LIU, Y.; e LEE, W. B. A methodology for building a semantically annotated multi-faceted ontology for product family modelling. *Advanced Engineering Informatics* 25 (2), pp. 147-161, 2011.

MALIK, Z. H. (2017). “Usability Evaluation of Ontology Engineering Tools”, In: *Computing Conference 2017*, London, UK.

MEALY, G. H. (1967). Another Look at Data. Proc. of the Joint Computer Conf., pp. 525-534.

MENDONÇA, F. M. OntoForInfoScience: metodologia para construção de ontologias pelos cientistas da informação – uma aplicação prática no desenvolvimento da ontologia sobre componentes do sangue humano (Hemonto). Tese de Doutorado. Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, Brasil, 2015.

MENDONÇA, F. M.; ALMEIDA, M. B. “OntoForInfoScience: a detailed methodology for construction of ontologies and its application in the blood domain”. In *Brazilian Journal of Information Science*, v. 10, p. 1., 2016.



MENDONÇA, F. M.; SOARES, A. L. Construindo ontologias com a metodologia OntoForInfoScience: uma abordagem detalhada das atividades do desenvolvimento ontológico In: *Ciência da Informação*, v. 46, p. 43-59, 2017.

MENDONÇA, F. M.; CASTRO, L. P.; SOUZA, JAIRO; ALMEIDA, M. B.; FELIPE, E. R. Onto4AllEditor: Um Editor Web Gráfico de Ontologias Direcionado a Diferentes Tipos de Desenvolvedores de Ontologias. In: *XIII Seminar on Ontology Research in Brazil (ONTOBRAS 2020)*, 2020, Vitória - ES. Proceedings of the XIII Seminar on Ontology Research in Brazil. , 2020. v.1. p.104 – 119.

MENDONÇA, F. M.; CASTRO, L. P.; SOUZA, J. F.; ALMEIDA, M. B.; FELIPE, E. R. Onto4AllEditor: um editor web gráfico para construção de ontologias por todos tipos de usuários da informação. In: *Em Questão*, Porto Alegre, v. 27, n. 3, p. 401-430, jul./set. 2021.

MICHAEL, E.; WATERFELD, W. Overview of the Neon Toolkit. *Ontology Engineering in a Networked World*, [s. 1], pp. 281-301, 2012.

MUSEN, M.A. The Protégé project: A look back and a look forward. *AI Matters. Association of Computing Machinery SIG-AI*, [s. 1], v. 1, n. 4, 2015.

NOY, N. F.; McGUINNESS, D. L. Ontology Development 101: A Guide to Creating Your First Ontology. *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880*, California, USA, 2001.

REED, S.L.; LENAT, D.B. Mapping Ontologies into Cyc, 2002. Disponível em: http://www.cyc.com/doc/white_papers/mapping-ontologies-into-cyc_v31.pdf. Acesso em: 25 de abril de 2012.

REZENDE, M. V. Avaliação de segurança cibernética no desenvolvimento de software embarcado automotivo: uma abordagem ontológica. Tese de Doutorado em Sistemas de Informação e Gestão do Conhecimento. Universidade FUMEC, Belo Horizonte, Brasil, 2020.

SANTOS, M. T. Estudo do processo de apropriação da ontologia pela Ciência da Informação no Brasil. Dissertação (Mestrado), Universidade Federal de Pernambuco, Recife, PE, Brasil, 2014.

SCHALLEY, A.C. Ontologies and ontological methods in linguistics. In: *Lang Linguist Compass*. 2019. 13:e12356.

SILVA, D. L.; SOUZA, R. R.; e ALMEIDA, M. B. Ontologias e vocabulários controlados: comparação de metodologias para construção. In: *Ciência da Informação*, Brasília, v. 37, n. 3, p. 60-75, set./dez. 2008.

SIRICHAROEN, W. V. Ontology Editors Approach for Ontology Engineering. *International Conference on Control, Automation and Robotics*, [s. 1.], 2018.

SMITH, B. Ontology and Informations Systems, 2004. Disponível em: <http://www.ontology.buffalo.edu/ontology>. Acesso em: 25 jan. 2016.

Smith, B.; Welty, C. Ontology: Towards a new synthesis. (2001). Disponível em: <http://www.cs.vassar.edu/~weltyc/papers/foisintro.pdf>. Acesso em: 20 dez. 2010.



SUÁREZ-FIGUEROA, M. C. (2008). NeOn D 5.4.1. NeOn Methodology for Building Contextualized Ontology Networks. NeOn project. Available in: <http://www.neon-project.org>. Acesso em: 06 de Abril de 2020.

SURE, Y.; STAAB, S.; e STUBER, R. On-To-Knowledge Methodology (OTKM). 2003. Disponível em: http://www.sfu.ca/~mhatala/iat881/papers/2003_ontohandbook_otkmethodology.pdf.

SWARTOUT, B. et al. Toward distributed use of large-scale ontologies. 1996. Disponível em: http://ksi.cpsc.ucalgary.ca/KAW/KAW96/swartout/Banff_96_final_2.html. Acesso em: 13 de março de 2014.

TEIXEIRA, L. M. D.; MODESTO, M. L.; EMYGDIO, J. L.; MOREIRA, C.; ALMEIDA, M. B. Conhecimento especializado no domínio da energia elétrica: estudo de caso sobre a aquisição do conhecimento para ontologia. *Encontros Bibli: Revista Eletrônica de Biblioteconomia e Ciência da Informação*, v. 26, p. 1-28, 2021.

Top-Quadrant (2020). “TopBraid Composer”. Disponível em: <https://www.topquadrant.com/products/topbraid-composer/>. Acesso em: 6 de Abril de 2020.

TORRES, S. Modelagem de domínios em Sistemas de Organização do Conhecimento (SOC): uma investigação em tesouros e ontologias para a informação legislativa. 2017. Tese (Doutorado em Ciência da Informação). Universidade Federal de Minas Gerais, Belo Horizonte, 2017.

USCHOLD, M.; KING, M. Towards a methodology for building ontologies. 1995. Disponível em: <http://citeseer.ist.psu.edu/uschold95toward.html> Acesso em: 11 mar. 2014.

VIGO, M.; BAIL, S.; JAY, C. STEVENS, R. Overcoming the pitfalls of ontology authoring: Strategies and implications for tool design. *International Journal of Human-Computer Studies*, [s. 1.], v. 72, n. 12, p. 835-845, 2014.

W3C 2020. Ontology editors. Available in: https://www.w3.org/wiki/Ontology_editors. Acesso em: 06 de Abril de 2020.

WAND, Y.; STOREY, V. C.; WEBER, R. (1999). An ontological analysis of the relationship construct in conceptual modeling. *ACM Transactions on Database Systems*, New York, v. 24, n. 4, p. 494-528, 1999.

WARREN, P. Ontology Users' Survey – Summary of Results. *The Knowledge Media Institute (KMi)*, [s. 1.], 2013.

