

**UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

Lucas Piazzini de Castro

Onto4ALL: Editor gráfico de ontologias

Juiz de Fora
2022

Lucas Piazzi de Castro

Onto4ALL: Editor gráfico de ontologias

Monografia apresentada ao curso de Ciência da Computação, do Instituto de Ciências Exatas, na Universidade Federal de Juiz de Fora, como requisito parcial para obtenção do grau de bacharel.

Orientador: Prof. Dr. Fabrício Martins Mendonça

Juiz de Fora

2022

Lucas Piazzi de Castro

Onto4ALL: Editor gráfico de ontologias

Monografia apresentada ao curso de Ciência da Computação, do Instituto de Ciências Exatas, na Universidade Federal de Juiz de Fora, como requisito parcial para obtenção do grau de bacharel.

Aprovada em 11 de agosto de 2022

BANCA EXAMINADORA

Prof. Dr. Fabrício Martins Mendonça (ICE/UFJF) - Orientador

Prof. Dr. Victor Stroele (ICE/UFJF) – Convidado

Profa. Dra. Regina Braga (ICE/UFJF) – Convidado

Prof. Dr. Jairo Francisco de Souza (ICE/UFJF) - Convidado

AGRADECIMENTOS

Aos meus pais, Edmilson e Adriana e a minha avó Cleia, pelo sustento, amor, carinho e paciência. Aos meus amigos e colegas de trabalho, pelo incondicional apoio e ajuda. Ao professor Fabrício pela orientação e amizade. Aos colegas de iniciação científica, empresa júnior e faculdade que contribuíram muito para o desenvolvimento do projeto. A minha namorada pelo companheirismo. Aos professores do Departamento de Ciência da Computação e Departamento de Matemática pelos conhecimentos passados. Aos funcionários da UFJF que me proporcionaram um ambiente de muito aprendizado e conhecimento. Agradeço também a todos os usuários do Stackoverflow e Github com quem tive contato, tanto ensinando quanto, principalmente, aprendendo algo sobre as tecnologias aqui utilizadas. Muito obrigado!

"Ninguém baterá tão forte quanto a vida. Porém, não se trata de quão forte pode bater, se trata de quão forte pode ser atingido e continuar seguindo em frente. É assim que a vitória é conquistada." (Rocky: Um lutador, 1976)

RESUMO

O uso e aplicações de ontologias têm crescido, notoriamente nos últimos anos, em Ciência da Computação e áreas afins, muito em razão da necessidade de se tratar semanticamente o enorme volume de dados na web, em ambientes empresariais e acadêmicos. Dentre as muitas aplicações de ontologias atualmente destacam-se: a representação formal de diferentes domínios do conhecimento, integração de bases de dados heterogêneas promovendo interoperabilidade, organização de dados e processos de negócio, disponibilização de dados nos formatos requeridos para web semântica. Considerando o processo de construção de ontologias por seus desenvolvedores, é possível notar que a criação de ontologias é um processo longo e oneroso, que necessita de conhecimento técnicos em modelagem conceitual, em lógica e aquisição de conhecimento, além de uma metodologia e um *software* editor de ontologias. Neste cenário, este trabalho apresenta o editor de ontologias Onto4ALL, o qual foi concebido tendo como base de fundamentação teórica a metodologia de construção OntoForInfoScience. O Onto4ALL tem como objetivo principal facilitar a criação e manutenção de ontologias através de uma interface gráfica e amigável, com compartilhamento on-line da ontologia pela *web*. Como resultados já obtidos pelo Onto4ALL pode-se destacar sua utilização em 4 universidades federais brasileiras, cursos já realizados em eventos e publicações na área. Ainda sim, o *software* precisa evoluir e ser melhor avaliado e, nesse sentido, este trabalho aplicou um formulário de avaliação de experiência do usuário com o *software* entre alunos de graduação em Ciência da Computação e pesquisadores participantes de um curso do Onto4ALL. Os resultados encontrados mostram que o *software* proposto propicia facilidades interessantes na construção de ontologias *lightweight* por usuários que têm pouca ou nenhuma experiência no tema, contribuindo assim para o aprendizado de ontologias em Ciência da Computação e áreas afins.

Palavras-chave: Editor de ontologia. Construção de ontologia. Representação do Conhecimento. Desenvolvimento *Web*. Modelagem Conceitual.

ABSTRACT

The use and applications of ontologies have grown, notably in recent years, in Computer Science and related areas, largely due to the need to semantically treat the enormous volume of data on the web, in business and academic environments. Among the many applications of ontology currently stand out: the formal representation of different domains of knowledge, integration of heterogeneous databases promoting interoperability, organization of data and business processes, availability of data in the required formats for the semantic web. Considering the ontology construction process by its developers, it is possible to notice that the creation of ontologies is a long and expensive process, which requires technical knowledge in conceptual modeling, logic and knowledge acquisition, in addition to a methodology and an editor software of ontologies. In this scenario, this work presents the ontology editor Onto4ALL, which was conceived based on the OntoForInfoScience construction methodology. The main objective is to facilitate the creation and maintenance of ontologies through a graphical and friendly interface, with online sharing of the ontology via the web. As results already obtained by Onto4ALL, it can be highlighted for its use in 4 Brazilian federal universities, courses already carried out in events and publications in the area. Still, the software needs to advance and be better evaluated and, in this sense, this work applied a user experience evaluation form with the software among undergraduate students in Computer Science and researchers participating in an Onto4ALL course. The results found show that the software developed provides interesting facilities in the construction of lightweight ontologies by users who have little or no experience in the subject, thus contributing to the learning of ontologies in Computer Science and related areas.

Keywords: Ontology Editor. Ontology building. Knowledge Representation. Web development. Conceptual Modelin.

LISTA DE FIGURAS

Figura 1.0 - Área de modelagem gráfica do Onto4ALL	35
Figura 1.1 - Menu de alterações gráficos dos objetos	37
Figura 1.2 - Chat entre colaboradores de uma ontologia	38
Figura 1.3 - Etapas da OntoForInfoScience no editor	39
Figura 1.4 - Paleta da BFO Ontology	40
Figura 1.5 - Class Expression Editor, o validador de axiomas	41
Figura 1.6 - Ontologia com problemas	42
Figura 1.7 - Gerenciador de ontologias	43
Figura 1.8 - Menu de exportação	44
Figura 1.9 - Menu de propriedades das relações com autopreenchimento	45
Figura 1.10 - Paleta de tesouros	46
Figura 1.11 - Menu de ajuda do editor	48
Figura 1.12 - Gerenciador de ontologias auxiliar	48
Figura 1.13 - Gráfico do nível de experiência com ontologia dos entrevistados	50
Figura 1.14 - Gráfico do nível de experiência com editores de ontologia dos entrevistados	50
Figura 1.15 - Gráfico da opinião dos usuários sobre modelagem visual de ontologias.....	51
Figura 1.16 - Gráfico da opinião dos usuários sobre a utilidade do console de avisos	52
Figura 1.17 - Gráfico da opinião dos usuários sobre a utilidade das etapas da metodologia OntoForInfoScience no editor	52

Figura 1.18 - Gráfico da opinião dos usuários sobre a utilidade da entrada de dados dos elementos da ontologia	53
Figura 1.19 - Indicador de conexão entre classe e relação	55

LISTA DE TABELAS

Tabela 1.0 - Tabela de metodologias para construção de ontologias	21
Tabela 1.1 - Tabela de editores de ontologia	25

LISTA DE ABREVIACÕES

API	<i>Application Programming Interface</i>
BFO	<i>Basic Formal Ontology</i>
CI	Ciência da Informação
CRUD	<i>Create, Read, Update, Delete</i>
CSS	<i>Cascading Style Sheets</i>
DCC	Departamento de Ciência da Computação
DPI	<i>dots per inch</i>
ECI	Escola de Ciência da Informação
HTML	<i>HyperText Markup Language</i>
IDE	<i>Integrated Development Environment</i>
IOF	<i>Industrial Ontologies Foundry</i>
JSON	<i>JavaScript Object Notation</i>
KOS	<i>Knowledge Organization Systems</i>
KRSS	<i>Knowledge Representation System Specification</i>
MVC	<i>Model View Controller</i>
OBO	<i>Open Biological and Biomedical Ontologies</i>
OWL	<i>Ontology Web Language</i>
PHP PHP	<i>Hypertext Preprocessor, originalmente Personal Home Page</i>
PLN	Processamento de linguagem natural
PPGCI	Programa de Pós-graduação Ciência da Informação
PPGGI	Programa de Pós-graduação Gestão da Informação
PPGGOC	Programa de Pós-Graduação em Gestão e Organização do
Conhecimento	
QoF	<i>Quality Of Life</i>
RDF	<i>Resource Description Framework</i>
RDFS	<i>Resource Description Framework Schema</i>
SI	Sistemas de Informação
SVG	<i>Scalable Vectorial Graphics</i>
TTL	<i>Turtle</i>
UFES	Universidade Federal do Espírito Santo
UFJF	Universidade Federal de Juiz de Fora

UFMG	Universidade Federal de Minas Gerais
UFO	<i>Unified Foundational Ontology</i>
UFPR	Universidade Federal do Paraná
UI	<i>User Interface</i>
UML	<i>Unified Modeling Language</i>
UNIFEI	Universidade Federal de Itajubá
UX	<i>User Experience</i>
W3C	<i>World Wide Web Consortium</i>
XMI XML	<i>Metadata Interchange</i>
XML	<i>Extensible markup language</i>

SUMÁRIO

1 INTRODUÇÃO	13
1.1 JUSTIFICATIVA	14
1.2 OBJETIVOS	16
1.3 QUESTÕES DE PESQUISA	16
2 FUNDAMENTAÇÃO TEÓRICA	18
2.1 ONTOLOGIA: CONCEITOS FUNDAMENTAIS	18
2.2 METODOLOGIAS PARA CONSTRUÇÃO DE ONTOLOGIAS	20
2.3 EDITORES DE ONTOLOGIA	24
3 METODOLOGIA DE PESQUISA	29
3.1 CLASSIFICAÇÃO DA PESQUISA	29
3.2 DESENVOLVIMENTO DO SOFTWARE	31
4 SOFTWARE ONTO4ALL	34
4.1 PRINCIPAIS FUNCIONALIDADES	34
4.2 FUNCIONALIDADES COMPLEMENTARES	47
5 RESULTADOS: AVALIAÇÃO DO EDITOR	49
5.1 FORMULÁRIO DE AVALIAÇÃO	49
5.2 DISCUSSÃO DOS RESULTADOS	54
6 PUBLICAÇÕES REALIZADAS	57
7 CONCLUSÕES E TRABALHOS FUTUROS	58
REFERÊNCIAS	60

1 INTRODUÇÃO

O termo “ontologia” é definido por diferentes áreas do conhecimento, entre elas a Filosofia e as Ciências da Informação e Computação. A parte filosófica da palavra, que, segundo o aristotelismo, é a parte da filosofia que tem por objetivo estudar as propriedades mais gerais do ser, apartada da infinidade de determinações que, ao qualificá-lo particularmente, ocultam sua natureza plena e integral (OXFORD LANGUAGES, 2022). No contexto das Ciências da Computação e da Informação, uma ontologia define um conjunto de primitivas representacionais com as quais é possível modelar um domínio de conhecimento ou discurso (GRUBER, 2008).

Como artefato de representação do conhecimento - *Knowledge Organization Systems* (KOS) - as ontologias vêm sendo utilizadas com maior frequência a partir da década de 1990, a partir dos trabalhos de Gruber (1992), Guarino (1998) e Sowa (2000), precursores do tema em Computação e CI. O aumento do uso de ontologias se deve a um conjunto de fatores, principalmente a busca pela melhoria da representação de modelos conceituais e interoperabilidade dos sistemas de informação, que a cada ano ficam mais complexos e com maior volume e processamento de dados (SMITH, WELTY, 2001).

O interesse no uso de ontologias em Ciência da informação/Computação aumentou consideravelmente e o seu uso já está difundido em diversos ramos do conhecimento da área. Vickery (1997) e Smith (2004) relatam algumas aplicações na área, entre elas:

1. Dar suporte a extração de informação (VICKERY,1997; SMITH, 2004).
2. Para a integração automática de um conjunto de vocabulários padronizados ou dicionários de dados relacionados a um domínio específico (VICKERY,1997; SMITH, 2004).
3. Alimentar um banco de dados com informações sobre categorias (conceitos) existentes no mundo/no domínio, além das propriedades referentes a esses conceitos, bem como as relações existentes entre eles (VICKERY,1997; SMITH, 2004).
4. Para a integração de bancos de dados, de *softwares* ou de modelos de negócio.

Mais recentemente, o uso de ontologias se expandiu para outras aplicações, como:

5. Classificação de documentos, que é um componente chave para várias aplicações, incluindo Sumarização de Texto, *Web Semântica*, Otimização de Mecanismos de Busca, Análise de Sentimentos entre muitos outros. A extração de palavras-chave de domínio de documentos ajuda a otimizar a tarefa de classificação de documentos envolvida na Recuperação de Informações. As técnicas de última geração existentes dependem amplamente da extração de palavras-chave com base na frequência do documento de termo (KOLLE et al., 2018)
6. Suporte a algoritmos e *frameworks* que utilizam ontologias como base de conhecimento para detecção de *fake news* - Bandara (2021) e Assis et al. (2021). Esses últimos autores, por exemplo, propuseram uma ontologia para fins de *fact-checking* denominada *ReVera Ontology* e o *framework* *ReVera*, que tem como finalidade indicar a veracidade das informações lidas pelo *framework*.
7. Publicação de dados para transparência pública como a *OntoTrans* (PEREIRA, 2020).

1.1 JUSTIFICATIVA

A construção de ontologias pode ser considerada um processo bastante complexo e oneroso, principalmente para desenvolvedores de ontologia, sejam eles estudantes, profissionais da informação, especialistas em diferentes domínios, que, muitas vezes, não estão tão familiarizados com o jargão técnico da área, além de conceitos de lógica formal e filosóficos necessários nesta atividade. Conceitos de programação orientada a objetos, tais como: classes abstratas e derivadas, instâncias, 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 que são necessários na execução as atividades relacionadas a ontologias.

Também a partir da década de 1990, surgiram editores de ontologia com diferentes características e cujo propósito geral era fornecer mais agilidade, facilidade e produtividade aos construtores de ontologia. Entre as questões principais que esses editores buscam melhorar estão:

1. Modelagem ontológica,

2. Inclusão dos elementos necessários às ontologias,
3. Verificação e validação dos erros de modelagem,
4. Formalização do conteúdo em alguma linguagem lógica. Por exemplo: OWL, RDF, RDFS, Turtle, etc.

Nos últimos anos, o número de editores cresceu consideravelmente, entre eles destacam-se: Protégé (MUSEN, 2015), Neon Toolkit (MICHAEL, WATERFELD, 2012), OLED (GUERSON et al., 2015), Top Braid Composer (TOP QUADRANT, 2019), OBOEdit (DAY RICHTER, HARRIS, HAENDEL, 2007) e o próprio Onto4ALL (MENDONÇA et al., 2020; MENDONÇA et al., 2021). O uso de editores de ontologia ainda é um desafio em Ciência da Computação e entre especialistas do domínio (que podem envolver todas áreas, como saúde, exatas e humanas). Mesmo no Protégé, editor mais bem avaliado entre seus usuários (MALIK, 2017; WARREN, 2013) e criado em 1987, seus usuários encontram dificuldades no uso de recursos que o editor oferece, conforme apontado em pesquisas sobre esse tema. Entre as dificuldades citadas em entrevistas com usuários do Protégé, Web Protége, TopBraid e SWOOP destacam-se, de acordo com Vigo (2004):

1. Necessidade de um ambiente mais colaborativo entre especialistas do domínio para modelagem da ontologia.
2. Falta de suporte para visualização de ontologias em ambientes preferidos dos usuários.
3. *Feedback* ruim na depuração de erros da ontologia.
4. Falta de suporte para pesquisa de termos em ontologias externas para reuso.
5. Interface pouco amigável na navegação pela hierarquia de classes; entre outros.

Diante do que foi apresentado, justifica-se a discussão do tema de construção de ontologias e um olhar mais aprofundado para a metodologia OntoForInfoScience e o *software* Onto4ALL, com o pretexto de avaliar se tal iniciativa fornece facilidades aos profissionais de informação e especialistas do domínio na hora de construir ontologias, além de possíveis contribuições na área como um todo. Vale ressaltar que embora o editor se oriente pela OntoForInfoScience e tenha sido construído em torno da mesma, 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 do *software* editor, servem para orientar os usuários nas etapas necessárias de construção, mesmo que segui-las não seja obrigatório. Ademais, a utilização de modelagem visual gráfica se difere dos demais editores citados, que são de base textual, o que facilita o entendimento do usuário comum ao construir suas ontologias. Já que a disposição de uma ontologia em formato de “desenho” é mais simples que especificá-la, inicialmente, em uma taxonomia e forma textual, como normalmente é encontrada nos editores de ontologia atuais.

1.2 OBJETIVOS

O **objetivo geral** é desenvolver um software editor de ontologias denominado Onto4ALL, disponível em: <https://onto4all.com/en> - que tem como diferencial em relação a outros editores de ontologia disponíveis, o uso de uma abordagem visual na modelagem de ontologias, que ocorre antes das atividades de especificação taxonômica e formal da mesma.

Como **objetivos específicos** tem-se:

1. Fornecer uma interface gráfica, compartilhável pela web, para construção de ontologias, não comum em outros editores da área.
2. Implementar um console de erros e avisos sobre o processo de construção de ontologias por seus desenvolvedores, baseado na ideia de uma IDE (*Integrated Development Environment*) para compiladores de código-fonte.
3. Comparar as funcionalidades presentes no Onto4ALL e nos demais editores de ontologia conhecidos.
4. Conduzir uma avaliação de experiência do usuário com o uso do editor em suas funcionalidades principais.

1.3 QUESTÕES DE PESQUISA

A presente pesquisa tem como objetivo responder às questões levantadas a seguir:

- QP1. O editor consegue apoiar os usuários a aprender sobre o tema e os conceitos necessários relacionados a construção de ontologias?
- QP2. A interface e a experiência de usuário fornecidas é o suficiente para os

usuários poderem usar a ferramenta adequadamente?

QP3. A modelagem gráfica de ontologias é um fator facilitador para que usuários menos experientes aprendam sobre construção de ontologias? E para usuários especialistas em ontologias?

QP4. Quais as dificuldades encontradas na modelagem gráfica de ontologias realizada como uma etapa anterior à especificação taxonômica e formal da ontologia?

2 FUNDAMENTAÇÃO TEÓRICA

2.1 ONTOLOGIA: CONCEITOS FUNDAMENTAIS

Ontologia é um tema já estudado a muito tempo pela Filosofia, a qual considera-a como “um ramo da metafísica que se preocupa com as coisas que existem” (BLACKBURN, 1996). Os filósofos, habitualmente, referem-se ao termo com “O” maiúsculo e compreendem "Ontologia" como uma descrição básica das coisas que verdadeiramente existem no mundo. Por outro lado, os cientistas da informação e da computação compreendem “ontologia” como um artefato de engenharia e um componente importante de sistemas de informação, referindo-se ao termo como “o” minúsculo, a fim de diferenciá-lo da Filosofia.

O estudo de ontologias na organização e representação do conhecimento surgiu como um ramo de pesquisa no final dos anos 80, propondo ser uma alternativa aos demais instrumentos usados para representar o conhecimento. Servindo como base de conhecimento formal e uma representação formal (em lógica) para interpretação do conteúdo por máquina, algoritmos e inferências.

Entre os componentes contidos em uma ontologia estão:

- **Classes**, que em representação de conhecimento, são uma coleção de objetos ou indivíduos (CALVANESE, 2002), que podem ser definidos por extensão (definindo seus membros) ou por intenção (especificando condições).
- **Relações (*object property*)**: são a forma que classes e indivíduos podem se relacionar uns com os outros (GOMEZ-PEREZ, 2004).
- **Indivíduos** são **instâncias** específicas de uma determinada classe. Representam a base de uma ontologia e podem ou não ser incluídos nela.
- **Axiomas**: são afirmações incluindo regras dispostas em uma formato lógico que junto corrobora com a teoria geral que a ontologia descreve em seu domínio de aplicação (GUARINO, 1998).
- **Data property**: provê uma relação que conecta classes ou instâncias desta classe ao seu valor do tipo de dados, que é uma medida ou estimativa do que é essa propriedade de dados.
- **Data type property**: São os tipos de dados das propriedades de

classes ou instâncias como, por exemplo, *strings*, números, datas, etc.

- **Annotation property:** As propriedades de anotação são usadas para colocar anotações em indivíduos, classes, propriedades e na ontologia, de maneira geral (W3C, 2022).

Outro conceito importante é a classificação dos diferentes **tipos de ontologia**. À presente pesquisa interessa dois tipos de classificação: (i) quanto ao grau de formalidade; (ii) quanto ao escopo/domínio (ontologia de alto nível ou fundamentação, ontologia de domínio, ontologia de tarefa (GUARINO, 1998).

Quanto ao **grau de formalidade** varia de linguagem natural informal a uma linguagem formal rigorosa. Usuários especialistas do domínio e a equipe de desenvolvimento da ontologia realizam esta tarefa tomando como entrada um conjunto de necessidades ontológicas para obter o propósito, escopo e nível de formalidade da ontologia, utilizando técnicas como entrevistadores físicos ou virtuais, entre eles Suárez-Figueroa (2008). Nesse âmbito, nosso editor é focado em ontologias semi-informais (*lightweight*). Em que os conceitos são conectados por associações bastante gerais do que por conexões formais estritas (DAVIES, 2010). Este fato não inviabiliza o uso do editor para ontologias de outras formalidades. Entretanto, deve-se levar em consideração que nossa ferramenta foi construída para atender, essencialmente, ontologias *lightweight*.

Quanto ao **escopo/domínio** das ontologias, elas podem ser classificadas segundo Guarino (1998) em:

- I. Ontologia de fundamentação, também conhecida como ontologia de alto nível, ontologia de nível superior, é uma ontologia que consiste em termos muito gerais (como "objeto", "propriedade", "relação") que são comuns em todos os domínios.
- II. Uma ontologia de domínio (ou ontologia específica de domínio) representa conceitos que pertencem a um domínio de conhecimento do mundo, como artes, matemática, ou política. Cada ontologia de domínio normalmente modela definições de termos específicos de domínio, especializando os termos introduzidos na ontologia de nível superior.
- III. Uma ontologia de tarefa descreve uma tarefa ou atividade genérica (como diagnóstico ou venda), especializando os termos introduzidos na ontologia de nível superior.

2.2 METODOLOGIAS PARA CONSTRUÇÃO DE ONTOLOGIAS

Historicamente, ontologias se concentram nas áreas de Modelagem Conceitual e de Representação de Conhecimento, como subáreas de Ciência da Computação. Todavia, outras áreas também se beneficiam do seu uso, entre elas, Linguística (SCHALLEY, 2019) e Ciência da Informação (ALMEIDA, 2013).

Com o aumento do uso de ontologias, surgiu a necessidade de novas metodologias para construí-las, principalmente devido a carência dos ontologistas em questões como: modelagem, lógica, conceitos técnicos da área, dentre outros (SMITH, 2004). Dentre as metodologias mais notáveis estão: Methontology (GÓMEZ-PEREZ, FERNANDEZ-LOPES, VICENTE, 1996), metodologia de Gruninger e Fox - TOVE (GRUNINGER, FOX, 1995), metodologia de Uschold e King - *Enterprise* (USCHOLD, KING, 1995), metodologia NeOn (SUÁREZ-FIGUEROA, 2008); *Up for ONtology* (UPON) (DE NICOLA, MISSIKOFF, NAVIGLI, 2009), método 101 (NOY, MCGUINESS, 2001). Em 2015, foi proposta a metodologia OntoForInfoScience (MENDONÇA, 2015), que visa uma abordagem com foco no detalhamento das tarefas e etapas do ciclo de desenvolvimento ontológico, com o objetivo de tornar mais acessível às atividades de construção de ontologias para diferentes tipos de desenvolvedores, podendo estes serem menos ou mais experientes.

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 as metodologias de processo de software da Ciência da Computação.

Trabalhos de revisão de literatura na área (JONES, BENCH-CAPON, VISSER, 1998; FERNÁNDEZ LÓPEZ, 1999; FERNÁNDEZ-LÓPEZ, CORCHO, 2004; SILVA, SOUZA, ALMEIDA, 2008; SANTOS, 2014; MENDONÇA, 2015) descrevem as metodologias para ontologias mais conhecidas e citadas, as quais estão apresentadas na tabela 1.0, incluindo seu nome, uma breve descrição e etapas previstas em cada uma delas.

Tabela 1.0 – 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 <i>ENTERPRISE</i> (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> (GOMÉZ-PÉREZ, FERNANDEZ LOPES e VICENTE,	Possibilita a construção de ontologias por reengenharia, utilizando-se do conhecimento de domínio.	<ol style="list-style-type: none"> 1. Especificação: 2. Aquisição de Conhecimento 3. Conceitualização

1996)	Domínio de aplicação: Diversos.	4. Integração 5. Implementação 6. Avaliação 7. Documentação
-------	---	--

Método 101 (NOY e GUINNESS, 2001)	Método usado no desenvolvimento de exemplos práticos utilizando o editor de ontologias Protégé. Domínio de aplicação: Diversos.	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 conhecimento consensual do mundo. Domínio de aplicação: Diversos.	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	1. Estudo de viabilidade 2. <i>Kickoff</i> 3. Refinamento 4. Avaliação 5. Aplicação e evolução

	conhecimento empresarial.	6. Geração de conhecimento 7. Captura de conhecimento 8. Recuperação e acesso
Metodologia <i>UP for ONtology</i> (UPON) (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 business).	1. <i>Workflow</i> de Requisitos 2. <i>Workflow</i> de Análise 3. <i>Workflow</i> de Desenvolvimento 4. <i>Workflow</i> de Implementação 5. <i>Workflow</i> de Teste
Metodologia <i>NeON</i> (SUÁREZ-FIGUEROA, 2010)	Metodologia para construção de redes ontológicas baseado em um desenvolvimento colaborativo Domínio de aplicação: Diversos	1. Gerenciamento de processos e atividades 2. Desenvolvimento orientado de processos e atividades 3. Suporte aos processos e atividades

Metodologia <i>MFPFO</i> (LIM, LIU e LEE, 2011)	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	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
---	--	---

		<p>ontologia de família de produtos multi facetada e anotada semanticamente;</p> <p>6. Avaliação e validação da ontologia.</p>
<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.</p> <p>Domínio de Aplicação: Diversos.</p>	<p>0. Avaliação da necessidade da ontologia</p> <p>1. Especificação</p> <p>2. Aquisição e Extração de Conhecimento</p> <p>3. Conceitualização</p> <p>4. Fundamentação Ontológica</p> <p>5. Formalização da Ontologia</p> <p>6. Avaliação</p> <p>7. Documentação</p> <p>8. Disponibilização</p>

Fonte: adaptado de Mendonça (2015).

2.3 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 Foundry (IOF)*. Mais adiante, na seção 4, foi destacado detalhadamente o nosso editor de ontologias *Onto4ALL*, citando seus recursos e funcionalidades. Neste capítulo será apresentado alguns trabalhos que discorrem sobre a utilização e avaliação de outros editores por seus usuários com base em critérios específicos. Isso dará a base para avaliar corretamente o *Onto4ALL* nas futuras seções.

Tabela 1.1: Lista de editores de ontologia

Editor	características principais	Referência
CmapTools	Editor de mapeamento conceitual. Permite representar nós e mapas conceituais em ambiente gráfico.	https://cmap.ihmc.us/cmaptools/
Graffo	Editor de código fonte aberto para a apresentação de classes, propriedades e restrições em OWL.	https://essepuntato.it/graffoo/
Menthor	Editor de código aberto, permite integração com a OntoUML, estereótipos e importação do Ent. <i>Architect</i> .	https://ontouml.org/ontouml/tooling/
NeOn Toolk	Inclui metodologia para desenvolvimento de ontologias em larga escala em ambiente distribuído.	http://neon-project.org/
OBO-Edit	Editor de código aberto, codificado em Java. Otimizado para o formato ontológico OBO.	http://oboedit.org/
OLED	Editor baseado na linguagem OntoUM fundamentada na UFO; oferece validação, simulação e detecção de padrões.	https://ontouml.org/ontouml/tooling/
Protégé	Editor de código aberto com interface para criação e edição; extensão por <i>plugins</i> mais processador de consultas.	https://protege.stanford.edu/
SWOOP	Editor de código aberto com suporte a OWL de arquitetura baseada em <i>Web</i> . Suporte a raciocínio.	http://www.mindswap.org/

yEd Graph Editor	Editor múltiplo comercial, permite trabalhar com grafos de redes semânticas e OWL.	https://www.yworks.com/products/ye
WebProtégé	Versão colaborativa na <i>Web</i> do Protégé. Suporta edições em OWL e OBO); importa e exporta formatos.	https://protege.stanford.edu/

Fonte: elaborada pelo autor.

Aspectos primordiais para softwares atualmente, como experiência do usuário (*User Experience - UX*) e interface do usuário (*User Interface - UI*) são citadas em pesquisas na área, além de avaliações de diversas outras questões pertinentes através de critérios estabelecidos. A seguir será destacado alguns dos pontos demonstrados nessas pesquisas.

De acordo com Warren (2013), uma pesquisa feita com 65 profissionais que utilizam ontologias demonstrou quais eram os editores mais utilizados. Entre as conclusões que foram geradas pela pesquisa estão:

- O Protégé foi o mais utilizado, sendo usado por 50% dos entrevistados.
- O editor comercial, TopBraid, foi utilizado por 14% dos entrevistados.
- O NeOn Toolkit (MICHAEL, WATERFELD, 2012) foi utilizado por 6% dos entrevistados.
- O CmapTools (CAN AS et al., 2004) foi utilizado por 5% dos entrevistados.
- O SWOOP (KALYANPUR et al., 2006) foi utilizado por 4% dos entrevistados.
- O OBO Edit (DAY-RICHTER, HARRIS, HAENDEL, 2007) foi utilizado por 3% dos entrevistados.

Uma pesquisa feita para avaliar características de usabilidades dos editores foi feita por Malik (2017). Foi dado a cada participante da pesquisa um conjunto de tarefas a serem realizadas em cinco editores diferentes. Após a realização das tarefas, foi pedido aos participantes que respondessem questões envolvendo a usabilidade de cada editor. Abaixo segue os resultados obtidos pela pesquisa:

- O Protégé foi o mais simples de se utilizar e exigiu o menor tempo para o cumprimento das tarefas.
- O tempo de execução das tarefas no Protégé foi cerca da metade do tempo em comparação com os editores TopBraid, NeOn e SWOOP.

- O CmapTool apresentou o pior resultado.

Outra pesquisa na área, dessa vez com cerca de 30 profissionais que utilizam ontologias, focou em selecionar qual seria o editor mais adequado para iniciantes (SIRICHAROEN, 2018). Entre os editores participantes da pesquisa estavam: Protégé, TopBraid *Composer*, NeOn Toolkit, SWOOP e OntoStudio. Dentre os resultados mais relevantes obtidos nessa pesquisa estão:

- O Protégé foi indicado como o mais adequado para iniciantes. Cerca de 70% dos profissionais o indicaram.
- Somente 10% indicaram o TopBraid.

Entre os principais fatores que elevaram o índice de indicação do Protégé foram:

1. A existência de um tutorial detalhado com uma ontologia verdadeira que mostra as diversas funcionalidades do editor através dessa ontologia.
2. A grande comunidade de usuários que o Protégé mantém, que além de utilizar o editor ainda criam plugins para a ferramenta.
3. A gratuidade do editor, não é cobrado nenhum custo para a sua utilização.
4. Ser *Open Source* (código aberto).

As principais limitações dos softwares editores de ontologia foram apontadas em (MENDONÇA et al., 2021), no qual os autores estabeleceram alguns critérios avaliativos. Nesta pesquisa foi conduzida uma avaliação individual de cada editor de ontologia explorado, tendo como base um conjunto de critérios básicos do processo de construção de ontologias. Os softwares editores analisados foram: CmapTools, Eddy for Graphol, Graffo, Hozo, ICOM, Menthor, NeOn *Toolkit*, OBO-Edit, OLED, Onto4ALL, OWLGrEd, Protégé, SWOOP, TopBraid *Composer*, yEd Graph Editor, WebProtégé, e chegou-se às seguintes conclusões:

1. **Visão gráfica da ontologia:** todos os editores analisados oferecem tal recurso.
2. **Editor disponível on-line na web:** dentre os editores analisados apenas yEd Graph Editor, WebProtégé e Onto4ALL 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 o do Onto4ALLEditor através da OntoForInfoScience. Ainda assim, o suporte fornecido por este editor precisa ser melhor vinculado e detalhando a

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 Onto4ALL 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 Onto4ALL 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 Onto4ALL 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.

3 METODOLOGIA DE PESQUISA

3.1 CLASSIFICAÇÃO DA PESQUISA

Nesta seção, é feita a classificação desta pesquisa com relação à forma de abordagem. Para isto, consultou-se as referências Lakatos e Marconi (1991), Gil (1991), Creswell (2003) sobre metodologia de pesquisa científica. Tendo como base a categorização de pesquisas científicas descrita pelos autores referenciados acima, foi possível classificar este trabalho de pesquisa como:

- **Pesquisa exploratória** (quanto aos objetivos de pesquisa), uma vez que o trabalho em questão proporciona maiores informações sobre o tema 'ontologia' no que se refere ao seu uso e aplicação no contexto da Ciência da Computação, metodologias e *softwares* para construção de ontologias.
- **Pesquisa bibliográfica, documental e com survey** (quanto aos procedimentos adotados), porque o trabalho aqui conduzido explorou referências bibliográficas na área de ontologias, bem como documentos sobre o tema em órgãos (instituições e consórcios) oficiais, tais com o W3C e o *Open Biomedical Ontologies*. Um *survey* foi realizado na parte de resultados da pesquisa para coletar as percepções dos usuários do *software* Onto4ALL quanto às suas funcionalidades e suporte às tarefas de construção de ontologias.
- **Pesquisa híbrida (quantitativa e qualitativa)** quanto à abordagem do problema, porque envolve coleta de dados da experiência do usuário com o *software* proposto com geração de dados quantitativos e interpretação qualitativa do fenômeno de construção de ontologias pelos seus desenvolvedores.

3.2 DESENVOLVIMENTO DO SOFTWARE

O desenvolvimento do editor teve início em junho de 2018, no âmbito de um projeto de iniciação científica da UFJF, sob orientação do professor Fabrício Martins Mendonça, cuja ideia era desenvolver um *software* editor de ontologias para colocar em prática a fundamentação teórica, etapas e tarefas previstas na metodologia de

construção ontologias OntoForInfoScience. Atualmente, o software tem a colaboração de 8 pessoas, sendo 5 doutores e 3 graduandos, cada um com um papel específico no desenvolvimento do software. Inicialmente, o editor era separado em dois segmentos: gerenciador de ontologias e interface gráfica de desenho. O primeiro tendo como objetivo toda a questão sistemática do editor como: autorização, autenticação, gerenciamento de informações, CRUD's (*Create, Read, Update, Delete*), etc. E o segundo, toda parte de desenho gráfico das ontologias e suas atribuições, como console de avisos, autopreenchimento de propriedades, etc. O software continua ativo, tendo atualizações gratuitas quase todos os meses e é utilizado em disciplinas de mestrado e doutorado em Universidades Federais do país, a saber: UFMG, UFJF, UFES e UFPR. Atualmente, a equipe do Onto4ALL se reúne semanalmente para discutir novas atualizações, correções de bugs e decidir os rumos que o editor deve tomar.

Quanto às tecnologias utilizadas, o Onto4ALL foi desenvolvido utilizando o framework Laravel¹, tendo como base a arquitetura *Model View Controller* (MVC) e a linguagem de programação PHP para o *back-end* e Javascript tanto para *back-end* quanto para *front-end*. Foram também utilizadas as linguagens HTML e CSS (com o framework Bootstrap²), além do pacote Laravel AdminLTE³ para o template das páginas. Optamos por utilizar o banco de dados em MySQL, junto com o phpmyadmin⁴ para administrá-lo. Em JavaScript, adotou-se uma biblioteca específica: o mxGraph⁵, modificada para permitir a criação de um componente de

¹ Laravel é um framework PHP livre e open-source criado por Taylor B. Otwell para o desenvolvimento de sistemas web que utilizam o padrão MVC (model, view, controller). Algumas características proeminentes do Laravel são sua sintaxe simples e concisa, um sistema modular com gerenciador de dependências dedicado, várias formas de acesso a banco de dados relacionais e vários utilitários indispensáveis no auxílio ao desenvolvimento e manutenção de sistemas. Disponível em: <https://laravel.com/>

² Bootstrap é um framework web com código-fonte aberto para desenvolvimento de componentes de interface e front-end para sites e aplicações web usando HTML, CSS e JavaScript, baseado em modelos de design para a tipografia, melhorando a experiência do usuário em um site amigável e responsivo. <https://getbootstrap.com/>

³ O pacote fornece um modelo de blade que você pode estender e possibilidades avançadas de configuração de menu. Além disso, inclui um conjunto de visualizações de autenticação com estilo AdminLTE. <https://github.com/jeroennoten/Laravel-AdminLTE>

⁴ phpMyAdmin é uma ferramenta de software livre escrita em PHP, destinada a administrar o MySQL pela Web. <https://www.phpmyadmin.net/>

⁵ O mxGraph é uma biblioteca de diagramação JavaScript que permite a criação rápida de aplicativos gráficos e gráficos interativos que são executados nativamente em qualquer navegador principal suportado por seu fornecedor. <https://jgraph.github.io/mxgraph/>

diagramação interativo. O controle de versão do código-fonte é feito através do Git⁶, e todo o projeto está hospedado em um repositório privado no GitHub⁷. Padrões de código limpo foram seguidos durante o desenvolvimento do projeto, além dos padrões do próprio *framework* Laravel e das linguagens PHP e Javascript.

⁶ Git é um sistema de controle de versão distribuído gratuito e de código aberto projetado para lidar com tudo, desde projetos pequenos a muito grandes com velocidade e eficiência. <https://git-scm.com/>

⁷ GitHub é uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git. <https://github.com/>

4 SOFTWARE ONTO4ALL

4.1 PRINCIPAIS FUNCIONALIDADES

Esta seção descreve o software Onto4ALL, em sua versão atual 4.2, apresentando suas características, funcionalidades e etapas para criação de ontologias no editor. Como já mencionado, a teoria que fundamenta o Onto4ALL 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, SOARES, 2017). Esse também é um dos propósitos do Onto4ALL, que pode ser usado em conjunto com tal metodologia, outra metodologia ou nenhuma, já que é um *software* independente, e pode ser utilizado desde o ramo profissional/comercial ou acadêmico.

O diferencial do Onto4ALL em relação aos demais editores avaliados nas pesquisas citadas corresponde às facilidades oferecidas no desenvolvimento por usuários sem experiência. Nesse sentido, foram incorporadas as seguintes funcionalidades principais:

1. Interface gráfica e colaborativa, disponível na web.
2. Chat em tempo real entre os colaboradores de uma ontologia.
3. Suporte metodológico para tarefas executadas no editor.
4. Auto preenchimento de algumas propriedades de classes, relações e instâncias.
5. Reuso de ontologias de alto nível, além de classes e relações individuais.
6. Validador de axiomas (*Class Expression Editor*).
7. Console de avisos e erros ontológicos exibido em tempo real na ferramenta. Indicando quais os erros o usuário está cometendo e onde localizá-los.
8. Relatórios de erros e das informações presentes nas ontologias.
9. Gerenciador de Ontologias
10. Exportação das ontologias para OWL e outros formatos.
11. Editor de tesouros.

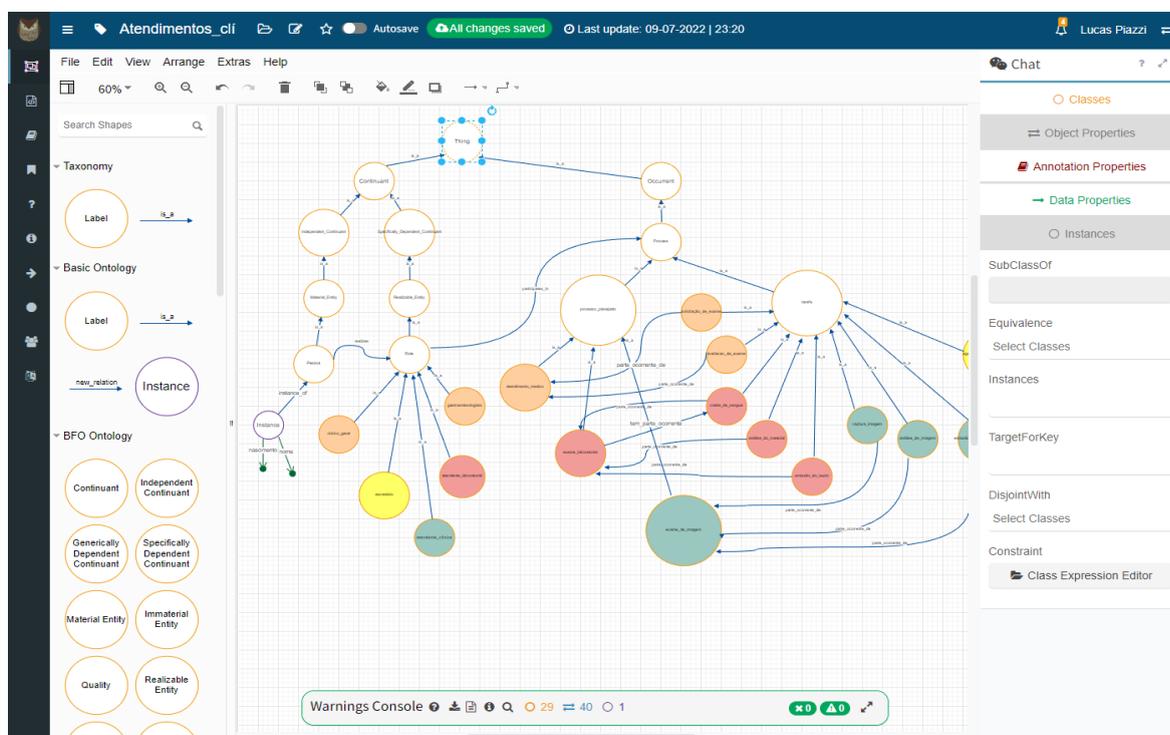
Nos parágrafos seguintes, é feita uma explicação mais detalhada de cada

uma das funcionalidades acima, com exemplos - através de figuras - destas funcionalidades no *software* editor.

A tela principal do editor é intuitiva e inclui uma **área de desenho gráfico para modelagem da ontologia no ambiente web** (vide Figura 1.0). As funcionalidades do Onto4ALL são acessadas através de um menu dinâmico, à esquerda da tela, com um ícone específico para cada uma. Também na esquerda da interface está a paleta de elementos gráficos (classes, relações e instâncias), a qual foi separada em 3 (três) grupos:

1. *Taxonomy*: elementos para construção de taxonomias – classes nomeadas por “label” e relação “is_a”;
2. *Basic Ontology*: elementos básicos para construção de ontologias – classes (círculo), instâncias (círculo menor) e relações “is_a” e “new_relation” (para relações do usuário);
3. *BFO Ontology*: constructos (elementos) para reuso de classes e relações das ontologias de alto nível *Basic Formal Ontology* (BFO) e a *Relation Ontology* (RO).

Figura 1.0: Área de modelagem gráfica do Onto4ALL.



Fonte: elaborado pelo autor.

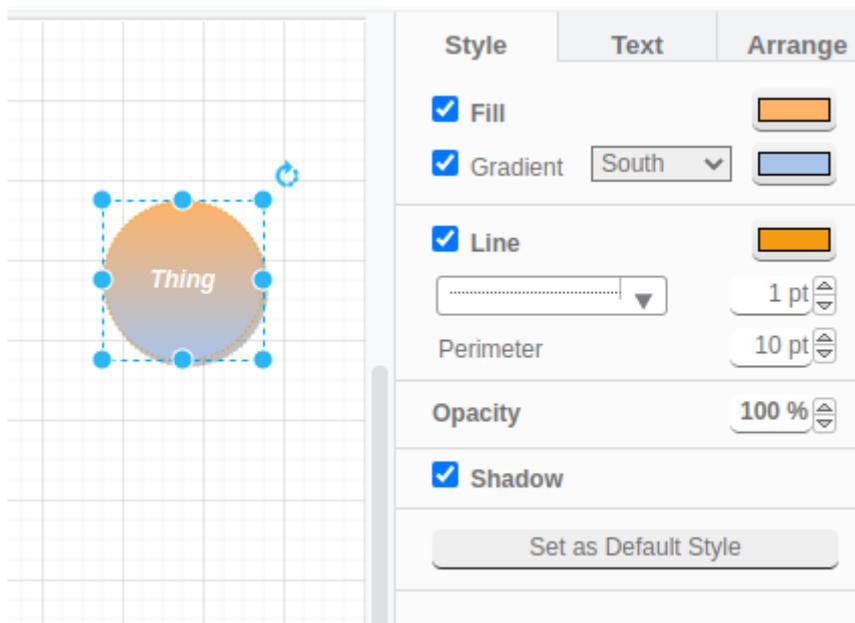
Ainda na Figura 1.0, a paleta à direita inclui os elementos textuais (propriedades) necessários para construção de ontologias separados em 5 (cinco) grupos:

1. **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);
2. **Object Properties**: representam as propriedades das relações da ontologia, tais como: `domain` e `range` (preenchidos automaticamente pelo editor), `equivalentTo`, `subpropertyOf`, `inverseOf`, `disjointWith`;
3. **Annotation Properties**: são as propriedades de anotação de uma classe ou relação da ontologia, tais como: `label`, `comments`, `isDefinedBy`, `incompatibleWith`;
4. **Data Properties**: representam as propriedades dos tipos de dados da ontologia, tais como: `value`, `domain`, `range`, `equivalentTo`, `subpropertyOf`;
5. **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 1.0 estão habilitados os grupos “Classes” (em laranja) e “Annotation Properties” (em vermelho) e “Data Properties” (em verde) porque a classe “Thing” está selecionada, e estão desabilitados os grupos “Object Properties” e “Instances” (em cinza escuro).

Todos esses objetos gráficos podem ter seu tamanho, cor, bordas, formatação, opacidade, sombra, posição, angulação e fonte alterados a qualquer momento, logo, é possível colorir sua ontologia de forma a destacar certo grupo de classes, ou então alterar a fonte de uma relação para deixá-la mais em destaque, por exemplo. Estão inclusos também funcionalidades de qualidade de vida do software (vide Figura 1.1), como a possibilidade de dar *zoom in* e *zoom out*, imprimir a ontologia atual, alterar o *grid* (linhas de fundo do desenho) do *background*, mudar a escala da página, adicionar camadas e desfazer e refazer alterações (similarmente como outras ferramentas tipo, Google Docs, Microsoft Paint, Microsoft Word).

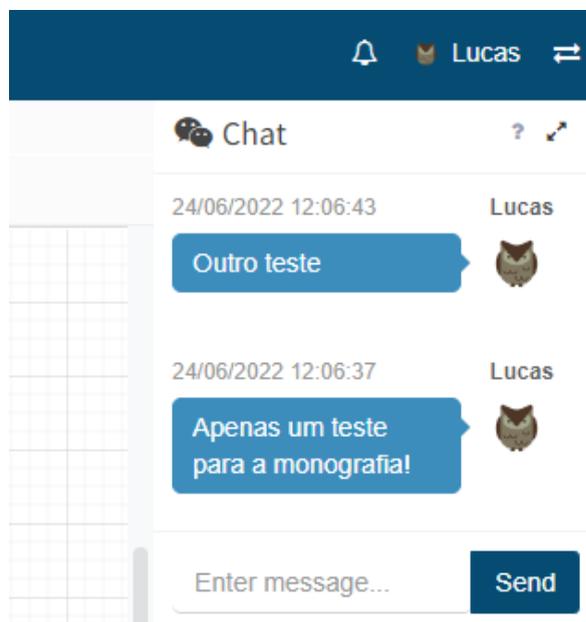
Figura 1.1: Menu de alterações gráficos dos objetos



Fonte: elaborado pelo autor.

Como parte de uma abordagem colaborativa para construção de ontologias, foi desenvolvido um *chat* de mensagens de textos entre os colaboradores de uma ontologia. Essas mensagens ficam armazenadas no banco de dados para que toda informação ali contida não se perca e possa ser acessada em qualquer momento. Apenas colaboradores que você adicionou à sua ontologia podem acessar o *chat*. Você também poderá acessar o *chat* de outras ontologias caso tenha sido adicionado como colaborador da mesma.

Figura 1.2: Chat entre colaboradores de uma ontologia.



Fonte: elaborado pelo autor.

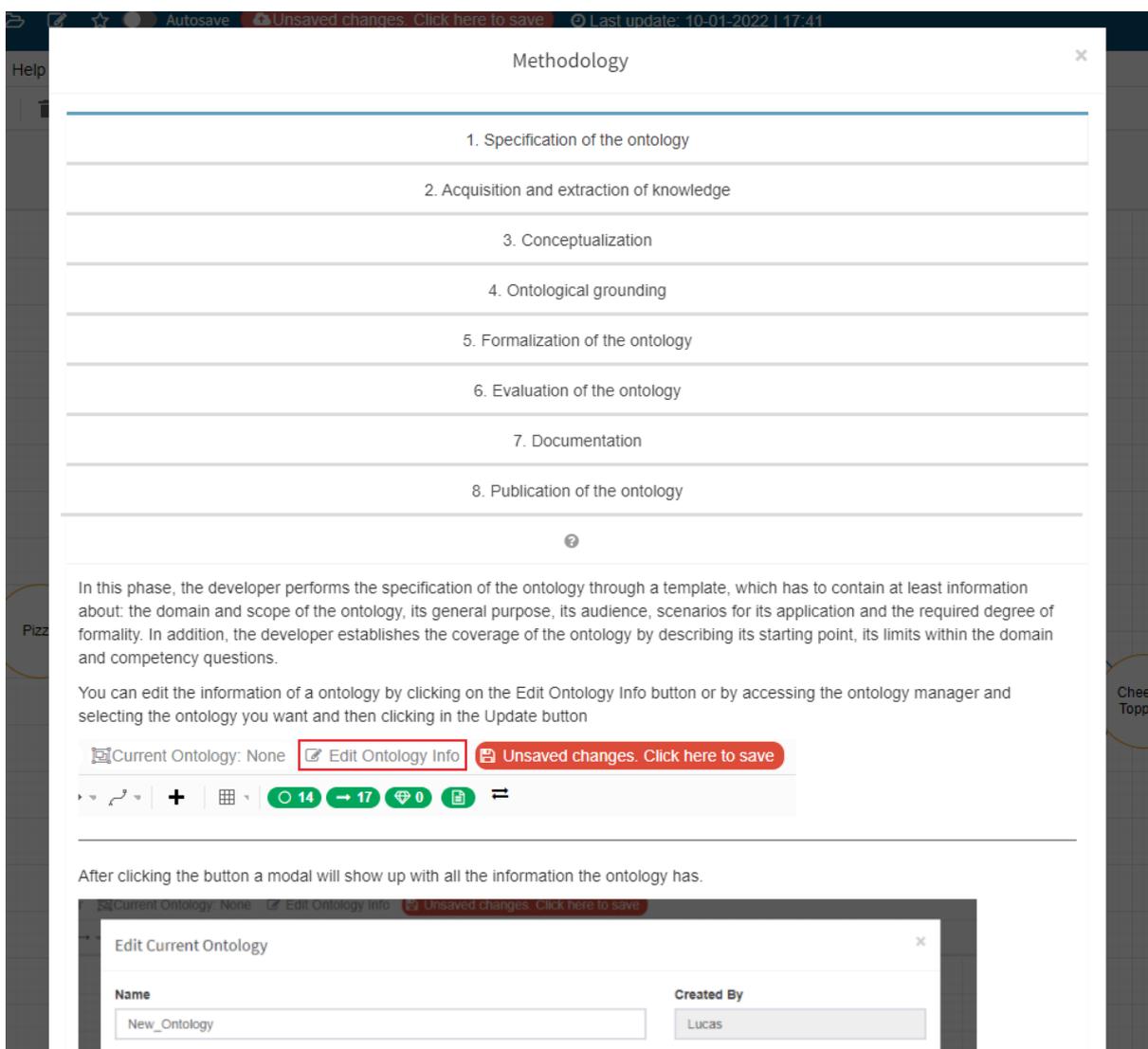
Sobre o **suporte metodológico**, o Onto4ALL é capaz de realizar sete etapas das oito previstas na OntoForInfoScience, a saber:

- 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 1.3). A Figura 1.3 ilustra a primeira etapa – especificação da ontologia – e como realizá-la a partir de um template de especificação. É possível clicar em cada uma das etapas listadas e assim ter as

informações necessárias de cada etapa e como realizá-la através do editor.

Figura 1.3: Etapas da OntoForInfoScience no editor.

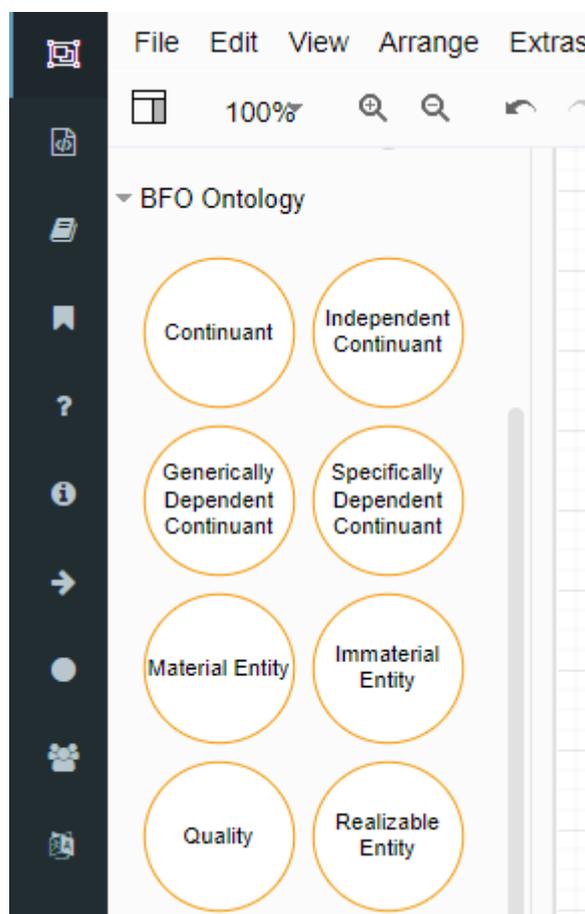


Fonte: elaborado pelo autor.

Na figura 1.4, que se localiza do lado esquerdo da página principal do editor, o usuário pode importar diretamente classes e relações da Basic Formal Ontology (BFO) apenas clicando e arrastando a classe/relação escolhida para dentro de sua ontologia. A BFO Ontology é uma **ontologia de fundamentação ou de alto nível** usada para ser a base de uma ontologia que está começando a ser construída, assim o usuário pode **reutilizar** algumas classes e relações deste tipo de ontologia, que estão previamente cadastradas no *software* editor. Atualmente, apenas o nome está embutido em cada classe/relação da paleta, mas no futuro, mais informações

(como as propriedades de cada classe/relação) já vão ser importadas juntamente com o desenho.

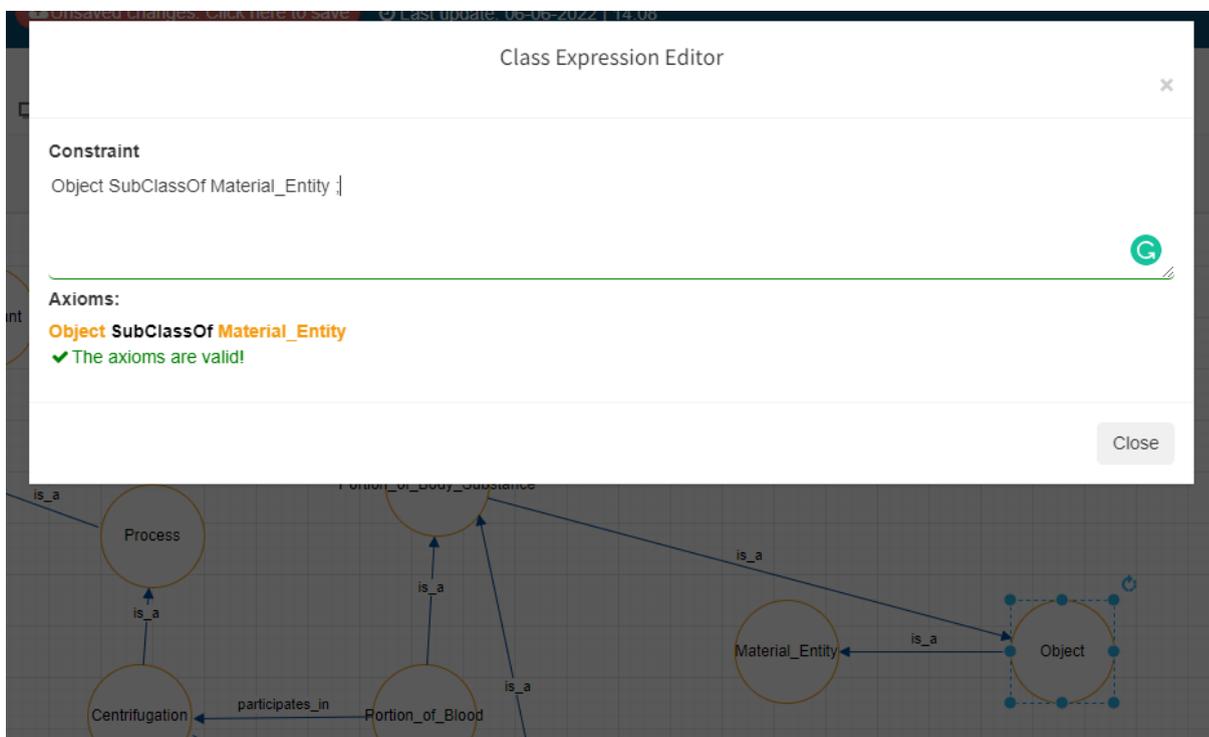
Figura 1.4: Paleta da BFO Ontology.



Fonte: elaborado pelo autor.

Na figura a seguir (1.5) podemos ver o **validador de axiomas**, acessado ao clicar no botão *Class Expression Editor* habilitado em qualquer classe da ontologia - no exemplo tem-se a classe *Object*. No exemplo com a classe *Object* o validador de axiomas faz uma requisição à OWL API que faz sua validação baseada na *OWL Manchester Syntax* e retorna uma mensagem do axioma validado - em verde para axiomas válidos e em vermelho para axiomas com erro de sintaxe. Neste exemplo, o usuário escreveu apenas um axioma simples, mas poderia escrever inúmeros que a ferramenta avalia da mesma forma.

Figura 1.5: Class Expression Editor, o validador de axiomas.



Fonte: elaborado pelo autor.

Um dos recursos mais relevantes implementados no Onto4ALL é seu **console de avisos e erros** de modelagem ontológica em tempo real. Entre os avisos e erros já disponibilizados na ferramenta estão:

1. classes duplicadas;
2. uso incorreto da relação “instance_of”;
3. herança múltipla;
4. uso indevido de relações inversas;
5. falta de anotações e metadados;
6. erros no uso do ambiente gráfico.

A Figura 1.6, 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 salvos em um arquivo-texto de erros da modelagem ontológica.

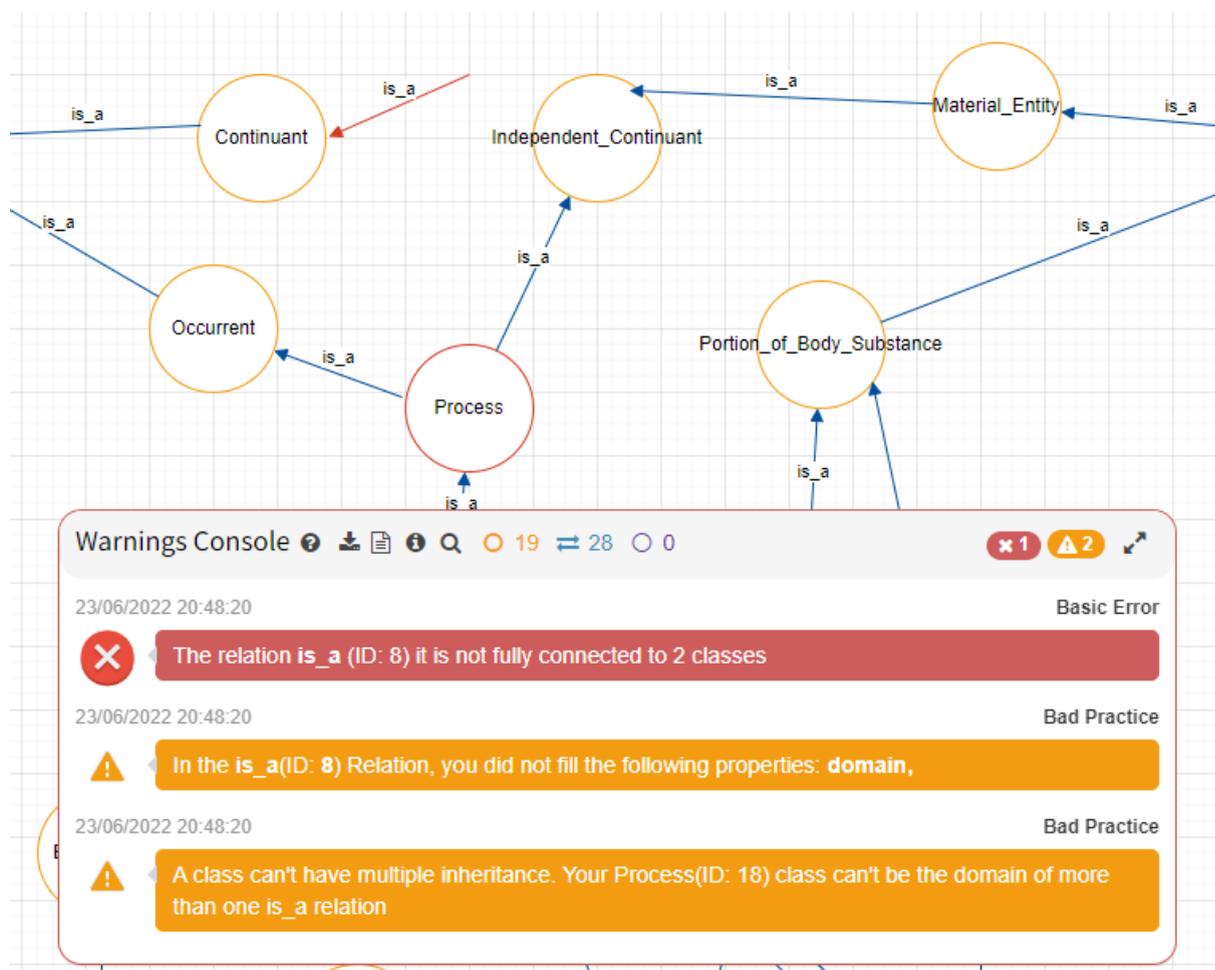
Na figura 1.6 tem-se um exemplo de ontologia com erros e avisos detectado

automaticamente pelo software, que mostra as mensagens em tempo real ao usuário. No exemplo, a relação *is_a* no topo esquerdo da figura não foi conectada corretamente a duas classes e tem-se um erro básico. Além disso, como essa mesma relação não foi conectada diretamente a seu *domain*, outro aviso é mostrado dizendo que essa propriedade precisa ser preenchida automaticamente - erro considerado má prática. Outro aviso de má prática no exemplo citado é sobre a classe *Process*, que está conectada simultaneamente pela relação *is_a* a duas superclasses (ou domínios), indicando herança múltipla, algo não recomendado ao se construir uma ontologia.

No caso dos erros mostrados na Figura 1.6, o usuário pode salvar em sua máquina um arquivo .txt do relatório de erros do console:

- *The relation is_a (ID: 8) it is not fully connected to 2 classes*
- *In the is_a(ID: 8) Relation, you did not fill the following properties: domain, A class can't have multiple inheritance.*
- *Your Process(ID: 18) class can't be the domain of more than one is_a relation.*

Figura 1.6: Ontologia com problemas.



Fonte: elaborado pelo autor.

Abaixo, na Figura 1.7, está o **gerenciador de ontologias**, onde o usuário tem uma linha do tempo de todas as ontologias que ele salvou, ou que foi compartilhada com ele. Aqui é possível ver algumas informações de cada ontologia, como: descrição, por quem foi criada, nome e quando foi feita a última edição. Também é possível abrir essa ontologia no editor clicando no botão *“Open in the editor”*.

Figura 1.7: Gerenciador de ontologias.

The screenshot displays a vertical list of five ontology update notifications in the Ontology Manager. Each notification includes an icon, the ontology name, the last update time, the creator's name, and a description. A button labeled 'Open in the editor' is provided for each entry.

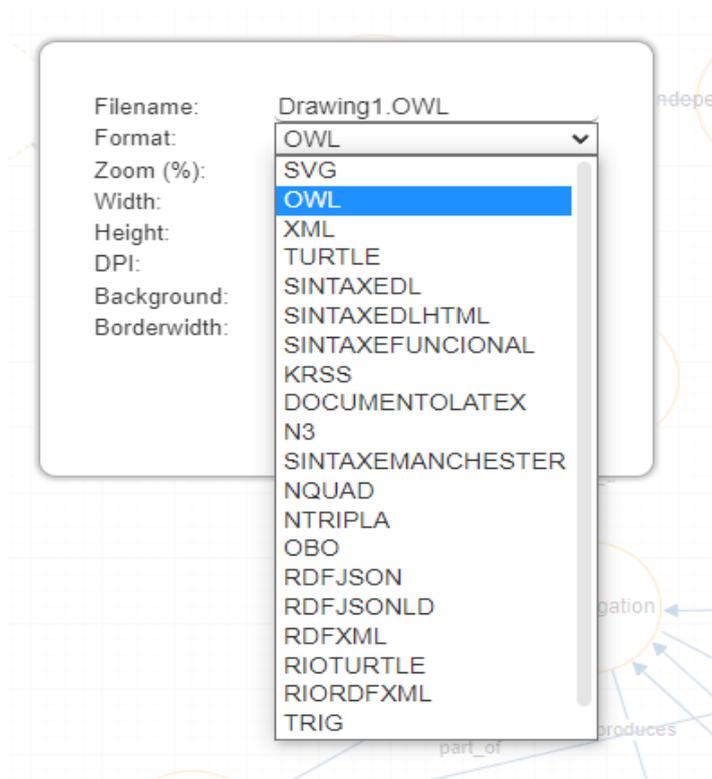
- HemoComponents was updated** (Last update: 09-07-2022 | 23:21 UTC, Created By: Fabricio Mendonca)
- Atendimentos_clínicos.xml was updated** (Last update: 09-07-2022 | 23:20 UTC, Created By: Recanto)
 - Description:** Ontologia de saúde para testes no Onto4all
- Modelo Caixa-Estoque was updated** (Last update: 09-07-2022 | 23:20 UTC, Created By: Fabricio Mendonca)
 - Description:** Esse modelo captura alguns conceitos do funcionamento de uma empresa. As caixas azuis são os estoques, guardam grandezas que são acumuladas ao longo do tempo de simulação. Eles são: estoque (quantos produtos foram produzidos e não vendidos) e caixa (quanto dinheiro a empresa acumulou até o momento). As elipses são as variáveis do modelo, podem conter valores ou expressões: produtividade; preço de mercado; custo unitário e preço. As setas azuis são os fluxos. Elas controlam quanto vai ser acrescido ou decrescido nos estoques a cada fatia de tempo da simulação.
- Mapa conceitual acervo MMissoes was updated** (Last update: 09-07-2022 | 23:20 UTC, Created By: Abeil Coelho Júnior)
- minha ontologia was updated** (Last update: 07-07-2022 | 23:34 UTC, Created By: Fabricio Mendonca)
 - Description:** Ontologia acadêmica para testes

Fonte: elaborado pelo autor.

A figura 1.8 mostra o menu de **exportação de ontologia**, em que o usuário pode exportar a ontologia modelada para um dos formatos atuais disponíveis no editor: imagem (.SVG), texto semi estruturado (.XML) e lógico (OWL/XML), TURTLE, SINTAXEDL, SINTAXEDLHTML, SINTAXEFUNCIONAL, KRSS, DOCUMENTOLATEX, N3, SINTAXEMANCHESTER, NQUAD, NTRIPLA, OBO, RDFJSON, RDFJSONLD, RIOTURTLE, RIORDFXML, TRIG, TRIX - todos esses formatos disponíveis na OWL API, que é um padrão W3C. No caso da exportação em formato SVG, é possível alterar alguns parâmetros para a exportação, entre eles: quantidade de zoom na imagem a ser exportada, altura, largura, *dots per inch* (DPI), cor do background e tamanho da borda. Para todos os formatos é possível dar nome ao arquivo a ser

salvo na máquina do usuário.

Figura 1.8: Menu de exportação.

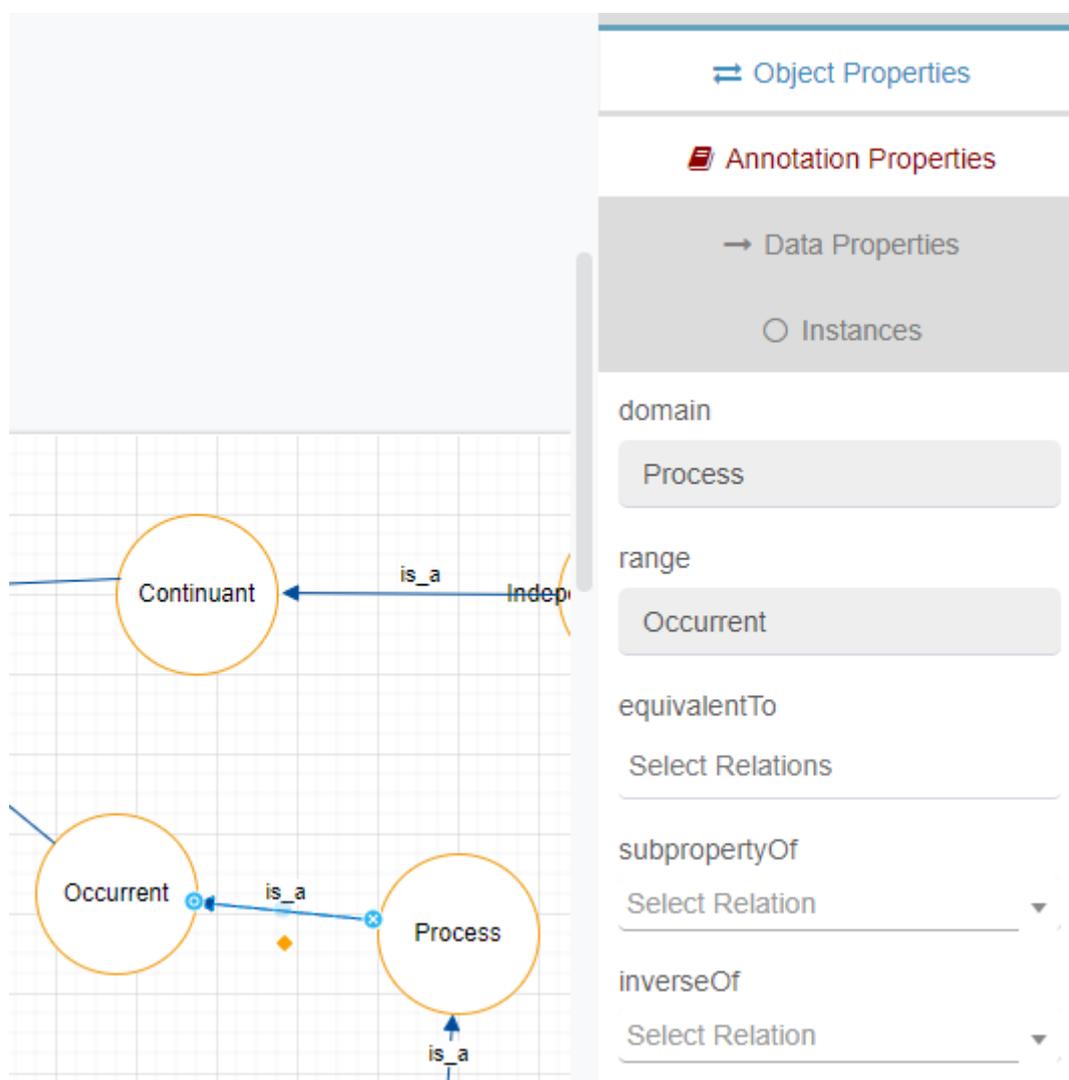


Fonte: elaborado pelo autor.

A relação *is_a* está selecionada na figura 1.9, com isso, o menu de propriedades da relação selecionada é mostrado ao lado direito. Como a relação está corretamente conectada em duas classes, as propriedades *domain* e *range* são **preenchidas automaticamente pelo editor**. Devido a essa relação, a classe *Occurrent* passa a ser uma subclasse de *Process*, logo, a propriedade *SubClassOf* de *Process* será automaticamente preenchida com o valor 'Ocurrent'. Além disso, a propriedade IRI, tanto da relação citada quanto das classes envolvidas já são automaticamente criadas pelo nosso editor, a qual é importante para identificar unicamente um elemento adicionado à ontologia. Já as propriedades *equivalentTo*, *subpropertyOf*, *inverseOf*, *disjointWith* das relações ficam com as opções de seleção travadas para apenas mostrar como valor outras relações já presentes na ontologia atual, evitando que o usuário preencha as propriedades com os valores em texto livre errados e não usando a seleção. O mesmo acontece com as propriedades *DisjointWith* e *Equivalence* das classes, e as propriedades *sameAs* e *differentAs*

das instâncias porém mostrando como opções de seleção apenas outras classes e instâncias, respectivamente, também presentes na ontologia atual.

Figura 1.9: Menu de propriedades das relações com autopreenchimento.

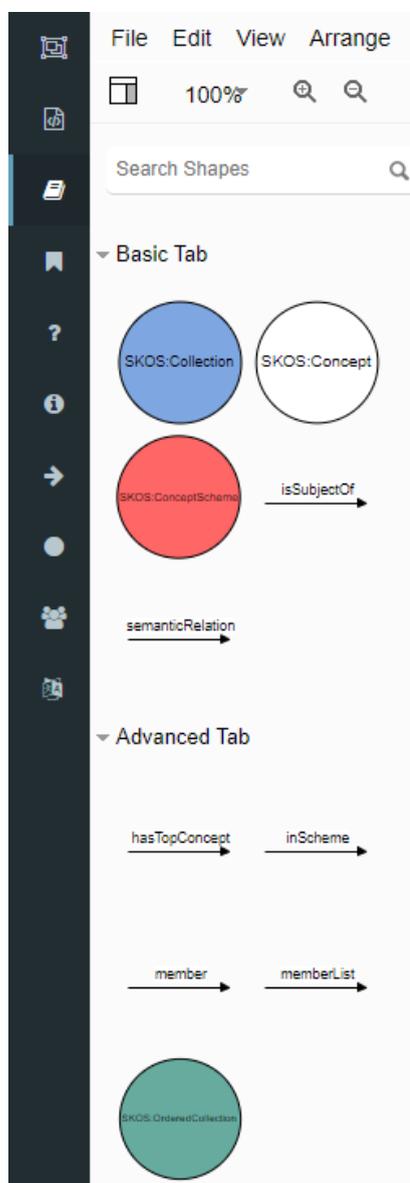


Fonte: elaborado pelo autor.

Por fim, a paleta de tesouros (Figura 1.10) funciona como um **editor de tesouros** dentro do Onto4ALL com as características específicas deste tipo de instrumento de representação da informação. Com a mesma lógica de funcionamento do editor de ontologias, o usuário clica e arrasta as figuras gráficas da paleta para dentro do seu tesouro como modelo gráfico. Diferentemente do editor de ontologias, não é possível cadastrar nenhum tipo de propriedade em cada objeto gráfico. O editor de tesouros é puramente visual, mas é possível exportar e importar tesouros do sistema através do nosso formato XML proprietário. Os tesouros criados

no editor não ficam salvos em nosso banco de dados, sendo necessário assim exportá-los em formato XML.

Figura 1.10: Paleta de tesouros.



Fonte: elaborado pelo autor.

4.2 FUNCIONALIDADES COMPLEMENTARES

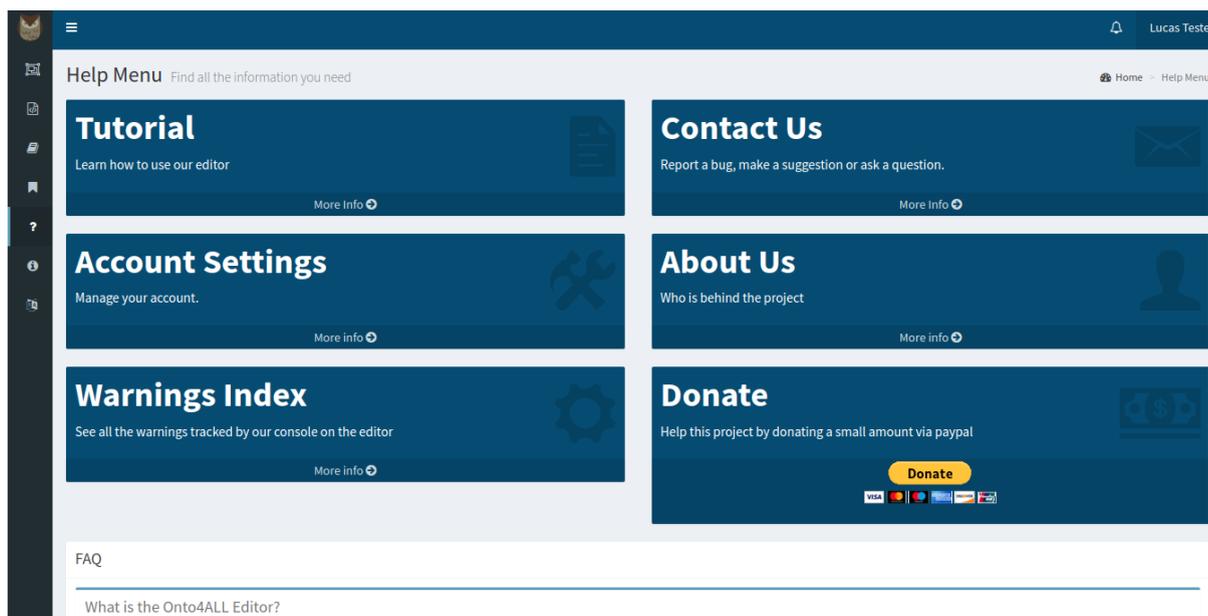
Outras funcionalidades complementares do *software* são descritas nesta subseção 4.2. O menu de ajuda (Figura 1.11) armazena diversas opções que podem ser de utilidade para o usuário, principalmente se ele estiver com dúvidas sobre o uso do editor. No menu está presente diversos recursos que quando clicados fará o redirecionamento do usuário para a área desejada, entre elas:

- Acesso ao tutorial do editor, onde está contido dicas de como utilizar o editor

de forma correta e informações adicionais sobre cada funcionalidade.

- Uma área para alterar informações do usuário, como: foto de perfil, *email* e senha.
- Uma tabela de avisos, que contém todos os avisos e erros que são rastreados pelo console de avisos.
- Um formulário de contato, no qual o usuário pode contatar diretamente a equipe do editor, para reportar um *bug*, fazer uma sugestão ou fazer uma pergunta.
- Uma área que especifica a equipe que desenvolveu e é responsável pelo projeto Onto4ALL.
- Um formulário de doação, caso algum usuário deseje doar fundos para serem investidos no editor.
- Uma aba de perguntas e dúvidas frequentes (FAQ) dos usuários, com as respostas para elas logo abaixo.

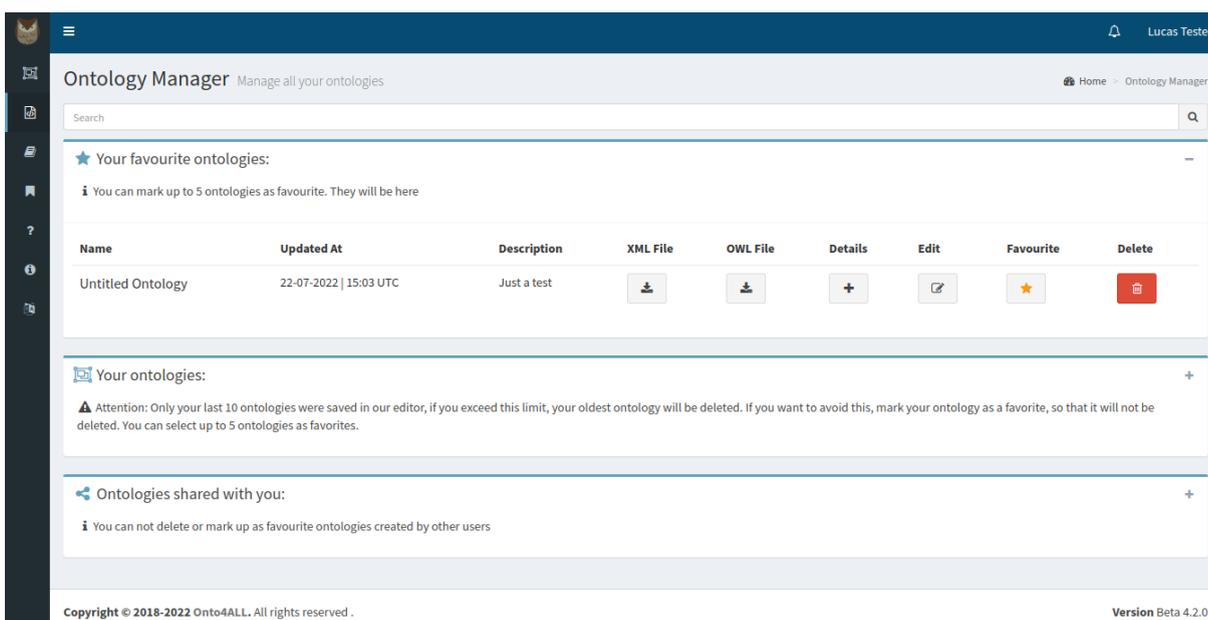
Figura 1.11: Menu de ajuda do editor



Fonte: elaborado pelo autor.

Também existe um gerenciador de ontologias auxiliar (Figura 1.12) ao que está presente na página principal do editor. Neste é possível ver em camadas as ontologias do usuário, as ontologias que ele marcou como favoritas e as ontologias que foram compartilhadas com ele. Além disso, é possível fazer uma busca por cada ontologia que ele tem acesso; acessar, editar, fazer o *download* e excluir cada uma delas.

Figura 1.12: Gerenciador de ontologias auxiliar



Fonte: elaborado pelo autor.

5 RESULTADOS: AVALIAÇÃO DO EDITOR

5.1 FORMULÁRIO DE AVALIAÇÃO

Para avaliar a experiência do usuário com o uso do software editor Onto4ALL e sua capacidade de suporte à construção de ontologias por usuários menos experientes no tema, foi conduzida uma pesquisa de avaliação (*survey*) com diversos usuários, entre eles estudantes de Ciência da Computação, pesquisadores e profissionais da informação. A maioria dos entrevistados foram estudantes do curso de Bacharelado Ciência de Computação da UFJF e participantes do minicurso 'Construindo ontologias com o editor nacional Onto4ALL' realizado na UFPR, ministrado pelo professor Fabrício Mendonça. Foi elaborado um formulário que buscou abranger os conteúdos citados nos capítulos anteriores, baseado num exercício prático (anexo 1) elaborado para o minicurso do Onto4ALL, dando um foco maior a avaliação das funcionalidades do editor e sua utilidade geral. Ao todo foram, 30 entrevistados através de um formulário no *Google Forms* disponível em: <https://forms.gle/aHXLGNzJP7KA6YKh7>. Algumas perguntas como: 'Por onde você conheceu o software' e 'Quanto você pagaria por mês para ter acesso a ferramenta?' foram omitidas neste capítulo, porque não contribuem de forma relevante para a presente pesquisa. Abaixo está o resultado da pesquisa de avaliação realizada com esses usuários.

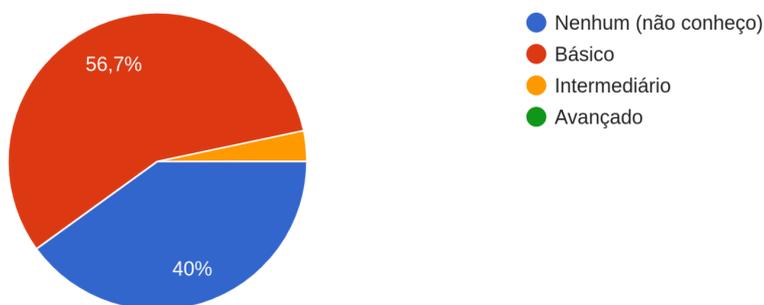
Com relação ao **perfil de usuários**, dentre os entrevistados, 13 são estudantes, 2 são bibliotecários e 1 é professor. O restante estão divididos em áreas de desenvolvimento, como analista de sistemas e desenvolvedor *web*, sendo que cerca de 46% dos entrevistados são da área de Ciência da Computação, 11% de Ciência da informação e 16% de Sistemas de Informação. Dentre todos os participantes da pesquisa, 4 possuem mestrado em diferentes áreas e 3 possuem doutorado em Gestão da Informação ou Gestão e Organização do Conhecimento.

Com relação ao **uso de ontologias**, boa parte dos entrevistados não têm nenhum conhecimento na área ou não a conhece, ou então possui um conhecimento básico, vide figura 1.13.

Figura 1.13: Gráfico do nível de experiência com ontologia dos entrevistados

1. Qual seu nível de experiência com ontologias?

30 respostas



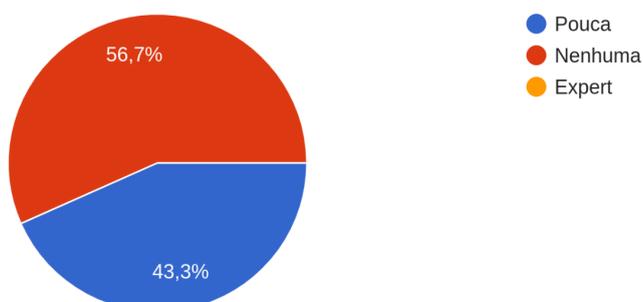
Fonte: elaborado pelo autor.

Em termos de **experiência com editores de ontologia**, 56,7% dos entrevistados disseram que não possuem nenhuma experiência com editores de ontologia e 43,3% disseram possuir pouca experiência. Já que uma boa parte dos entrevistados não tem muito conhecimento no assunto, a presente pesquisa vai avaliar, principalmente, questões de UI e UX e modelagem gráfica do editor. Não aprofundando muito em questões mais técnicas e recursos mais complexos do editor, como por exemplo o editor de axiomas.

Figura 1.14: Gráfico do nível de experiência com editores de ontologia dos entrevistados

3. Já utilizou ou tem experiência em algum editor de ontologia?

30 respostas



Fonte: elaborado pelo autor.

Respondendo às questões de pesquisa 3 e 4 - QP3 e QP4 - sobre as **funcionalidades do editor**, de acordo com a opinião dos usuários (Figura 1.15), é notório que a **modelagem gráfica de ontologias** é muito importante na hora de construir ontologias, sendo que de todos os entrevistados, cerca de 66% concluíram que esse tipo de modelagem é muito útil. E nenhum dos participantes acredita que essa abordagem não ajuda ou então é pouco útil. Corroborando com o que foi dito em capítulos anteriores.

Figura 1.15: Gráfico da opinião dos usuários sobre modelagem visual de ontologias

6. Qual sua opinião sobre a modelagem gráfica (visual) de ontologias?

30 respostas



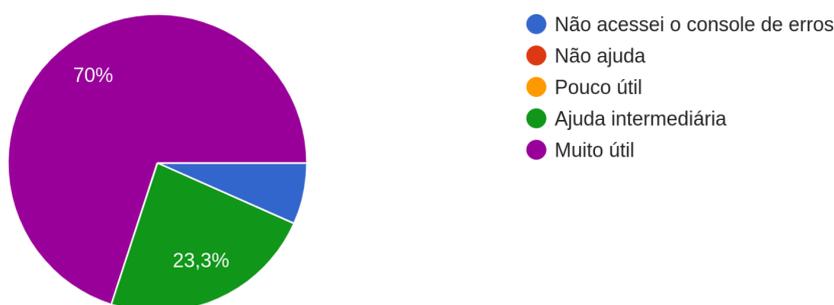
Fonte: elaborado pelo autor.

Em relação ao **console de avisos**, todos os entrevistados que o utilizaram o acharam muito útil ou de ajuda intermediária, como mostrado na figura 1.16. Como é um console que mostra os erros e más práticas cometidas pelos usuários em tempo real, isso ajuda a orientar o usuário na hora de construir sua ontologia, sendo uma funcionalidade de uso recorrente e prático, visto que, para acessar a funcionalidade, basta clicar no botão para abrir o console. Na pesquisa, apenas 2 usuários não acessaram o console.

Figura 1.16: Gráfico da opinião dos usuários sobre a utilidade do console de avisos

8. Como avalia o uso do console de erros para construção da ontologia?

30 respostas



Fonte: elaborado pelo autor.

Dentre os usuários que acessaram o menu que contém as **etapas da metodologia** OntoForInfoScience e suas descrições, todos marcaram que a presença das etapas é muito útil ou é de ajuda intermediária para construção de ontologias. Tendo em vista que boa parte dos participantes da entrevista não tem um conhecimento sólido sobre ontologias, essa funcionalidade foi essencial para o usuário saber qual próxima etapa realizar dentro do editor.

Figura 1.17: Gráfico da opinião dos usuários sobre a utilidade das etapas da metodologia OntoForInfoScience no editor

9. Qual a utilidade das etapas da metodologia do editor na construção da ontologia?

30 respostas



Fonte: elaborado pelo autor.

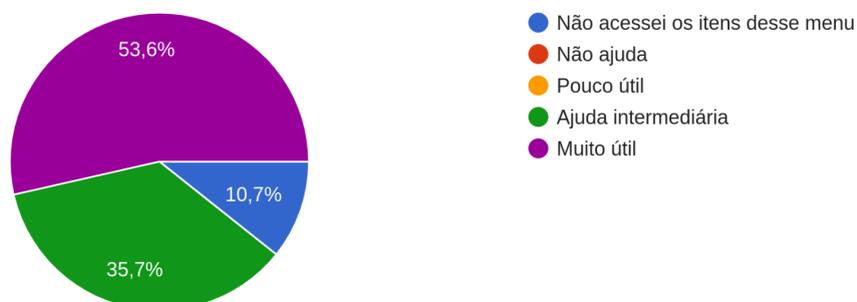
Considerando os resultados obtidos pelos gráficos das Figuras 1.16 e 1.17, é possível afirmar que tais resultados respondem à questão de pesquisa 1 (QP1), a qual se refere ao fato do software desenvolvido apoiar os usuários a aprender sobre o tema e os conceitos necessários relacionados a construção de ontologias.

Diante da **inserção de dados nos elementos da ontologia**, novamente, todos os entrevistados que acessaram essas funcionalidade, acharam ela muito útil ou de ajuda intermediária. Nota-se que as funcionalidades principais que buscam auxiliar usuários menos experientes, obtiveram opiniões muito positivas. E que cerca de 90% (Figura 1.18) dos usuários chegaram na etapa de preenchimento de propriedades das ontologias, e portanto, utilizaram o menu lateral à direita.

Figura 1.18: Gráfico da opinião dos usuários sobre a utilidade da entrada de dados dos elementos da ontologia.

10. Como você avalia a entrada de dados dos elementos da ontologia (Classes, Object Properties, Annotations, Data Properties, Instances) no menu lateral à direita?

28 respostas



Fonte: elaborado pelo autor.

Ao final do formulário, foram deixados campos para os entrevistados se expressarem mais livremente com relação às características do editor. De maneira geral, as **principais dificuldades** que teve os entrevistados com o uso do editor foram:

1. Construção e validação de axiomas.
2. Importação de ontologias.
3. Dificuldade em encaixar os objetos gráficos.

Os principais **pontos positivos** relatados foram:

1. Interface intuitiva e fácil de usar.
2. Presença do console de avisos.
3. Visual polido e bonito da interface.
4. Modelagem gráfica de ontologias.
5. Utilidade.

Os pontos relatados servem como resposta à questão de pesquisa 2 (QP2), a qual questiona se a interface e experiência de usuário fornecidas pelo Onto4ALL é o suficiente para os usuários poderem usar a ferramenta adequadamente.

Entre as principais **sugestões de melhorias** propostas pelos entrevistados estão:

1. Melhoria na importação de ontologias.
2. Acesso mais facilitado ao preenchimento de propriedades das ontologias.

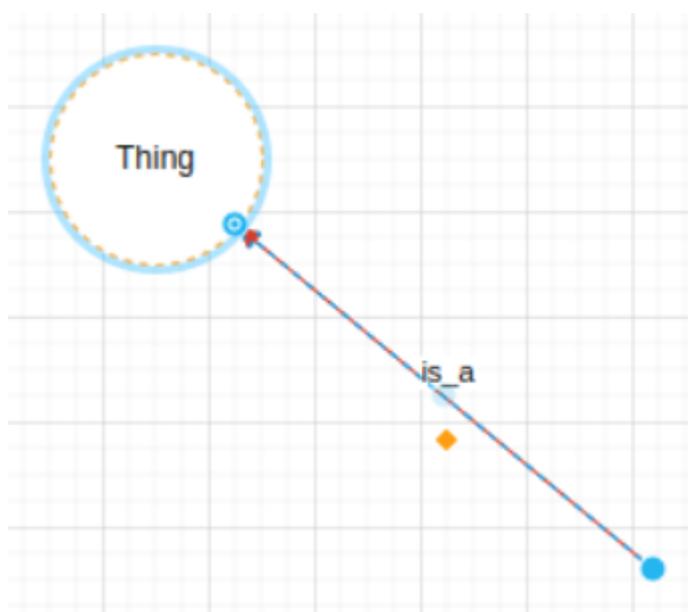
3. Aprimorar o recurso de validação dos axiomas.

Esses pontos de melhoria corroboram com a resposta da questão de pesquisa 1 (QP1), citada anteriormente.

5.2 DISCUSSÃO DOS RESULTADOS

Sobre as dificuldades citadas pelos entrevistados, isso reforça a necessidade de aprimoramento nos pontos citados, principalmente na questão de axiomas, que é uma característica bem complexa e onerosa da construção de ontologias. Logo, é vital dar mais suporte ao usuário durante o uso dessa funcionalidade. Uma possível solução seria a adição de um mini tutorial anexado ao editor de axiomas, dando exemplos e ensinando a criar axiomas usando a *OWL Manchester Syntax*. Sobre a importação de ontologias, é necessário melhorar a parte técnica e de usabilidade, visto que alguns usuários relataram problemas durante a importação de arquivos .OWL para o editor. Em relação à dificuldade em conectar os objetos gráficos, também é uma questão de usabilidade que pode ser resolvida reforçando os indicadores de conexão (que já existem) quando uma classe/instâncias é conectada em outras através de relações. Neste caso, ao arrastar uma relação até as proximidades de uma classe, sua borda ficará azulada, indicando que a classe irá se conectar com a relação em questão, vide figura 1.19.

Figura 1.19: Indicador de conexão entre classe e relação.



Fonte: elaborado pelo autor.

Com relação aos pontos positivos, reforçam-se as principais características propostas pelo *software*, tais como: usabilidade, interface gráfica ao invés de textual, dar um maior suporte e foco nos usuários menos experientes na área durante a construção de ontologias.

Pode-se, pois, perceber que as funcionalidades presentes no editor, principalmente as que envolvem o desenvolvimento de ontologias de modo gráfico é de grande ajuda para novos ontologistas. Prova disso é o crescente número de usuários se cadastrando e usando o editor nos últimos anos, principalmente na área acadêmica, na qual o editor vem sendo utilizado em aulas de mestrado e doutorado em diversas universidades federais brasileiras: a Universidade Federal de Juiz de Fora (UFJF), no Programa de Pós-Graduação em Ciência da Informação (PPGCI) da Universidade Federal do Espírito Santo (UFES), em aulas do Programa de Pós-Graduação em Gestão e Organização do Conhecimento (PPG-GOC) da Universidade Federal de Minas Gerais (UFMG) e do Programa de Pós-graduação Gestão da Informação (PPGGI) na Universidade Federal do Paraná (UFPR). O editor também já obteve interesse de empresas como Globo.com e JusBrasil. As negociações estão em andamento para possíveis parcerias. Até o momento o editor teve cerca de 100 usuários cadastrados no sistema.

É importante ressaltar que mais pesquisas de avaliação do *software* precisam ser aplicadas, principalmente em diferentes tipos de desenvolvedores de ontologias. Um cenário mais realista seria avaliar o editor com mais usuários experientes. Algo que será feito, por exemplo, no OntoBras 2022 - seminário de pesquisa em ontologia no Brasil -, onde será realizado um minicurso do editor para os participantes. A mesma ação será feita para outros cursos envolvendo o Onto4ALL no futuro.

No futuro é válido também uma nova pesquisa entre usuários ontologistas realizando tarefas em diversos editores, tais como: Protégé, OLEDB, CMAPs, e o Onto4ALL, comparando a curva de aprendizagem em cada editor e as facilidades para realização das tarefas, desde as mais simples até as mais complexas.

6 PUBLICAÇÕES REALIZADAS

Durante os quatro anos de desenvolvimento do editor publicamos diversos artigos, entre eles:

- MENDONÇA, F. M.; CASTRO, L. P. . OntoForInfoScience e Onto4ALLEditor: metodologia e editor de ontologias como facilitadores na construção de ontologias por especialistas do domínio e cientistas da informação. *Fronteiras da Representação do Conhecimento (online)*, v. 1, p. 145-173, 2021.
- MENDONÇA, F. M.; EMYGDIO, J. L. ; CASTRO, L. P. ; FELIPE, E. R. . Onto4ALLEditor: a Graphic Web Ontology Editor for Information Science. *Fronteiras da Representação do Conhecimento (online)*, v. 1, p. 70-94, 2021.
- 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 os tipos de usuários da informação. *EM QUESTÃO*, v. 27, p. 401-430, 2021.
- 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 (Onto4AllEditor: A Graphical Web Ontology Editor Oriented Different Types of Ontology Developers). 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.

7 CONCLUSÕES E TRABALHOS FUTUROS

Foi apresentada uma visão geral sobre ontologias, suas metodologias e editores, além da motivação e desenvolvimento do editor de ontologias Onto4ALL. É notório que através das características do editor apresentadas, análise de funcionalidades deste e de outros editores e dos resultados obtidos pela pesquisa, que o Onto4ALL cumpre sua função no que diz respeito a corroborar com a construção e manutenção de ontologias por novos ontologistas e ontologistas mais experientes através do uso do ambiente gráfico, interação e compartilhamento de ontologias entre os usuários pela *web*.

Através da avaliação feita do editor e dos dados aqui expostos, a pesquisa atingiu seus objetivos esperados ao responder às 4 (quatro) questões de pesquisa levantadas. O editor, de fato, consegue apoiar os usuários durante a construção de ontologias. E a modelagem gráfica de ontologias auxilia ainda mais os usuários durante o processo de construção, principalmente considerando a interface intuitiva e simples do editor. Com os dados coletados será possível focar nos pontos de melhorias citados pelos usuários, para fornecer uma experiência ainda mais positiva para os usuários.

Com a presente pesquisa muito se aprendeu com relação a ontologias e desenvolvimento *web*, o uso do framework Laravel e da biblioteca mxGraph, a qual possibilitou um ganho enorme de conhecimento nas linguagens PHP e Javascript, respectivamente. Modelagem de sistemas e engenharia de *software* foram primordiais para todo o desenvolvimento do editor, a medida que cada nova funcionalidade ia sendo implementada. Através do formulário de pesquisa aplicado, obteve-se também uma visão do que melhorar em questões de interface do sistema e experiência do usuário em geral. Táticas de código limpo, refatoração e gerenciamento de equipe também tiveram seu papel importante para o bom andamento da pesquisa.

No que se refere a trabalhos futuros da presente pesquisa estão previstas novas funcionalidades ao editor de ontologias Onto4ALL. Entre essas novas funcionalidades pode-se citar: um módulo ou *plug-in* de processamento de linguagem natural (PLN) para atender a etapa 2 da metodologia OntoForInfoScience, aquisição e extração do Conhecimento, a única etapa que ainda não é atendida pelo editor, pois envolve aquisição de conhecimento. Através de PLN visamos contemplar

pelo menos parcialmente esta etapa, de forma que o usuário possa já começar a construir sua ontologia mesmo sem uma base teórica forte. Por exemplo, através de recursos de PLN, nosso editor poderia identificar possíveis classes e relações em um texto escrito pelo usuário. E essa ontologia começaria a ser formada através do mesmo, facilitando a jornada do usuário.

Entre outras futuras funcionalidades previstas estão:

- modo noturno (escuro) do editor;
- *ontology hub* (um acervo onde os usuários poderão mostrar suas ontologias para diversas pessoas dentro e fora do editor);
- importação e exportação de ontologias em outros formatos;
- evolução de funcionalidades no editor de tesauros, muito usados no âmbito da Ciência da Informação;
- possibilidade de adição de plugins pela comunidade, assim como é feito no Protégé;
- inclusão de *reasoners* como plugins ou não para validação da ontologia além do console de erros e avisos já implementado;
- implementação de novos tipos de erros e más práticas na construção de ontologia no console do Onto4ALL.

Por fim, como conclusão final do trabalho apresentado cabe destacar o papel do Onto4ALL dentre os demais editores de ontologia disponíveis atualmente. Para cumprir seu propósito de uma abordagem colaborativa e visual para facilitar a construção de ontologias, o Onto4ALL faz uso de interface gráfica, dentre outros recursos que garantem a geração de conteúdo formal em OWL. Além da interface intuitiva, o editor destaca-se pelo console de avisos para erros ontológicos e más práticas de construção. Na comparação com outros editores, destaca-se o uso de recursos gráficos e o acesso via *web*, uma vez que a maioria dos editores com esses recursos têm alto custo de licença. Não procede comparar o Onto4ALL ao Protégé, visto que são ferramentas em estágios completamente diferentes, com abordagens distintas quanto à modelagem gráfica de ontologias, além de serem criadas em contextos diferentes e com recursos bastante díspares. Entretanto, por todos os motivos apresentados, o Onto4ALL se justifica especialmente para construção de ontologias *lightweight* por especialistas de domínio, profissionais da informação, professores e estudantes de Ciência da Computação.

REFERÊNCIAS

- 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.
- BLACKBURN, S. **The Oxford dictionary of philosophy**. Oxford; New York: Oxford University Press, 1996.
- Building Contextualized Ontology Networks. **NeOn project**. Available in: <http://www.neon-project.org>. Acesso em: 06 de Abril de 2020.
- D. Calvanese, G. De Giacomo and M. Lenzerini, "Description logics: foundations for class-based knowledge representation," **Proceedings 17th Annual IEEE Symposium on Logic in Computer Science**, 2002, pp. 359-370, doi: 10.1109/LICS.2002.1029843.
- DAVIES, John. (2010). Lightweight Ontologies. 10.1007/978-90-481-8847-5_9.
- 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.; CORCHO, O. Methodologies and methods for building ontologies. In: FERNÁNDEZ, M.; CORCHO, O. **Ontological engineering**. London: Springer, 2004. p. 107-153.
- GÓMEZ-PÉREZ, A.; FERNANDEZ, M.; VICENTE, A. J. Towards a method Gomez-Perez, Asuncion & Fernández-López, Mariano & Corcho, Oscar. (2004). *Ontological Engineering: With Examples from the Areas of Knowledge Management, E-Commerce and the Semantic Web*.
- GRUBER, T. (1992). **What is an Ontology?** Disponível em: <http://www.ksl.stanford.edu/ks-is-an-ontology.html> Acesso: 20 maio de 2020.
- GRUBER, T. (2008). "Ontology". In Liu, Ling; Özsu, M. Tamer (eds.). **Encyclopedia of Database Systems**. Springer-Verlag. ISBN 978-0-387-49616-0.
- 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, Italia, 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.

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.

Linguist Compass. 2019. 13:e12356.

M. C. Suárez-Figueroa, K. Dellschaft, E. Montiel-Ponsoda, B. Villazón-Terrazas, Z. Yufei, G. Aguado de Cea, A. García, M. Fernández-López, A. Gómez-Pérez, M. Espinoza, M. Sabou. NeOn Deliverable D5.4.1. NeOn Methodology for Building Contextualized Ontology Networks. NeOn Project. <http://www.neon-project.org>. February 2008.

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

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.

MEYLER, Charlie. The Importance Of SEO Ontology In 2021, c2021. Embryo. Disponível em: <https://www.embryodigital.co.uk/the-importance-of-seo-ontology-in-2021/>. Acesso em: 06 de jun. de 2022.

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

N. Guarino. 1998. **Formal Ontology in Information Systems**: Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy (1st. ed.). IOS Press, NLD.

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.

O. Bandara, D. Jayarathne, D. Shashinika and L. Ranathunga, "Ontology Based Fake News Detection for Sinhala Language," 2021 **6th International Conference on Information Technology Research (ICITR)**, 2021, pp. 1-6, doi: 10.1109/ICITR54349.2021.9657257.

P. Kollé, S. Bhagat, S. Zade, B. Dand and C. S. Lifna, "Ontology based Domain Dictionary," 2018 **International Conference on Smart City and Emerging Technology (ICSCET)**, 2018, pp. 1-4, doi: 10.1109/ICSCET.2018.8537346.

PEREIRA, Arthur M.; CAPPELLI, Claudia; BAIÃO, Fernanda A.; NUNES, Vanessa T.; DIARR, Bruna. **OntoTrans: An Ontology on Transparency**. In: **WORKSHOP DE COMPUTAÇÃO APLICADA EM GOVERNO ELETRÔNICO (WCGE)**, 8. , 2020, Cuiabá. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2020 . p. 84-95. ISSN 2763-8723. DOI: <https://doi.org/10.5753/wcge.2020.11260>.

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 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.

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.

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

SOERGEL, D. **Functions of a Thesaurus / Classification / Ontological Knowledge Base**. College of Library and Information Services, University of Maryland, 1997.

SOUZA, João Victor de; ASSIS, Elias Cyrino de; MENDONÇA, Fabrício Martins; SOUZA, Jairo Francisco de. **ReVera Framework: Um Framework para rastreabilidade em fact-checking automático**. In: **SIMPÓSIO BRASILEIRO DE TECNOLOGIA DA INFORMAÇÃO E DA LINGUAGEM HUMANA (STIL)**, 13. , 2021, Evento Online. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2021 .

p. 209-216. DOI: <https://doi.org/10.5753/stil.2021.17800>.

SUÁREZ-FIGUEROA, M. C. (2008). NeOn D 5.4.1. NeOn Methodology for Taniar, D., & Rahayu, J. W. (2006). Web semantics and ontology. IGI Global. <https://doi.org/10.4018/978-1-59140-905-2>

to conceptualize domain ontologies. In: European Conference on Artificial Intelligence – ECAI, 1996, Budapest, Hungary.

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

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.

VICKERY, B. C. Ontologies. Journal of Information Science, v. 23, n. 4, p. 277-286, 1997. Ontology. In: Oxford Languages. Disponível em: <https://languages.oup.com/google-dictionary-en/>. Acesso em: 06/06/2022.

W3C. OWL Web Ontology Language Semantics and Abstract Syntax , 2022. OWL Web Ontology Language Semantics and Abstract Syntax Section 2. Abstract Syntax . Disponível em: <https://www.w3.org/TR/owl-semantics/syntax.html#:~:text=Annotation%20properties%20are%20used%20to,importing%20information%20from%20other%20ontologies..> Acesso em: 09 jul. 2022.

WAND, Y.; STOREY, V. C.; WEBER, R. (1999). “**An ontological analysis of the relationship construct in conceptual modeling**”. In: *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.

ANEXO I - Exercício prático do minicurso de Onto4ALL



ONTO4ALL

CONSTRUÇÃO DE ONTOLOGIAS
Professor: Fabrício Martins Mendonça

Minicurso:

Construindo ontologias com o editor Onto4ALL

Exercício Prático

O objetivo desta atividade é exercitar na prática a construção de um trecho de uma ontologia lightweight no domínio Acadêmico, aplicando conceitos apresentados ao longo do minicurso através do editor de ontologias Onto4ALL.

1. Acesse o software através do link: <https://onto4all.com/en> e registre-se no editor de ontologias com um **login** e **senha**.
2. Escolha o **idioma** de sua preferência (inglês ou português) no menu suspenso do lado esquerdo.
3. Inicie a **modelagem** de sua ontologia na área de desenho, não esquecendo de **dar um nome** e preenchendo seu **template de especificação** (Edit ontology) no menu superior da ferramenta.
4. A aquisição de conhecimento para essa ontologia de teste no domínio acadêmico, envolve as seguintes **regras de negócio**:
 - As pessoas em uma universidade podem desempenhar os seguintes papéis: aluno, professor e funcionário.
 - Os alunos são de graduação ou de pós-graduação.
 - Os alunos estão matriculados em disciplinas.
 - Os funcionários são técnicos ou administrativos.
 - Os alunos de pós-graduação possuem um orientador, que é um professor, e participam de um grupo de pesquisa.

- Os professores estão vinculados a um grupo de pesquisa e ministram uma ou mais disciplinas.

*Obs.: para esse item 4) do Exercício, basta construir classes e relações (*object properties*) do conhecimento apresentado.

5. Represente **relações inversas** nos seguinte casos:

- Professor ministra uma ou mais Disciplinas
- Aluno de Pós Graduação possui Orientador

6. Preencha algumas **propriedades de anotação** para as classes *alunos de graduação, de pós-graduação e professor*.

7. Crie pelo menos **1 instância** para:

- Aluno de pós-graduação;
- Orientador;
- Disciplina;
- Grupo de Pesquisa.

8. Crie **2 (duas) propriedades de dados (data properties)** nas seguintes classes de sua ontologia:

- propriedade *Nome* do tipo literal na classe *Disciplina*;
- propriedade *Matrícula* do tipo integer na classe *Aluno*;

9. Elabore **axiomas** na linguagem OWL Manchester Syntax que represente as seguintes regras em linguagem natural:

- Professor ministra uma ou mais disciplinas.
- Orientador é um Professor que orienta ao menos um Aluno de Pós Graduação e participa de Grupo de Pesquisa.

10. **Compartilhe sua ontologia** desenvolvida no Onto4ALLEditor com algum colega do minicurso e peça a ele para escrever em um documento texto à parte seu **parecer de validação** para a ontologia, exercendo o papel de um especialista na validação do domínio representado.

Formulário de Avaliação do Onto4ALLEditor:

Gostaria também de solicitar que preencham o formulário de avaliação de experiência do usuário (User eXperience - UX) da atual versão do Onto4ALLEditor, disponível na tela de login ou no link abaixo:

- <https://docs.google.com/forms/d/e/1FAIpQLScoGqInGepjQ6eKO5gJ2hiTcV4hOtuEsquaGK0L74CK7jKV1g/viewform>