



**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO**  
**CENTRO TECNOLÓGICO**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

Clarete Aparecida Diniz Gomes

# **Ferramentas de Desenvolvimento de Ontologias para o método SABiOx**

Vitória, ES

2026

Clarete Aparecida Diniz Gomes

# **Ferramentas de Desenvolvimento de Ontologias para o método SABiOx**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Informática.

Universidade Federal do Espírito Santo – UFES

Centro Tecnológico

Programa de Pós-Graduação em Informática

Orientador: Prof. Dr. Vítor E. Silva Souza

Coorientador: Prof. Dra. Camila Zacché de Aguiar

Vitória, ES

2026

Ficha catalográfica disponibilizada pelo Sistema Integrado de Bibliotecas - SIBI/UFES e elaborada pelo autor

---

G633f GOMES, CLARETE APARECIDA DINIZ, 1983-  
Ferramentas de Desenvolvimento de Ontologias para o método SABiOx / CLARETE APARECIDA DINIZ GOMES. - 2025.  
94 f. : il.

Orientador: Vítor E. Silva Souza.  
Coorientadora: Camila Zacché de Aguiar.  
Dissertação (Mestrado em Informática) - Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Engenharia de Ontologias. 2. Método SABiOx. 3. Ferramentas de ontologias. 4. Integração. I. Souza, Vítor E. Silva. II. Aguiar, Camila Zacché de. III. Universidade Federal do Espírito Santo. Centro Tecnológico. IV. Título.

CDU: 004

---

Clarete Aparecida Diniz Gomes

## **Ferramentas de Desenvolvimento de Ontologias para o método SABiOx**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Informática.

Trabalho aprovado. Vitória, ES, 21 de outubro de 2025:

---

**Prof. Dr. Vítor E. Silva Souza**  
Orientador

---

**Prof. Dra. Camila Zacché de Aguiar**  
Coorientadora

---

**Monalessa Perini Barcellos**  
Membro Interno (PPGI/UFES)

---

**Fernanda Araújo Baião Amorim**  
(Membro Externo (PUC-Rio))

Vitória, ES  
2026



# **Ferramentas de Desenvolvimento de Ontologias para o método SABiOx**

**Clarete Aparecida Diniz Gomes**

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo como requisito parcial para a obtenção do grau de Mestre em Informática.

Aprovada em 21 de outubro de 2025.

Documento assinado digitalmente  
**gov.br** VITOR ESTEVAO SILVA SOUZA  
Data: 22/10/2025 07:39:39-0300  
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Vítor E. Silva Souza  
Orientador

Documento assinado digitalmente  
**gov.br** CAMILA ZACCHE DE AGUIAR  
Data: 22/10/2025 08:27:17-0300  
Verifique em <https://validar.iti.gov.br>

Prof<sup>a</sup>. Dr<sup>a</sup>. Camila Zacché de Aguiar  
Coorientadora

Documento assinado digitalmente  
**gov.br** MONALESSA PERINI BARCELLOS  
Data: 22/10/2025 11:43:27-0300  
Verifique em <https://validar.iti.gov.br>

Prof<sup>a</sup>. Dr<sup>a</sup>. Monalessa Perini Barcellos  
Membro Interno

Documento assinado digitalmente  
**gov.br** FERNANDA ARAUJO BAIÃO AMORIM  
Data: 22/10/2025 14:31:18-0300  
Verifique em <https://validar.iti.gov.br>

Prof<sup>a</sup>. Dr<sup>a</sup>. Fernanda Araújo Baião Amorim  
Membro Externo, participação remota

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO  
Vitória/ES, 21 de outubro de 2025

*Dedico este trabalho à minha filha Júlia. Meu amor, minha parceira pra tudo.*

# Agradecimentos

Agradeço, antes de tudo, a Deus e Nossa Senhora Aparecida por terem me guardado e protegido em todas as idas e vindas, por me darem coragem e persistência para continuar, por serem minha luz e guia em todos os momentos.

Agradeço meu esposo, Marcos Vinícius, pelo apoio, pela compreensão nos dias difíceis e por acreditar tanto em mim.

Agradeço à minha filha Júlia que mesmo sem entender o quão importante era pra mim esse momento, me ajudou e me deu forças para seguir.

Agradeço meus pais, Júbio (*in memoriam*) e Maria, pelos valores e ensinamentos. Tenho certeza de que meu pai estaria orgulhoso e minha mãe mesmo sem entender muito sobre o mundo acadêmico nunca deixou de me incentivar e apoiar.

Aos meus irmãos, agradeço por acreditarem em mim, sem eles não seria quem sou. Em especial, agradeço ao Claudinei que sempre insistiu no meu ingresso ao mestrado e não me deixou desistir de tentar.

Agradeço a rede de mulheres incríveis que tenho na minha vida e tornaram mais suave a minha ausência junto à minha filha. Minhas irmãs Néia e Claudete, minha sogra Rosângela e minha querida mãe Maria, vocês foram essenciais para que eu pudesse estar em outro lugar e minha filha continuar tendo um carinho e cuidado materno ainda assim.

Ao meu orientador, Vítor, agradeço pelos ensinamentos, apoio, paciência, calma, compreensão nos momentos difíceis e, acima de tudo, por sua humanidade. Que sorte a minha encontrar alguém assim para me guiar nessa etapa tão significativa na minha vida.

Agradeço à minha co-orientadora Camila por toda colaboração e ensinamentos, por confiar em mim contribuir em continuidade ao seu projeto.

Agradeço a todos os professores pelo compartilhamento de seus conhecimentos e pelas valiosas contribuições. Quero, em especial, expressar minha gratidão à professora Monalessa, que colaborou ativamente em uma etapa da minha pesquisa e acreditou em meu trabalho, obrigada por ser fonte de inspiração profissional e pessoal.

Por fim, agradeço aos colegas de disciplinas e do NEMO que, mesmo com poucos momentos de convívio, foram importantes durante essa fase.

*“Dizem que coragem não é não ter medo, mas sim fazer mesmo com medo.”*

# Resumo

O uso de ferramentas para o desenvolvimento de ontologias aumentou nos últimos anos, mas ainda apresenta desafios relacionados à seleção e integração de ferramentas que apoiem todas as fases do processo. Embora os métodos ofereçam diretrizes estruturadas para a construção de ontologias, há carência de estudos que indiquem, de forma sistemática, ferramentas adequadas para cada fase e atividade, bem como formas de integração entre elas.

Este trabalho teve como objetivo identificar, selecionar e aplicar ferramentas de apoio ao método SABiOx, mapeando sua utilização em cada etapa e avaliando sua eficácia por meio de um estudo prático. A pesquisa foi conduzida em três etapas: (i) levantamento bibliográfico sobre métodos e ferramentas de desenvolvimento de ontologias; (ii) realização de um *survey* com engenheiros de ontologia para identificar práticas e percepções quanto ao uso de ferramentas; (iii) seleção das ferramentas de acordo com cada fase do método SABiOx; e (iv) aplicação prática das ferramentas selecionadas no desenvolvimento de uma ontologia existente (OOC-O) cobrindo todas as fases do método SABiOx.

Os resultados mostraram que nenhuma ferramenta, isoladamente, é capaz de atender a todas as fases e atividades previstas no método SABiOx. Ainda assim, a utilização combinada das soluções analisadas viabilizou a execução completa do processo. O principal desafio identificado está na falta de interoperabilidade, uma vez que a integração entre ferramentas ainda não ocorre de maneira automatizada. Apesar disso, foi possível manter o fluxo de informações entre as fases, assegurando a reutilização e adaptação dos artefatos e garantindo a consistência do processo. O estudo demonstrou que a utilização do método SABiOx pode ser ampliada pela seleção criteriosa de ferramentas adequadas a cada fase, reforçando sua utilidade e valor prático. A associação do método a soluções já consolidadas e reconhecidas na área o tornou mais completo e aplicável, apresentando-se como um referencial não apenas funcional, mas também efetivamente utilizável no desenvolvimento de ontologias.

**Palavras-chaves:** Engenharia de Ontologias; Método SABiOx; Ferramentas de ontologias; Integração.

# Abstract

The use of tools for ontology development has increased in recent years, but challenges remain in selecting and integrating tools that support all stages of the process. Although methods offer structured guidelines for ontology construction, there is a lack of studies that systematically indicate appropriate tools for each stage and activity, as well as ways to integrate them.

This study aimed to identify, select, and apply tools to support the SABiOx method, mapping their use at each stage and evaluating their effectiveness through a practical study. The research was conducted in three stages: (i) a literature review on ontology development methods and tools; (ii) a survey of ontology engineers to identify practices and perceptions regarding the use of tools; (iii) selecting tools according to each phase of the SABiOx method; and (iv) practical application of the selected tools in the development of an existing ontology (OOC-O) covering all phases of the SABiOx method.

The results showed that no single tool is capable of addressing all phases and activities outlined in the SABiOx method. Nevertheless, the combined use of the solutions analyzed enabled the complete execution of the process. The main challenge identified is the lack of interoperability, since integration between tools does not yet occur automatically. Despite this, it was possible to maintain the flow of information between phases, ensuring the reuse and adaptation of artifacts and guaranteeing the consistency of the process.

The study demonstrated that the use of the SABiOx method can be expanded by carefully selecting tools that are appropriate for each phase, reinforcing its usefulness and practical value. The association of the method with already consolidated and recognized solutions in the area made it more complete and applicable, presenting itself as a reference that is not only functional but also effectively usable in the development of ontologies.

**Keywords:** Ontology Engineering; SABiOx Method; Ontology Tools; Integration.

# Lista de ilustrações

Figura 1 – Tipos de ontologias de acordo com o grau de generalidade (Guarino, 1998). . . . .	20
Figura 2 – Níveis de generalidade de ontologias (adaptado de (Falbo <i>et al.</i> , 2013)).	20
Figura 3 – Fases e atividades do método SABiOx. . . . .	25
Figura 4 – Perfil dos Participantes. . . . .	37
Figura 5 – Uso das Ferramentas em EO. . . . .	39
Figura 6 – Características das Ferramentas. . . . .	40
Figura 7 – Percepção do Uso de Ferramentas em EO. . . . .	41
Figura 8 – Relevância das Propriedades das Ferramentas em EO. . . . .	42
Figura 9 – Documento de Especificação de Requisitos da OOC-O elaborado num editor $\text{\LaTeX}$ . . . . .	52
Figura 10 – Documento de Referência da Ontologia OOC-O elaborado no editor $\text{\LaTeX}$ . . . . .	53
Figura 11 – Modularização feita no Visual Paradigm e integrada ao Documento de Referência da Ontologia no editor $\text{\LaTeX}$ . . . . .	56
Figura 12 – Diagrama OntoUML do Módulo Class da OOC-O no Visual Paradigm.	56
Figura 13 – Código no formato Turtle gerado pelo plugin OntoUML no Visual Paradigm. . . . .	57
Figura 14 – Documento da Ontologia Operacional OOC-O elaborado no editor $\text{\LaTeX}$ .	58
Figura 15 – Representação da Ontologia OOC-O importada no Protégé. . . . .	60
Figura 16 – Github com documentação e arquivos da Ontologia OOC-O . . . . .	61

# Lista de tabelas

Tabela 1 – Questões de Pesquisa do mapeamento sistemático. . . . .	26
Tabela 2 – Lista alfabética de editores de ontologia (Mendonça <i>et al.</i> , 2021). . . .	29
Tabela 3 – Ferramentas utilizadas para construção de ontologias e número de usuários.	50
Tabela 4 – Ferramentas utilizadas e suporte geral nas fases do método SABiOx . .	62
Tabela 5 – Avaliação das ferramentas em relação às propriedades. . . . .	64

# Lista de abreviaturas e siglas

EO	Engenharia de Ontologias
SABiO	<i>Systematic Approach for Building Ontologies</i>
SABiOx	<i>Extended Systematic Approach for Building Ontologies</i>
UML	<i>Unified Modelling Language</i>
UP	<i>Unified Process</i>
gUFO	<i>gentle Unified Foundational Ontology</i>

# Sumário

1	<b>INTRODUÇÃO</b>	14
1.1	Motivação	15
1.2	Objetivos da Pesquisa	16
1.3	Método de Pesquisa	17
1.4	Organização da Dissertação	18
2	<b>REVISÃO DA LITERATURA</b>	19
2.1	Ontologias	19
2.2	Métodos para o Desenvolvimento de Ontologias	21
2.3	Ferramentas para o Desenvolvimento de Ontologias	25
2.4	Trabalhos Relacionados	31
3	<b>SURVEY: USO DE FERRAMENTAS EM ENGENHARIA DE ONTOLOGIAS</b>	35
3.1	Condução do <i>Survey</i> e Síntese dos Dados	36
3.2	Discussão dos Resultados	42
3.3	Considerações finais do capítulo	47
4	<b>APLICAÇÃO DE FERRAMENTAS NO MÉTODO SABIOX</b>	49
4.1	Fase de Requisitos	51
4.2	Fase de Configuração	53
4.3	Fase de Captura	54
4.4	Fase de Design	57
4.5	Fase de Implementação	58
4.6	Avaliação	62
5	<b>CONSIDERAÇÕES FINAIS</b>	69
5.1	Contribuições	71
5.2	Trabalhos Futuros	72
	<b>REFERÊNCIAS</b>	73
	<b>APÊNDICES</b>	79
	<b>APÊNDICE A – QUESTIONÁRIO APLICADO</b>	80

# 1 Introdução

As ontologias estão presentes na Ciência da Computação em áreas como Inteligência Artificial, Engenharia de Software e Web Semântica. As ontologias são utilizadas para designar “um artefato de engenharia criado usando um vocabulário específico e usado para descrever uma porção da realidade, além de um conjunto explícito de premissas a respeito do significado pretendido com as palavras do vocabulário” (Guarino, 1998). Como qualquer artefato, ontologias têm um ciclo de vida: elas são projetadas, implementadas, avaliadas, alteradas, reutilizadas etc. (Gangemi; Presutti, 2009).

O desenvolvimento de ontologias não é uma tarefa trivial. A Engenharia de Ontologias é uma tarefa complexa e colaborativa, desafiadora tanto para engenheiros novatos quanto experientes (Reginato *et al.*, 2022; Noppens; Liebig, 2009). Para auxiliar nessas dificuldades vários métodos de Engenharia de Ontologias (EO) foram propostos para auxiliar o desenvolvimento de ontologias de alta qualidade, como por exemplo eXtreme Design (Blomqvist; Hammar; Presutti, 2016), SABiO (Falbo, 2014), SABiOx (Aguiar; Souza, 2024), SAMOD (Peroni, 2017), Neon (Suárez-Figueroa; Gómez-Pérez; Fernández-López, 2011), LOT (Poveda-Villalón *et al.*, 2022). Eles decompõem o processo em fases e recomendam atividades para cada uma. Os métodos de EO possuem diferenças entre si, mas, em geral, eles consideram que as principais atividades da EO envolvem definir os requisitos da ontologia, capturar e desenvolver a conceituação da ontologia, implementar a ontologia e avaliá-la.

Como forma de facilitar o processo de construção de ontologia, a EO também possui ferramentas para auxiliar no processo de desenvolvimento. Elas surgiram como uma forma de automação para otimizar e acelerar o processo de desenvolvimento de ontologias e são cruciais para formalizar modelos conceituais ontológicos, disseminar conhecimento, implementar ontologias legíveis por máquina e garantir a qualidade da ontologia (Rautenberg *et al.*, 2008; Tudorache, 2020).

Nesse contexto, atualmente existe uma ampla gama de ferramentas para apoiar a EO. Algumas são ferramentas criadas para atividades de EO (por exemplo, Protégé<sup>1</sup>), e outras que existem para outros propósitos, mas que atendem e são utilizadas para atividades de EO (por exemplo, Visual Paradigm<sup>2</sup>).

Durante o processo de desenvolvimento de uma ontologia, selecionar as ferramentas mais adequadas e que forneçam os recursos necessários é uma decisão importante e pode não ser uma tarefa fácil (Abburu; Babu, 2013). Na escolha de ferramentas vários fatores

---

<sup>1</sup> <<https://protege.stanford.edu/>>

<sup>2</sup> <<https://www.visual-paradigm.com/>>

devem ser levados em consideração, como escopo da ontologia, características da equipe (por exemplo, tamanho, experiência), contexto da organização (por exemplo, políticas, preferências) e uso pretendido da ontologia. Considerar as características da ferramenta também é relevante (por exemplo, disponibilidade, recursos).

Alguns métodos de EO auxiliam na seleção de ferramentas, sugerindo algumas para apoiar suas atividades (Iqbal *et al.*, 2013; Poveda-Villalón *et al.*, 2022). As ferramentas de apoio ao desenvolvimento de ontologias vem evoluindo ao longo dos anos e algumas ferramentas se tornaram muito populares (por exemplo, Protégé), porém ainda há necessidade de entender quais ferramentas oferecem o melhor suporte para todo o processo de EO (Cardoso, 2007; Vigo *et al.*, 2014). No geral essas ferramentas não estão totalmente integradas, atendem apenas algumas atividades e não a todo processo de desenvolvimento da ontologia (Corcho; Fernández-López; Gómez-Pérez, 2003). Para isso, é necessário investigar as ferramentas existentes, suas características e funcionalidades, quais ferramentas são utilizadas por engenheiros de ontologia e o quão úteis são consideradas por eles.

## 1.1 Motivação

Dada a diversidade de métodos, ferramentas e linguagens para o desenvolvimento de ontologias, a decisão sobre quais utilizar pode se tornar difícil. Neste contexto, é importante compreender as etapas envolvidas, o suporte à construção colaborativa e o ciclo de vida da ontologia de acordo com o método escolhido. Construir uma ontologia pode se tornar uma tarefa difícil e demorada quando se tem que selecionar as ferramentas adequadas com os recursos necessários (Abburu; Babu, 2013).

Corcho, Fernández-López & Gómez-Pérez (2003) afirmam, em um estudo sobre diversas abordagens para desenvolvimento de ontologias, que nenhuma das apresentadas está totalmente madura se as compararmos com métodos de Engenharia de Software e Engenharia do Conhecimento. As propostas não são unificadas e é necessário um grande esforço para criar um método consensual para construção de ontologias.

Alguns métodos indicam ferramentas para utilização em suas atividades como, por exemplo, WebODE, indicada por METHONTOLOGY (Arpírez *et al.*, 2001); e OntoEdit, indicada por On-To-Knowledge (Sure; Angele; Staab, 2003; Staab *et al.*, 2001). No entanto, a maioria dos métodos ainda não oferece uma indicação precisa de quais ferramentas utilizar em cada atividade. Na maioria dos casos as ferramentas são sugeridas para somente algumas fases ou atividades, existem muitas ferramentas semelhantes e uma incapacidade de interoperabilidade entre as mesmas (Corcho; Fernández-López; Gómez-Pérez, 2003).

Embora as ferramentas de criação de ontologias tenham visto muitas melhorias nos últimos anos, mais notavelmente demonstradas pela popularidade do Protégé como uma

ferramenta de desenvolvimento de ontologia integrada (Cardoso, 2007), ainda não temos uma boa compreensão da eficácia das ferramentas existentes ou de que tipo de suporte de ferramenta os autores de ontologias precisam para criar, explorar, reutilizar, refatorar e depurar ontologias com sucesso. Uma das grandes dificuldades na construção de ontologias é a disponibilidade de suporte de ferramenta adequado que auxilie usuários novatos e experientes durante todo o processo de desenvolvimento. Geralmente as ferramentas atendem apenas algumas atividades do processo e não estão totalmente integradas, dessa forma não atendem a todo o ciclo de vida da ontologia (Corcho; Fernández-López; Gómez-Pérez, 2003).

Diante de tais apontamentos, pode-se perceber que a criação de ontologias não é tão simples e que ferramentas adequadas para atividades do ciclo de vida de uma ontologia podem tornar o processo de criação mais fácil e acessível. Com um método sugerindo as ferramentas adequadas para suas atividades e sendo essas integradas entre si, seria uma facilidade tanto para ontologistas experientes e principalmente para ontologistas iniciantes, servindo como ponto de partida para uso de ferramentas e integração das mesmas, favorecendo a construção de ontologias de maior qualidade. Neste contexto, essa pesquisa concentrou-se na identificação e na integração de ferramentas que atendam as atividades propostas por um método de EO, como forma de avanço num processo de auxílio na construção de ontologias. As propostas foram feitas no âmbito do método SABiOx (Aguiar; Souza, 2024), em desenvolvimento pelo grupo no qual esta pesquisa foi conduzida.

## 1.2 Objetivos da Pesquisa

Este trabalho tem como **objetivo geral** instrumentar as fases do método SABiOx utilizando ferramentas de apoio para a construção de ontologias. Esse objetivo geral pode ser detalhado nos seguintes **objetivos específicos**:

1. Identificar e selecionar ferramentas existentes que possam ser utilizadas de forma eficaz em diferentes fases do método SABiOx para a construção de ontologias;
2. Analisar ferramentas indicadas e apresentar formas de integrá-las ao método SABiOx nas diferentes fases propostas;
3. Avaliar a utilização das ferramentas integradas ao método SABiOx por meio de aplicação prática.

## 1.3 Método de Pesquisa

Para alcançar os objetivos da pesquisa, foram realizadas as seguintes etapas. Na primeira etapa, foi realizado um levantamento bibliográfico com base no material indicado pelos orientadores sobre ontologias, incluindo artigos, trechos de livros, teses e dissertações publicadas relevantes e conceituados na área, proporcionando um embasamento teórico sólido sobre o tema. Além disso, foi conduzida uma busca bibliográfica *ad hoc* para investigar diferentes ferramentas e metodologias de desenvolvimento de ontologias, permitindo a identificação de trabalhos já desenvolvidos com objetivos semelhantes aos da pesquisa.

Em seguida, iniciou-se um mapeamento sistemático para obter resultados mais concretos em relação às ferramentas utilizadas na construção de ontologias em diferentes etapas do processo. Esse estudo contou com a colaboração de uma aluna de mestrado que pesquisa sobre o mesmo tema e duas alunas de doutorado com pesquisas relacionadas, sob a supervisão dos orientadores e da coorientadora. O mapeamento, ainda em andamento, utilizou uma *string* de busca que resultou em quase 2.000 artigos. Após a aplicação do primeiro filtro, foram identificadas diversas ferramentas empregadas no processo de construção de ontologias. Esse levantamento revelou que a pesquisa possui grande relevância, pois muitos trabalhos indicam a ausência de ferramentas que abranjam todo o processo e destacam a dificuldade enfrentada por iniciantes na área. Por meio desse estudo obtivemos novos *insights* de pesquisa na mesma linha desse trabalho.

A partir das leituras do mapeamento, constatou-se que um dos principais problemas apontados na literatura é a falta de ferramentas que cubram todas as etapas do desenvolvimento de ontologias e a ausência de integração entre elas, dificultando significativamente o processo. Motivada por essa questão, surgiu a proposta de aplicação de um survey para ontologistas, com o objetivo de coletar dados que confirmassem as problemáticas levantadas na literatura. O *survey* foi realizado entre o final de 2023 e o início de 2024, trazendo resultados valiosos, mostrou que apontamentos da literatura corroboram com a realidade de engenheiros de ontologia em seu cotidiano. Este estudo resultou na publicação do artigo *Investigating Tool Usage in Ontology Engineering: a Survey* (Gomes *et al.*, 2024a).

Após aplicação do *survey* e com a identificação das ferramentas mais utilizadas e as características consideradas relevantes por ontologistas, foi realizada a etapa prática do estudo. Foi selecionada a ontologia *Object-Oriented Code Ontology* (OOC-O) (Aguilar; Falbo; Souza, 2019) para exemplificação, por ser adequada para aplicação ao método. Com o apoio das ferramentas selecionadas, a ontologia foi reconstruída desde a primeira até a última fase do método SABiOx. Em cada fase, foram levantados os resultados esperados definidos pelo método e, em seguida, apresentou-se a ferramenta que se adequava àquela fase e como ela atendia ao que era proposto e os resultados que ela produzia. Dessa forma, foram apresentadas as ferramentas que contribuem para todas as etapas do método SABiOx.

Terminada a exemplificação prática das ferramentas foi realizada a etapa de avaliação das mesmas. Primeiramente foi realizada uma avaliação num âmbito mais geral onde analisaram-se as fases que cada ferramenta atendia e qual o nível de atendimento, se total, parcial ou inexistente. Em seguida utilizou as propriedades da ISO 25010 (ISO, 2011) já levantadas no *survey*, para uma avaliação um pouco mais criteriosa e específica. A partir das propriedades foram elaborados critérios de atendimento da ferramenta para cada propriedade e cada ferramenta foi avaliada, indicando se atendia a cada critério de forma total, parcial, não atende ou não se aplica, conseguindo assim uma avaliação mais rigorosa e completa.

## 1.4 Organização da Dissertação

Neste capítulo inicial foram apresentadas as principais ideias desta dissertação, descrevendo o contexto de aplicação, motivações, objetivos e método de pesquisa. O restante desta dissertação está organizado da seguinte forma:

- **Capítulo 2 - Revisão da literatura:** apresenta o embasamento teórico relacionado a ontologias, ferramentas e métodos de desenvolvimento de ontologias, relevantes para este trabalho;
- **Capítulo 3 - Uso de ferramentas em Engenharia de Ontologias:** apresenta os resultados de aplicação de um *survey* onde foi investigado o uso de ferramentas em Engenharia de Ontologias;
- **Capítulo 4 - Aplicação de ferramentas no método SABiOx:** apresenta as ferramentas que podem ser utilizadas nas diferentes fases e atividades do método SABiOx, indicando o que é realizado em cada etapa e quais demandas específicas cada ferramenta deve atender. Demonstra o uso na prática e apresenta a avaliação realizada nas ferramentas aplicadas, apresentando os critérios utilizadas para avaliação e como se procedeu a mesma;
- **Capítulo 5 - Considerações Finais:** apresenta as considerações finais do trabalho, as contribuições e perspectivas de trabalhos futuros para continuidade e aprimoramento do trabalho.

## 2 Revisão da Literatura

Este capítulo resume assuntos que servem de fundamentação teórica para o trabalho apresentado nesta dissertação: ontologias (Seção 2.1), métodos e ferramentas para o desenvolvimento de ontologias (respectivamente, seções 2.2 e 2.3). Por fim, apresentam-se também os trabalhos relacionados (Seção 2.4).

### 2.1 Ontologias

Uma ontologia é uma especificação formal e explícita de uma conceituação compartilhada (Gruber, 1993). De acordo com a comunidade da Web Semântica, “conceituação” trata-se de um modelo abstrato que representa uma realidade, identificando os conceitos mais relevantes para sua descrição; “explícita” indica que os conceitos e as restrições associadas ao seu uso são definidos explicitamente; “formal” quer dizer que a ontologia é especificada de forma rigorosa, permitindo que seja compreendida e processada por máquinas; e “compartilhada” refere-se ao fato que o conteúdo reflete um conhecimento consensual dentro de uma determinada comunidade, favorecendo a padronização e a reutilização da informação (Ding, 2001).

As ontologias têm sido utilizadas na Ciência da Computação desde meados da década de 1960, especialmente nos campos da Inteligência Artificial e da Representação do Conhecimento. Elas são definidas como um artefato de engenharia criado usando um vocabulário específico e usado para descrever uma porção da realidade, além de um conjunto explícito de premissas a respeito do significado pretendido com as palavras do vocabulário (Guarino, 1998). Reconhecidas como ferramentas conceituais essenciais, as ontologias vêm ganhando cada vez mais espaço em pesquisas na Ciência da Computação. Nos últimos quinze anos, houve um crescimento nos estudos sobre o tema em diferentes áreas da computação (Reginato *et al.*, 2022), especialmente modelagem conceitual (Oberle, ; Mendonça *et al.*, 2021) e Inteligência Artificial (Arrieta *et al.*, 2020; Confalonieri; Guizzardi, 2023). Além disso, consolidaram-se como a principal abordagem para lidar com aspectos semânticos em iniciativas de integração semântica (Júnior *et al.*, 2021).

Ontologias podem ser classificadas de diversas formas de acordo com a literatura. Uma das formas mais conhecidas é baseada no grau de generalidade e foi proposta por Guarino (1998). Essa classificação define quatro tipos de ontologias: Ontologias de Fundamentação, Ontologias de Domínio, Ontologias de Tarefa e Ontologias de Aplicação. Ontologias de Fundamentação abordam conceitos muito gerais, como como espaço, tempo, matéria, objeto, evento, ação, etc. Servem de base para ontologias de domínio e de tarefa. Ontologias de Domínio descrevem o vocabulário relacionado a um domínio genérico (como

medicina ou automóveis) e Ontologias de Tarefa descrevem, respectivamente, ou uma tarefa ou atividade genérica (como vendas ou diagnóstico) por especialização dos termos introduzidos na ontologia de fundamentação. Ontologias de Aplicação descrevem conceitos dependentes ao mesmo tempo de um domínio e uma tarefa particulares, as quais são frequentemente especializações de ambas as ontologias relacionadas. Essa classificação é representada na Figura 1.

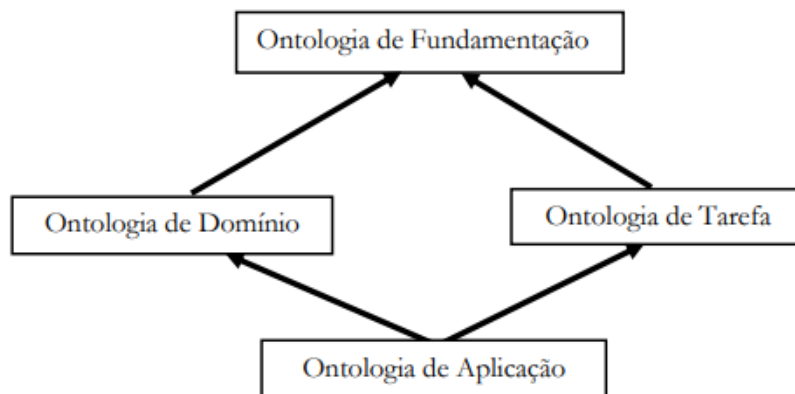


Figura 1 – Tipos de ontologias de acordo com o grau de generalidade (Guarino, 1998).

d'Aquin *et al.* (2011) classificam as ontologias em ontologias de fundamentação e de domínio de forma semelhante a Guarino (1998), porém adicionando um nível entre elas, chamado de Ontologias de Núcleo (*Core Ontologies*). Uma ontologia de núcleo fornece uma definição precisa do conhecimento estrutural em uma área específica que cobre diferentes domínios de aplicação. São construídas baseadas em ontologias de fundamentação e representam um refinamento dessas, adicionando conceitos e relações específicos da área considerada (d'Aquin *et al.*, 2011).

Falbo (2014) apresentam a variação de generalidade entre os tipos de ontologias como uma linha contínua, variando de ontologias de fundamentação para ontologias de domínio, conforme mostrado na Figura 2. Os autores afirmam ainda que pode haver diferentes níveis de generalidade nas ontologias dentro dos modelos que estão classificadas.

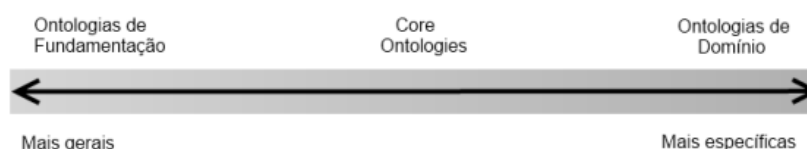


Figura 2 – Níveis de generalidade de ontologias (adaptado de (Falbo *et al.*, 2013)).

Guizzardi *et al.* (2007) classificam as ontologias em ontologias de referência e ontologias operacionais. As ontologias de referência são tipos especiais de modelos conceituais e são construídas com o objetivo de fazer a melhor descrição possível da realidade. É um tipo especial de modelo conceitual, um artefato de engenharia com o requisito adicional de

representar um modelo de consenso dentro de uma comunidade. A partir de uma ontologia de referência com consenso dos usuários, pode-se criar a versão operacional da mesma. As ontologias operacionais, por sua vez, quando o objetivo principal é garantir propriedades computacionais desejáveis. Então, são implementações das ontologias e podem ser lidas por uma máquina (Falbo, 2014).

## 2.2 Métodos para o Desenvolvimento de Ontologias

Engenharia de Ontologias é o estudo das atividades que dizem respeito ao processo de desenvolvimento de ontologias, ao ciclo de vida de ontologias e aos métodos, ferramentas e linguagens para construção de ontologias (Gómez-Pérez; Fernández-López; Corcho, 2004b).

A partir dos anos 1990, verificou-se um grande número de iniciativas para o uso de ontologias em sistemas de informação. Foram desenvolvidas a partir de então uma série de métodos e propostas de como conduzir a construção de uma ontologia. Ao longo de todos esses anos, a criação de modelos conceituais foi motivada pela busca de melhorias na representação em sistemas de informação (Mendonça; Castro, 2021). Esta seção apresenta alguns dos principais métodos de construção de ontologias encontrados na literatura.

O método Methontology (Fernández-López; Gómez-Pérez; Juzgado, 1997) é utilizado na Engenharia de Ontologias com base no processo de desenvolvimento de software e segue um processo rigoroso baseado em três categorias: Gerenciamento, Desenvolvimento e Suporte. O ciclo de vida da ontologia é baseado em protótipos e o método incentiva o reúso de ontologias existentes e ciclos iterativos de desenvolvimento. As atividades incluem a definição das tarefas, garantia da qualidade dos artefatos, modelagem do conhecimento, implementação e manutenção da ontologia, aquisição de conhecimento do domínio, avaliação técnica da ontologia e integração com ontologias existentes. O método também controla as mudanças de acordo com as versões da documentação, software e ontologia. Além disso, o método considera o reúso de ontologias existentes e ciclos iterativos de desenvolvimento.

OntoClean (Guarino *et al.*, 2000) é um método voltado à análise ontológica com base em uma ontologia formal de propriedades. Utiliza um conjunto central de meta-propriedades — identidade, unidade, rigidez e dependência — para estruturar essa análise. Embora contribua significativamente com uma fundamentação formal para a avaliação de ontologias, não se propõe a tratar do processo completo de sua construção.

Nicola, Missikoff & Navigli (2005) apresentam UPON, um método de engenharia de ontologias baseado no *Unified Process* (UP) e suportado pela *Unified Modelling Language* (UML), que adota uma abordagem iterativa e incremental. Ele possui quatro fases: *Inception*, *Elaboration*, *Construction* e *Transition*, cada uma com suas próprias iterações e workflows, incluindo *Requirements*, *Analysis*, *Design*, *Implementation* e *Test*. O objetivo de cada fase é capturar requisitos, realizar análises conceituais e semânticas,

modelar a ontologia e implementá-la em termos de componentes, além de realizar testes e lançar a ontologia. O método UPON segue uma abordagem metodológica rigorosa e disciplinada para o desenvolvimento de ontologias. Os autores falam sobre ausência de atividades padronizadas na construção e documentação de ontologias.

Pinto, Tempich & Staab (2009) apresentam a plataforma DILIGENT, que tem como objetivo facilitar a engenharia e evolução de ontologias em um ambiente distribuído. O artigo discute os desafios de se desenvolver ontologias em um ambiente distribuído e apresenta as características da plataforma DILIGENT, que incluem um conjunto de ferramentas para a criação e gerenciamento de ontologias e um modelo de colaboração baseado em agentes. O método segue cinco passos: *Build*, construção de uma ontologia consensual e compartilhada do domínio, contendo termos de alto nível e sem esgotar as possibilidades de representação; *Local Adaptation*, a ontologia inicialmente construída e distribuída de modo que seus usuários realizem mudanças locais específicas de acordo com a percepção local e registrem requisições de mudanças para a ontologia compartilhada; *Analysis*, as ontologias locais alteradas e as requisições enviadas são analisadas em um painel de controle a fim de definir quais mudanças devem fazer parte da nova versão da ontologia compartilhada; *Revise*, a ontologia compartilhada é revisada e atualizada a partir das requisições dos usuários a fim de mantê-la próxima das necessidades reais; e *Local Update*, os usuários da ontologia atualizam a ontologia local com a nova versão da ontologia compartilhada que incorpora as mudanças requeridas.

Segundo Suárez-Figueroa *et al.* (2012), o método NeOn é voltado à construção de redes de ontologias com reuso de recursos ontológicos e não ontológicos. Ele inclui nove cenários, um glossário de processos e atividades, diferentes ciclos de vida e um guia metodológico. Os cenários são:

1. Da especificação à implementação – construção desde a especificação até a implementação;
2. Reuso e reengenharia de recursos não ontológicos – uso de recursos não ontológicos que requerem reengenharia;
3. Reuso de recursos ontológicos – reuso direto de recursos ontológicos;
4. Reuso e reengenharia de recursos ontológicos – reuso de recursos ontológicos com necessidade de reengenharia;
5. Reuso e fusão de recursos ontológicos – combinação de dois ou mais recursos ontológicos;
6. Reuso, fusão e reengenharia de recursos ontológicos – uso de dois ou mais recursos ontológicos que necessitam de reengenharia;

7. Reúso de padrões de design de ontologias – uso de padrões de design ontológico;
8. Reestruturação de recursos ontológicos – uso de recursos ontológicos reestruturados (modularizado, reduzido, estendido ou especializado);
9. Localização de recursos ontológicos – usados na construção de redes em diferentes linguagens e culturas.

Esses cenários podem ser combinados, mas o primeiro é essencial e integra os demais. Durante todo o processo, é necessário realizar atividades como aquisição de conhecimento, documentação, gestão de configuração, avaliação e validação.

O método *Systematic Approach for Building Ontologies* (SABiO) (Falbo, 2014) apresenta uma abordagem sistemática para a construção de ontologias em domínios específicos. O método consiste em cinco fases principais, sendo elas: (I) identificação do propósito e elicitação dos requisitos, (II) captura e formalização da ontologia, (III) projeto, (IV) implementação e (V) teste. SABiO também considera processos de apoio (aquisição de conhecimento, reúso, documentação, gerência de configuração, avaliação) que são executados em paralelo ao processo de desenvolvimento. O método trata tanto o desenvolvimento de ontologias de referência quanto ontologias operacionais. SABiO considera que os seguintes papéis estão envolvidos no desenvolvimento de ontologias: (i) especialista de domínio, que é o especialista no domínio da ontologia e fornece o conhecimento que deve ser modelado e implementado na ontologia de domínio; (ii) usuário da ontologia, representando alguém que pretende usar a ontologia para um determinado propósito; (iii) engenheiro de ontologia, responsável pela ontologia de referência e, assim, pelas fases iniciais do processo de desenvolvimento da ontologia; (iv) projetista de ontologias, responsável pelo projeto de uma ontologia operacional; (v) programador de ontologias, responsável pela implementação de uma ontologia operacional; (vi) testador de ontologias, responsável por testar uma ontologia operacional (Falbo, 2014).

Mendonça & Soares (2017) apresentam o método OntoForInfoScience, que detalha os passos necessários do processo de construção de ontologias em uma linguagem simples e adequada a profissionais diversos, característica que a diferencia de outras metodologias. OntoForInfoScience engloba, ao todo, uma pré-etapa, onde o desenvolvedor realiza uma avaliação prévia sobre a real necessidade da construção de uma ontologia para o problema analisado, e mais oito etapas, sendo elas: 1) especificação da ontologia; 2) aquisição e extração de conhecimento; 3) conceitualização; 4) fundamentação da ontologia; 5) representação formal da ontologia através de uma linguagem lógica; 6) avaliação da ontologia; 7) documentação; e 8) disponibilização da ontologia.

Dentre os métodos pesquisados, destaca-se o SABiOx – *Extended Systematic Approach for Building Ontologies* (Aguar; Souza, 2024), pois foi o método escolhido para

realização deste trabalho. SABiOx tem o propósito de guiar a construção de ontologias (de qualquer domínio) de forma detalhada e apoiado por princípios ágeis, com ciclos iterativos e incrementais. O método é baseado no método SABiO (Falbo, 2014), ampliado com detalhes sobre o ciclo de vida da ontologia e suas atividades.

O método é formado por cinco fases, cada uma define um ciclo de vida de fase composto por um conjunto de atividades. Cada fase se preocupa com um aspecto diferente do processo de Engenharia de Ontologias. A fase de Requisitos elicit os requisitos para a ontologia, ou seja, o que ela pretende capturar/representar. A fase de Configuração define a linha de base (por exemplo, linguagem de modelagem, reutilização de outras ontologias, etc.) para a construção de modelos de ontologia. A fase de Captura identifica e modela a conceituação (ou seja, conceitos, relações, axiomas, etc.) para atender aos requisitos elicitados. A fase de Design especifica recursos tecnológicos (por exemplo, linguagem de codificação, reutilização de vocabulários existentes, etc.) para a implementação da ontologia. Finalmente, a fase de Implementação codifica a ontologia em uma linguagem operacional (por exemplo, OWL) (Aguiar; Souza, 2024).

As fases seguem uma sequência em que cada uma aproveita o que foi feito na anterior, permitindo que o ciclo de vida vá se desenvolvendo aos poucos. Com isso, o ciclo é pensado para apoiar a construção e o crescimento contínuo das ontologias ao longo do tempo. Ou seja, passar várias vezes por essas fases ajuda a fazer melhorias ou mudanças na ontologia. Em cada etapa do ciclo de vida, podem acontecer vários ciclos de vida, baseados em princípios ágeis inspirados na Engenharia de Software.

Aguiar & Souza (2024) indicam que cada uma das fases do SABiOx define um ciclo de vida de fase, composto por atividades principais, que consistem nos procedimentos principais para produzir o resultado daquela fase, e atividades de suporte, que fornecem procedimentos auxiliares para as atividades principais. As atividades de suporte são organizadas em seis grupos: (i) aquisição de conhecimento, que envolve buscar e extrair informações de diferentes fontes; (ii) documentação, que serve para registrar os resultados das outras atividades feitas no processo, como a criação de documentos de especificação, referência, catálogos de conceitos, visões e modelos da ontologia; (iii) gerenciamento de configuração, que cuida do controle de versões e de qualquer mudança nos artefatos criados; (iv) avaliação, que verifica a qualidade do que foi produzido, usando questões de competência, validação com instâncias de ontologia e testes; (v) reutilização, que aproveita materiais ontológicos já existentes, como textos, vocabulários e outras ontologias; e, por fim, (vi) publicação, que é a etapa em que a ontologia final é disponibilizada para uso.

A Figura 3 fornece uma visão geral das fases (círculos de linha sólida) e atividades (pontos sobre as linhas do círculo) do SABiOx.

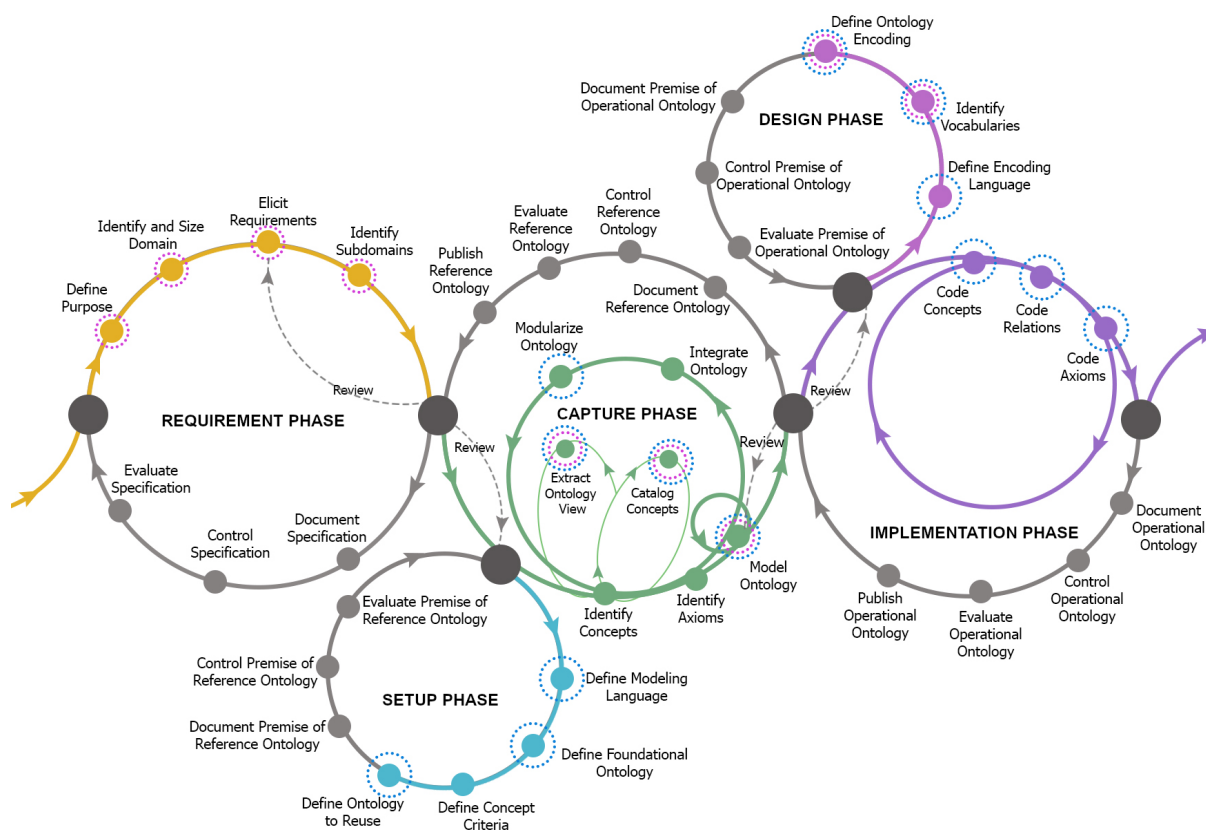


Figura 3 – Fases e atividades do método SABiOx.

## 2.3 Ferramentas para o Desenvolvimento de Ontologias

Nos últimos anos, pesquisadores propuseram diversas ferramentas para o desenvolvimento de ontologias. Ferramentas utilizadas para construir, editar, anotar e mesclar ontologias são chamadas de ferramentas de ontologia (Abburu; Babu, 2013).

Essas ferramentas desempenham um papel importante tanto nas etapas de desenvolvimento das ontologias (construção, anotação, mesclagem, etc.) quanto para o uso em aplicações (Laera; Tamma, 2005). Elas permitem que os usuários definam novos conceitos, relações e instâncias (Noy; Musen, 2002). Além da capacidade de importar e estender ontologias emergentes, as ferramentas de desenvolvimento podem conter alguns recursos adicionais, como navegação gráfica, busca e verificação de restrições.

Para um estudo mais completo foi realizado um mapeamento sistemático com o intuito de investigar ferramentas de desenvolvimento de ontologias. De acordo com Kitchenham, Budgen & Brereton (2011), um mapeamento sistemático (ou estudo exploratório) realiza um amplo estudo em um tópico de tema específico e visa identificar evidências disponíveis sobre esse tema.

Conforme discutido anteriormente, existe uma gama de ferramentas de construção de ontologias e é um desafio identificar as ferramentas que auxiliam no processo de

construção. Portanto, o objetivo deste mapeamento é identificar quais ferramentas são usadas para a construção da ontologia com base em diferentes estágios do processo.

Para obter informações para alcançar o objetivo, foram definidas treze questões de pesquisa (QPs) apresentadas na Tabela 1, junto com o *rationale* usado para considerá-las.

Tabela 1 – Questões de Pesquisa do mapeamento sistemático.

<b>ID</b>	<b>Questão</b>	<b><i>Rationale</i></b>
QP1	Onde foi publicado?	Identificação do meio de publicação (workshop, conferência ou revista).
QP2	Qual é o ano de publicação?	Visão temporal das publicações sobre o uso de ferramentas na construção de ontologias.
QP3	Que tipo de pesquisa/avaliação foi apresentada no artigo?	Identificação do tipo de pesquisa realizada na publicação, levando em consideração as categorias definidas em (Wi-eringa <i>et al.</i> , 2006), a saber: Pesquisa de Avaliação; Proposta de Solução; Pesquisa de Validação; Artigo Filosófico; Artigo de Opinião; e Artigo de Experiência Pessoal.
QP4	Que ferramenta(s) é(são) citada(s) no artigo?	Identificação de ferramentas.
QP5	Quais são as funcionalidades das ferramentas encontradas?	Identificação das funcionalidades das ferramentas encontradas.
QP6	Em que fase do processo de construção da ontologia o artigo indica o uso da(s) ferramenta(s)?	Identificação em quais fases do desenvolvimento da ontologia as ferramentas estão sendo utilizadas.
QP7	Qual é a disponibilidade da ferramenta?	Identificação das características da ferramenta, tais como se é gratuita ou paga, se é de código aberto ou fechado, se está disponível ou não mais online, etc.
QP8	Quais são as limitações da(s) ferramenta(s)?	Identificação de possíveis limitações que restringem o uso da ferramenta.
QP9	Como a(s) ferramenta(s) foi(foram) avaliada(s)?	Identificação de testes/avaliações realizados nas ferramentas mencionadas.
QP10	Que documentação (além do artigo) acompanha a(s) ferramenta(s)?	Identificação se a ferramenta citada possui documentação específica.

QP11	Qual é o público-alvo/nível de experiência exigido do usuário da(s) ferramenta(s)?	Identificação para qual nível de experiência a ferramenta citada foi criada.
QP12	A ferramenta fornece APIs para facilitar a integração com outras?	Identificação de facilidades para integração da ferramenta citada com outras ferramentas.
QP13	Quais são os benefícios da(s) ferramenta(s)?	Identificar nos artigos se essas informações são explícitas.

A expressão de busca adotada neste estudo foi desenvolvida considerando-se a busca por termos que remetem a ferramentas de construção de ontologias e termos de limitação de área, tipo de documento e linguagem. Após diversos testes com diversas expressões diferentes, a seguinte expressão de busca foi utilizada:

TITLE-ABS-KEY ( ( editor OR environment OR tool ) AND ( "ontolog\* development"OR "ontolog\* build\*"OR "ontolog\* engineering"OR "ontolog\* construction"OR "ontolog\* implementation"OR "ontolog\* evaluation"OR "ontology-driven" ) AND ( LIMIT-TO ( SUBJAREA , "COMP" ) ) AND ( LIMIT-TO ( DOCTYPE , "ar" ) OR LIMIT-TO ( DOCTYPE , "cp" ) ) AND ( LIMIT-TO ( LANGUAGE , "English" ) ).

A biblioteca digital usada como fonte das publicações para esse estudo foi a *Scopus* ([www.scopus.com](http://www.scopus.com)). Foi selecionada essa biblioteca, pois têm demonstrado boa cobertura de publicações em estudos secundários registrados na literatura.

A proposta foi selecionar as publicações em quatro etapas sendo elas: 1) Seleção Primária e Catalogação; 2) Remoção de Duplicações; 3) Seleção de Publicações Relevantes – 1º filtro e 4) Seleção de Publicações Relevantes – 2º filtro. No entanto, o estudo foi realizado apenas até a etapa 3. As publicações selecionadas ao longo do procedimento de seleção das publicações foram registradas em uma planilha, onde encontra-se armazenado o ano e o título da publicação. Esse estudo foi realizado com a colaboração de uma aluna de mestrado que pesquisa sobre o mesmo tema e duas alunas de doutorado com pesquisas relacionadas, sob a supervisão dos orientadores e da coorientadora.

A primeira etapa — Seleção Primária e Catalogação — consistiu em aplicar a expressão de busca, apresentada anteriormente, usando o mecanismo de busca da biblioteca scopus. Nessa etapa foram retornados 1.954 trabalhos.

Na segunda etapa — Remoção de Duplicações — buscamos estudos indexados por mais de uma vez. Foram identificados vinte e seis trabalhos duplicados que foram removidos, restando um total de 1.928 estudos.

Na terceira etapa — Seleção de Publicações Relevantes – 1º filtro — o título,

resumo e palavras-chave das publicações selecionadas foram analisados, considerando-se o seguinte critério de inclusão (CI): O estudo apresenta (não apenas menciona) uma ou mais ferramentas de construção de ontologias. Considerou os seguintes critérios de exclusão: (CE1) Não fornece um resumo; (CE2) É apenas um resumo; (CE3) Não está escrito em inglês; (CE4) É uma cópia ou uma versão mais antiga de outra publicação que já foi considerada; (CE5) Não é um estudo primário (por exemplo, editoriais, resumos de palestras, tutoriais, etc.); (CE6) Não é possível ter acesso à versão completa da publicação. Foram analisados os artigos levantados, buscando atender os critérios de inclusão e exclusão. Essa etapa do mapeamento foi muito útil para o levantamento de ferramentas existentes na literatura que são apresentadas neste capítulo e para embasamento em outras partes deste trabalho.

De acordo com [Gómez-Pérez, Fernández-López & Corcho \(2004a\)](#), as ferramentas de suporte às atividades de engenharia de ontologias podem ser agrupadas em seis categorias:

1. Ferramentas de desenvolvimento: incluindo ferramentas e suítes integradas usadas para construir uma nova ontologia do zero. Além das funções comuns de edição e navegação, essas ferramentas geralmente oferecem suporte à documentação de ontologias, exportação e importação de ontologias de/para diferentes formatos e linguagens de ontologias, edição gráfica de ontologias, gerenciamento de bibliotecas de ontologias, etc.;
2. Ferramentas de avaliação: usadas para avaliar o conteúdo das ontologias e suas tecnologias relacionadas. A avaliação de conteúdo de ontologias busca reduzir problemas quando precisamos integrar e utilizar ontologias e tecnologias baseadas em ontologias em outros sistemas de informação;
3. Ferramentas de mesclagem e alinhamento: dão suporte à integração de duas ou mais ontologias, consolidando as informações comuns entre ambas e criando uma nova ontologias com os termos unificados; também mapeiam os conceitos de uma ontologia para uma outra sem necessariamente juntá-las, criando correspondência entre elementos semelhantes ou equivalentes;
4. Ferramentas de anotação: ajudam a adicionar instâncias de conceitos e relações e a manter marcações baseadas em ontologias (semi)automaticamente em páginas da Web;
5. Ferramentas de consulta e mecanismos de inferência: permitem fazer perguntas para ontologia, buscando informações que foram organizadas dentro dela; e descobrir novas informações que não foram diretamente registradas mas que podem ser concluídas com base em regras e relações definidas na ontologia. Normalmente, estão fortemente relacionadas à linguagem usada para implementar ontologias;

6. Ferramentas de aprendizagem: derivam ontologias (semi) automaticamente de textos em linguagem natural, bem como fontes e bancos de dados semiestruturados, por meio de técnicas de aprendizado de máquina e análise de linguagem natural.

Essa pesquisa de mestrado está focada no primeiro grupo, as ferramentas de desenvolvimento de ontologias, embora há casos em que uma ferramenta ou suíte possa atender a mais de um grupo de atividades ao mesmo tempo (Gómez-Pérez; Fernández-López; Corcho, 2004a).

Editores de ontologias são ferramentas importantes para auxiliar na elaboração de ontologias. Eles facilitam o desenvolvimento e o gerenciamento de ontologias, a definição e a modificação de conceitos, propriedades, axiomas e restrições, e alguns deles permitem a inspeção e a navegação em ontologias (Nieto, 2003). A seguir estão listados alguns exemplos de ferramentas de construção de ontologias organizados por Mendonça & Castro (2021):

Tabela 2 – Lista alfabética de editores de ontologia (Mendonça *et al.*, 2021).

<b>Editor</b>	<b>Características Principais</b>	<b>Referência</b>
<i>Cmap Tools</i>	Editor de mapeamento conceitual. Permite representar nós e mapas conceituais em ambiente gráfico.	<a href="https://cmap.ihmc.us/cmaptools/">https://cmap.ihmc.us/cmaptools/</a>
<i>Eddy for Graphol</i>	Editor que usa a linguagem gráfica Graphol. Desenvolve grafos compostos por nós e conexões evitando textos.	<a href="https://www.obdasystems.com/eddy">https://www.obdasystems.com/eddy</a>
<i>Graffo</i>	Editor de código fonte aberto para a apresentação de classes, propriedades e restrições em OWL.	<a href="https://essepuntato.it/graffoo/">https://essepuntato.it/graffoo/</a>
<i>Hozo</i>	Editor, servidor e gerenciador de ontologias em ambiente distribuído; baseado na Web para colaboração.	<a href="http://www.hozo.jp/">http://www.hozo.jp/</a>
<b>ICOM</b>	Permite manipular múltiplas ontologias; interface gráfica com suporte a diagrama de classes e raciocinador.	<a href="https://www.inf.unibz.it/franconi/icom/">https://www.inf.unibz.it/franconi/icom/</a>
<i>Menthor</i>	Editor de código aberto, permite integração com a OntoUML, estereotipos e importação do Ent. Architect.	<a href="https://ontouml.org/ontouml/tooling/">https://ontouml.org/ontouml/tooling/</a>

<i>NeOn Toolkit</i>	Inclui metodologia para desenvolvimento de ontologias em larga escala em ambiente distribuído.	<a href="http://neon-project.org/">http://neon-project.org/</a>
<i>OBO-Edit</i>	Editor de código aberto, codificado em Java. Otimizado para o formato ontológico OBO.	<a href="http://oboedit.org/">http://oboedit.org/</a>
<i>OLED</i>	Editor baseado na linguagem OntoUML, fundamentada na UFO; oferece validação, simulação e detecção de padrões.	<a href="https://ontouml.org/ontouml/tooling/">https://ontouml.org/ontouml/tooling/</a>
<i>OWL GrEd</i>	Interface gráfica para editar e visualizar ontologias, extensa o por plugins inclusive para modelagem UML.	<a href="http://owlgred.lumii.lv/">http://owlgred.lumii.lv/</a>
<i>Protégé</i>	Editor de código aberto com interface para criação e edição extensão por plugins mais processador de consultas.	<a href="https://protege.stanford.edu/">https://protege.stanford.edu/</a>
<i>SWOOP</i>	Editor de código aberto com suporte a OWL de arquitetura baseada em Web. Suporte a raciocínio.	<a href="http://www.mindswap.org/">http://www.mindswap.org/</a>
<i>TopBraid Composer</i>	IDE comercial para modelagem RDF em interface gráfica. Tem capacidade de inferência, e consultas.	<a href="https://www.topquadrant.com/products/topbraidcomposer/">https://www.topquadrant.com/products/topbraidcomposer/</a>
<i>yEd Graph Editor</i>	Editor múltiplo comercial, permite trabalhar com grafos de redes semânticas e OWL.	<a href="https://www.yworks.com/products/yed">https://www.yworks.com/products/yed</a>
<i>Web Protégé</i>	Versão colaborativa na Web do Protege . Suporta edições em OWL e OBO; importa e exporta formatos.	<a href="https://protege.stanford.edu/">https://protege.stanford.edu/</a>

Diante da variedade de ferramentas existentes para apoiar na construção de ontologias, ainda é necessário entender quais combinações de ferramentas oferecem o suporte mais adequado para todo o ciclo de desenvolvimento. Com esse objetivo, realizamos um *survey* para obter evidências sobre o uso de ferramentas para dar suporte à EO. Investigamos quais ferramentas têm sido usadas, suas principais características, benefícios e dificuldades

percebidos pelos usuários, bem como propriedades consideradas mais relevantes pelos mesmos. Os detalhes sobre a condução desse *survey* e os resultados obtidos estão apresentados no Capítulo 3.

## 2.4 Trabalhos Relacionados

Na literatura existem diversos trabalhos abordando ferramentas para Engenharia de Ontologias. Alguns deles propõem ferramentas para apoiar algum método específico, outros discutem diferentes métodos e ferramentas, e há também estudos que realizam comparação entre diversas ferramentas existentes e suas funcionalidades. Esses trabalhos são brevemente descritos nesta seção.

[Almeida \(2003\)](#) apresenta um roteiro para a construção de ontologias bibliográficas utilizando uma ferramenta automatizada, mostrando a importância dessa técnica para a organização e integração de informações em diferentes domínios, porém o que apresenta são somente ferramentas de edição e navegação de ontologias, não atendendo a outras fases de um processo de construção.

[Nieto \(2003\)](#) apresenta um relatório técnico com uma visão abrangente sobre o conceito de ontologias, suas características, funções e aplicações. O documento apresenta também as principais etapas para construção de ontologias e descreve linguagens e ferramentas disponíveis, trazendo detalhes e informações relevantes sobre elas. A autora também discute brevemente alguns projetos de pesquisa desenvolvidos, identificando os avanços em pesquisas na área e as contribuições.

O trabalho de [Gangemi \(2003\)](#) traz uma visão detalhada das metodologias e ferramentas desenvolvidas pelo laboratório OntoLab, destacando, por exemplo, a ontologia fundacional DOLCE e a metodologia OntoClean. O autor discute como essas abordagens auxiliam na definição precisa dos conceitos e relações, promovendo ontologias com maior consistência e clareza semântica. Além disso, ressalta a importância de ferramentas que permitam não apenas a modelagem, mas também a análise e a integração de ontologias em sistemas diversos.

[Corcho, Fernández-López & Gómez-Pérez \(2003\)](#) realizam uma análise e comparação de metodologias, ferramentas e linguagens utilizadas na construção de ontologias. O trabalho destaca que, embora existam diversos recursos disponíveis, ainda não há um ambiente de desenvolvimento integrado que atenda a todas as necessidades do processo ontológico, como construção, evolução, avaliação e manutenção. Os autores abordam a evolução das definições de “ontologia”, discutem diferentes abordagens para construí-las e avalia as capacidades e limitações das ferramentas e linguagens existentes, concluindo que um ambiente de trabalho unificado é crucial para suportar não apenas diferentes linguagens e metodologias, mas também facilitar a interoperabilidade entre aplicações.

Correndo & Alani (2007) discutem a evolução das ferramentas voltadas para a construção colaborativa de ontologias, impulsionadas pelos princípios da Web 2.0 e pela demanda por representações semânticas na Web Semântica. O artigo destaca a necessidade de ferramentas que apoiem a criação colaborativa de ontologias pois a maioria das ferramentas para construção de ontologias existentes não oferece suporte para colaboração distribuída. A pesquisa detalha diversas ferramenta comparando suas funcionalidades em termos de edição de modelos, formalidade, suporte a instâncias e tags, e recursos para discussão e consenso entre os usuários. O objetivo principal é avaliar o estado da arte dessas ferramentas e entender como elas podem evoluir para atender às expectativas dos usuários na construção de conhecimento estruturado de forma colaborativa.

Subhashini & Akilandeswari (2011) apresentam uma revisão dos principais softwares disponíveis para a construção de ontologias. As ontologias são representações formais de conhecimento em um domínio específico, que podem ser usadas para facilitar a recuperação de informações, a interoperabilidade entre sistemas e a tomada de decisões automatizadas. Cada ferramenta é apresentada com uma descrição de suas principais características, vantagens e desvantagens, bem como exemplos de uso e um resumo das principais publicações que a mencionam.

Abburu & Babu (2013)<sup>1</sup> apresentam um levantamento das principais ferramentas para construção de ontologias, que são essenciais para criar ou reutilizar ontologias em um formato legível por máquina, classificando-as em duas categorias: ferramentas dependentes de linguagem (como Ontolingua, WebOnto, OilEd) e ferramentas independentes de linguagem (como Protégé, TopBraid Composer, NeOn Toolkit). O trabalho também compara essas ferramentas com base em critérios como suporte a múltiplos usuários, recursos de consistência, suporte gráfico, entre outros, oferecendo um panorama útil para quem busca selecionar ferramentas adequadas às suas necessidades. Apesar desse trabalho se assemelhar à pesquisa relatada nesta dissertação, o nosso objetivo busca apresentar ferramentas que atendam um método completo, cobrindo todo o processo de desenvolvimento de uma ontologia.

O trabalho de Dudáš, Zamazal & Svátek (2014) apresenta um mapeamento detalhado do panorama de ferramentas de visualização de ontologias, articulando esse levantamento com as diferentes necessidades e contextos de uso dos usuários. Este artigo aborda a visualização de ontologias, sendo como uma etapa essencial não apenas para o compartilhamento, mas também para o desenvolvimento e a inspeção das ontologias. Os autores realizaram uma análise abrangente de diversas ferramentas, categorizando-as com

---

<sup>1</sup> Apesar deste artigo atualmente não ser mais encontrado em buscas na Web, a autora mantém uma cópia do artigo obtida quando o mesmo encontrava-se disponível, que pode ser acessada em <https://ufes.es/abburu-babu-injasr2013>. Atualmente, o site dos arquivos do periódico (<https://injasr.org/Archive.html>) elenca as diferentes edições do mesmo, porém os links estão quebrados (apontam para resultados 404 – Não encontrado).

base em cenários de uso (como edição, inspeção, aprendizado e compartilhamento) e em suas funcionalidades. O estudo fornece à comunidade acadêmica e profissional um roteiro de ferramentas de visualização, cujo qual foi avaliado através de um questionário aplicado a especialistas da área.

No contexto do *Ontology Summit 2013*, [Neuhaus et al. \(2013\)](#) propõem diretrizes para usuários e desenvolvedores acerca da avaliação de ontologias. Os autores destacam como problema encontrado a falta de uma metodologia padronizada para o desenvolvimento de ontologias e falta de consenso na avaliação das mesmas. Baseado em práticas estabelecidas em engenharia de sistemas e engenharia de software consideram cinco aspectos de qualidade das ontologias: inteligibilidade, fidelidade, artesanato, adequação e implantabilidade. Os autores reforçam a importância de um ambiente integrado para o desenvolvimento de ontologias para que a avaliação possa ser feita em todas as fases do ciclo de desenvolvimento da ontologia. O trabalho se assemelha a minha pesquisa por levantar a mesma problemática: a inexistência de ferramentas que apoiem integralmente o ciclo de vida, desde a modelagem até a manutenção das ontologias, porém o estudo é direcionado para a avaliação do processo de desenvolvimento de ontologias.

O artigo de [Vigo et al. \(2014\)](#) analisa as dificuldades enfrentadas por autores de ontologias, sobretudo na utilização de ferramentas. Por meio de entrevistas com especialistas, os autores identificam a fragmentação das ferramentas existentes e a um entendimento das eficácia das ferramentas para tarefas comuns de autoria como criação, exploração, reutilização, comparação de versões, depuração e testes. Uma das principais conclusões é que há uma tensão entre dois estilos de desenvolvimento: definição formal versus hierarquias construídas manualmente. O artigo sugere recomendações de design para ferramentas futuras, com foco em torná-las mais acessíveis tanto para especialistas quanto para usuários novatos.

[Khondoker & Mueller \(2010\)](#) conduziram uma pesquisa através de questionário, aplicado para 32 ontologistas, para compreender quais ferramentas são mais utilizadas por desenvolvedores de ontologias e quais são suas percepções sobre essas ferramentas, tarefas e processos. O estudo revelou que embora existam muitas opções disponíveis, os participantes escolheram apenas algumas delas, a maioria ferramentas específicas de EO, com o Protégé sendo o predominante. O estudo fornece um feedback para desenvolvedores de ferramentas e de ontologias, a fim de aprimorar futuras soluções, segundo os autores, estudos sobre ferramentas e experiência de usuários é um tema pouco abordado ainda.

A análise dos trabalhos relacionados evidencia que, embora haja uma variedade significativa de ferramentas e métodos para desenvolvimento de ontologias, ainda existe uma lacuna na oferta de ambientes integrados que conciliem usabilidade, suporte colaborativo e que possam oferecer suporte de ferramentas do início ao fim do processo de desenvolvimento de ontologias. Essa observação mostra a relevância em estudos que possam

identificar e investigar ferramentas que possam ser integradas e acompanhar todo o ciclo de desenvolvimento, possibilitando que o processo se torne mais fácil, prático e confiável.

### 3 *Survey*: Uso de ferramentas em Engenharia de Ontologias

*Surveys* têm como objetivo identificar as características de uma população ampla, extrapolando os dados obtidos de uma amostra representativa de indivíduos (Easterbrook *et al.*, 2008). São utilizados para gerar uma visão abrangente da situação e retratar o estado atual (Wohlin *et al.*, 2012).

Buscando identificar e examinar o uso de ferramentas em Engenharia de Ontologias para compreender melhor esse tema em termos práticos, foi realizado um *survey* (Wohlin *et al.*, 2012). A opção por esse método foi devido à sua capacidade de alcançar um grande número de pessoas, algo que seria difícil ou inviável por meio de entrevistas ou estudos de caso, por exemplo.

Este *survey* e a análise dos dados coletados foi realizado em colaboração com mais quatro alunas: Glaice Kelly da S. Quirino, Jordana S. Salamon, Rosiane da Silva Sangali e Isaufânia S. R. Timóteo. Todos os co-autores envolvidos possuem interesse na área e trabalham com esse tema em suas pesquisas. Deste estudo temos como resultado um artigo publicado no *Proceedings of the 17th Seminar on Ontology Research in Brazil* (ONTOBRAS 2024) (Gomes *et al.*, 2024a).

Para realização deste *survey*, seguimos o processo delineado por Wohlin *et al.* (2012). Sendo assim, o estudo foi dividido em cinco etapas principais: delimitação, onde determinamos o escopo e os objetivos do estudo; planejamento, com a definição do desenho do estudo; execução, que abrangeu a coleta dos dados; análise e interpretação, envolvendo o exame das informações para obter conclusões; e, por fim, apresentação e consolidação, em que os resultados foram comunicados.

Com base nos objetivos traçados, estabelecemos três questões centrais de pesquisa: **(RQ1)** Quais ferramentas são utilizadas e quais são suas principais características? **(RQ2)** Quais benefícios e dificuldades foram identificados durante o uso dessas ferramentas? **(RQ3)** Quais atributos são considerados essenciais em ferramentas voltadas para EO? Definir as questões de pesquisa foi importante para formular o questionário, pois os dados que seriam coletados deveriam responder a essas perguntas. Utilizamos um formulário eletrônico como instrumento de pesquisa, desenvolvido no Google Forms. Este foi estruturado em três seções, precedidas por um termo de consentimento para participação. A primeira seção, composta por oito perguntas fechadas, visava caracterizar os participantes. A segunda seção continha 11 perguntas fechadas relacionadas às RQ1 e RQ2, com uma questão aberta para complemento textual. A terceira seção, voltada para a

RQ3, incluía uma pergunta fechada e duas abertas, com o objetivo de coletar informações adicionais, comentários e sugestões. Algumas perguntas empregaram escalas ordinais de concordância, sem a opção neutra, para obter respostas mais conclusivas. O formulário foi disponibilizado como parte do pacote de dados do estudo (Gomes *et al.*, 2024b).

A amostra foi composta por engenheiros de ontologia, selecionados por sua experiência e conhecimento no desenvolvimento de ontologias e no uso de ferramentas de suporte. O procedimento do estudo envolveu três fases: na primeira fase, um piloto foi conduzido para testar o formulário e o protocolo do estudo. Um especialista em ontologias foi convidado a responder o questionário com a função de analisar tempo de resposta e qualidade do questionário, ele respondeu o mesmo fornecendo feedback sobre o tempo necessário, possíveis dificuldades e sugestões de melhoria. Com base nesse retorno, realizamos ajustes no formulário. A segunda fase consistiu no envio de convites para a participação, direcionados a grupos de pesquisa, listas de discussão, redes de contatos em instituições acadêmicas e profissionais, além de autores de artigos relevantes. Também incentivamos os participantes a compartilharem o convite com outros possíveis interessados. Por fim, na terceira fase, realizamos a coleta de dados por meio dos formulários preenchidos, organizamos as informações em tabelas e gráficos e realizamos a análise dos resultados.

Os convites para participar da pesquisa foram enviados em dois momentos: o primeiro no final de dezembro de 2023 e o segundo no final de janeiro de 2024. No total, entramos em contato com 227 pessoas, obtendo 74 respostas até 1º de março de 2023, o que corresponde a uma taxa de resposta de 32,6%.

### 3.1 Condução do *Survey* e Síntese dos Dados

Para perguntas onde os participantes podiam selecionar mais de uma resposta, os valores absolutos somam mais do que 74, e os percentuais ultrapassam 100%. Para simplificar, os valores percentuais foram arredondados à primeira casa decimal. O conjunto completo de dados, incluindo tabelas e gráficos, está disponível no pacote do estudo (Gomes *et al.*, 2024b).

O perfil dos respondentes foi identificado por meio de perguntas sobre sua formação acadêmica, conhecimento sobre ontologias e experiência em Engenharia de Ontologias (EO). Também investigamos o contexto em que os participantes trabalharam com ontologias (se acadêmico ou industrial), os tipos de ontologias desenvolvidas (de referências ou operacionais) e a frequência com que utilizavam metodologias sistemáticas de EO.

A maior parte dos participantes é do Brasil (50; 67,6%), seguida por Espanha (8; 10,8%), Rússia (4; 5,4%), Holanda (3; 4,1%), Argentina (3; 4,1%) e França (2; 2,7%), entre outros países. Um respondente não informou sua nacionalidade. A maioria dos participantes possui formação em áreas relacionadas à Tecnologia da Informação, como Ciência da

Computação (73%), Engenharia da Computação (8,1%) e Sistemas de Informação (10,8%). Um caso excepcional foi um participante com formação em Serviço Social que também possui graduação em Ciência da Computação. A maior parte atua no contexto acadêmico (70,3%) e tem experiência de cinco anos ou mais em EO (62,2%), com um nível elevado de especialização na área (58,1%). Em relação aos tipos de ontologias desenvolvidas, 20,3% dos participantes criaram apenas ontologias operacionais, 32,4% apenas ontologias de referência, e 47,3% desenvolveram ambos os tipos. Além disso, 60,8% utilizam metodologias sistemáticas frequentemente, enquanto 21,6% afirmaram empregá-las sempre. A Figura 4 apresenta dados sobre os participantes.

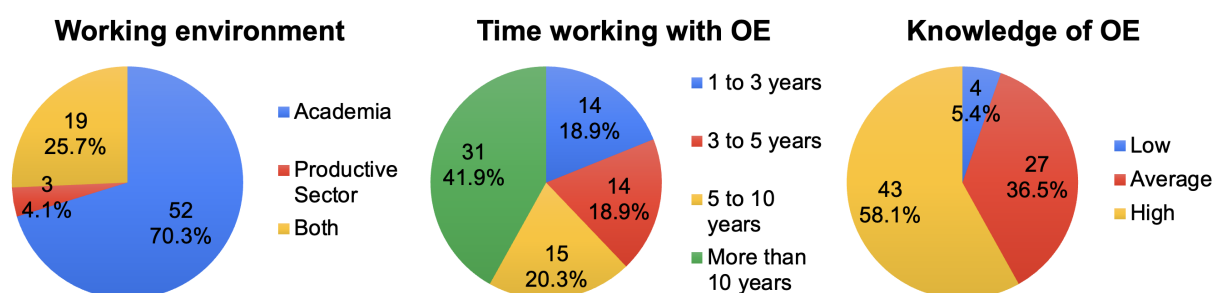


Figura 4 – Perfil dos Participantes.

Para uma melhor organização, os dados foram sintetizados e as respostas do questionário foram divididas em quatro grandes categorias: Uso de ferramentas em EO e Características das ferramentas de EO (esses dois tópicos vinculados à RQ1), Percepções sobre o uso de ferramentas de EO (vinculado à RQ2) e Propriedades das ferramentas de EO (este vinculado à RQ3). A seguir são apresentados os detalhes sobre cada um dos tópicos:

## Uso de ferramentas em EO

Neste tópico, perguntamos quais ferramentas os participantes usaram, com que frequência as usaram, como encontraram as ferramentas e em quais atividades de EO eles usaram ferramentas específicas de EO (ou seja, ferramentas que abordam tarefas de EO, como editores de ontologia) e genéricas (ou seja, ferramentas independentes de domínio que abordam tarefas gerais, como editores de texto). Nessa etapa do levantamento de ferramentas o objetivo foi fazer um comparativo com a etapa realizada anteriormente, na qual foram pesquisadas as ferramentas citadas na revisão de literatura. Agora, conhecendo as ferramentas de uso no cotidiano dos ontologistas, foi possível observar se a utilização se relaciona com a literatura. Para atender esse objetivo iniciou perguntando sobre a frequência com que usam ferramentas para dar suporte à EO, a maioria dos participantes (55,4%) informou que sempre usa ferramentas, 39,2% indicaram que usam ferramentas na maioria das vezes e 5,4% raramente as usam. Em relação às ferramentas usadas, as ferramentas específicas de EO foram predominantes.

Protégé (85,1%) foi a ferramenta mais usada, seguida por Visual Paradigm (43,2%), Astah (41,9%) e OntoUML-vp-plugin (36,5%). Protégé é uma plataforma gratuita e de código aberto que fornece uma interface gráfica para o desenvolvimento de ontologias e uma arquitetura com um conjunto de ferramentas para construir modelos de domínio e aplicações baseadas em conhecimento com ontologias (Guerson *et al.*, 2015). Visual Paradigm e Astah são ferramentas que dão suporte à criação de modelos conceituais e gerar código a partir dos modelos. OntoUML-vp-plugin,<sup>1</sup> por sua vez, é um plugin projetado para modeladores OntoUML que adiciona recursos ao Visual Paradigm para permitir o uso de estereótipos OntoUML em diagramas de classe, verificação de modelo e transformação. Outras ferramentas específicas de EO usadas pelos participantes foram OLED (Guerson *et al.*, 2015) (20,3%) e Menthor (Moreira *et al.*, 2016) (16,2%). Ambos são editores OntoUML que dão suporte à construção de modelos e verificação formal, incorporando suporte para uma variedade de recursos metodológicos para desenvolver, avaliar e implementar ontologias. Ferramentas genéricas também foram usadas, principalmente Planilhas Eletrônicas (27%), Editores de Texto (27%) e Ferramentas de Slide/Apresentação (24,3%). Além das ferramentas listadas na figura, outras foram mencionadas pelos participantes, como OntoTrack, Onto4AllEditor, Diagram.io, Miro, Widoco, Astrea, OnToology, Hermit, Themis, CmapTools, ONTOFOX, Enterprise Architect, PlantUML, Coggle.it e yEd Graph Editor.

Quanto à forma como os participantes descobriram as ferramentas, a maioria (82,1%) recebeu recomendações de outras pessoas (por exemplo, amigos, colegas de trabalho/estudo, professores). 39,2% descobriram as ferramentas em cursos ou aulas; 37,8% pesquisando na Internet; 36,5% usaram ferramentas empregadas onde trabalham, 35,1% usaram ferramentas desenvolvidas pela organização em que trabalham; e 17,6% declararam ter usado ferramentas indicadas em abordagens sistemáticas de EO. Para investigar o suporte de ferramentas fornecido às fases de desenvolvimento de ontologias, pedimos aos participantes que indicassem as fases que foram suportadas por ferramentas de EO e o tipo de ferramentas usadas (específicas de EO ou genéricas). A Especificação de Requisitos foi suportada principalmente por ferramentas genéricas (89,2% dos participantes indicaram que usaram esse tipo de ferramenta nesta fase). As outras fases tiveram predominância de ferramentas específicas de EO: 64,9% em Conceitualização e Formalização de Ontologias; 71,2% em Design de Ontologias, 73% em Implementação de Ontologias e 58,1% em Verificação, Validação e Testes. A Figura 5 resume dados sobre o uso de ferramentas em EO.

<sup>1</sup> <<https://github.com/OntoUML/ontouml-vp-plugin>>

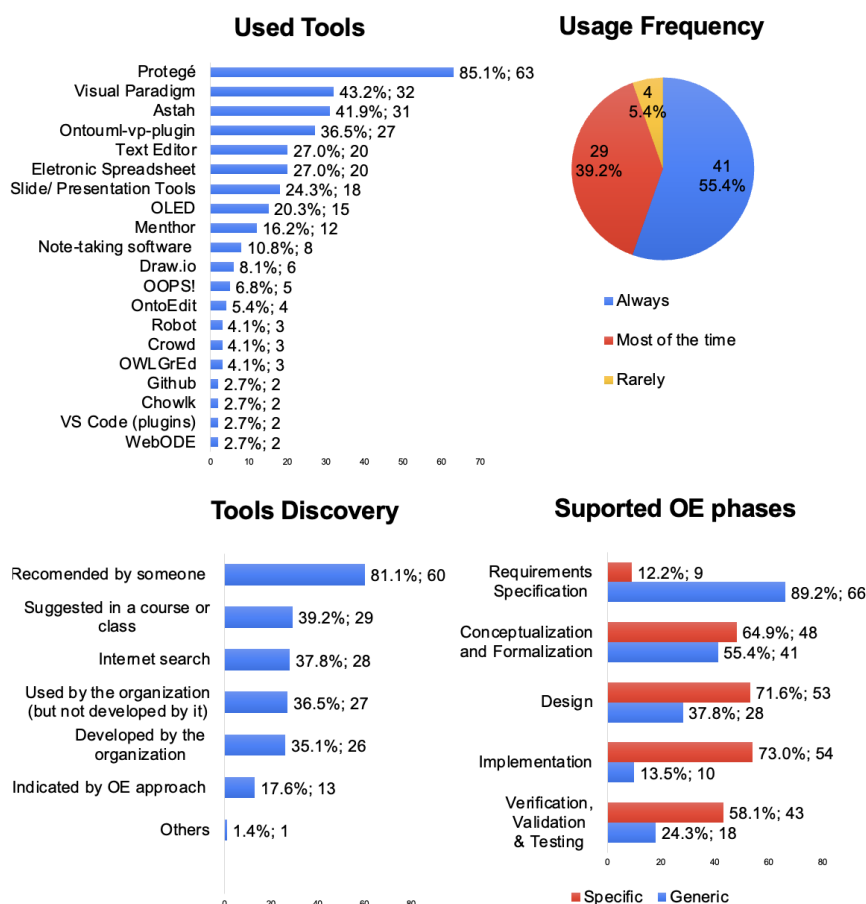


Figura 5 – Uso das Ferramentas em EO.

### Características das ferramentas de EO

Neste tópico, perguntamos aos participantes sobre a disponibilidade, funcionalidades e suporte integrado das ferramentas para EO. Em relação à disponibilidade, a maioria dos participantes (63,5%) declarou ter usado apenas ferramentas gratuitas, 33,8% informaram que algumas das ferramentas que usaram são gratuitas e 2,7% dos participantes não usaram ferramentas gratuitas. Em relação às funcionalidades, fornecemos uma lista de opções para os participantes indicarem quais foram oferecidas pelas ferramentas que usaram. As funcionalidades mais indicadas foram representação da hierarquia de conceitos da ontologia (86,5%), visualização da ontologia (85,1%), criação de diagramas (82,4%), verificação da ontologia (59,5%), implementação da ontologia operacional (56,8%) e validação da ontologia (50%). Também perguntamos aos participantes se as ferramentas forneceram suporte integrado para o processo de EO. 47,3% dos participantes declararam que as ferramentas fornecem suporte integrado parcialmente; 41,9% informaram que as ferramentas não fornecem suporte integrado e 10,8% afirmaram que as ferramentas fornecem suporte integrado ao processo de EO. A Figura 6 resume os resultados.

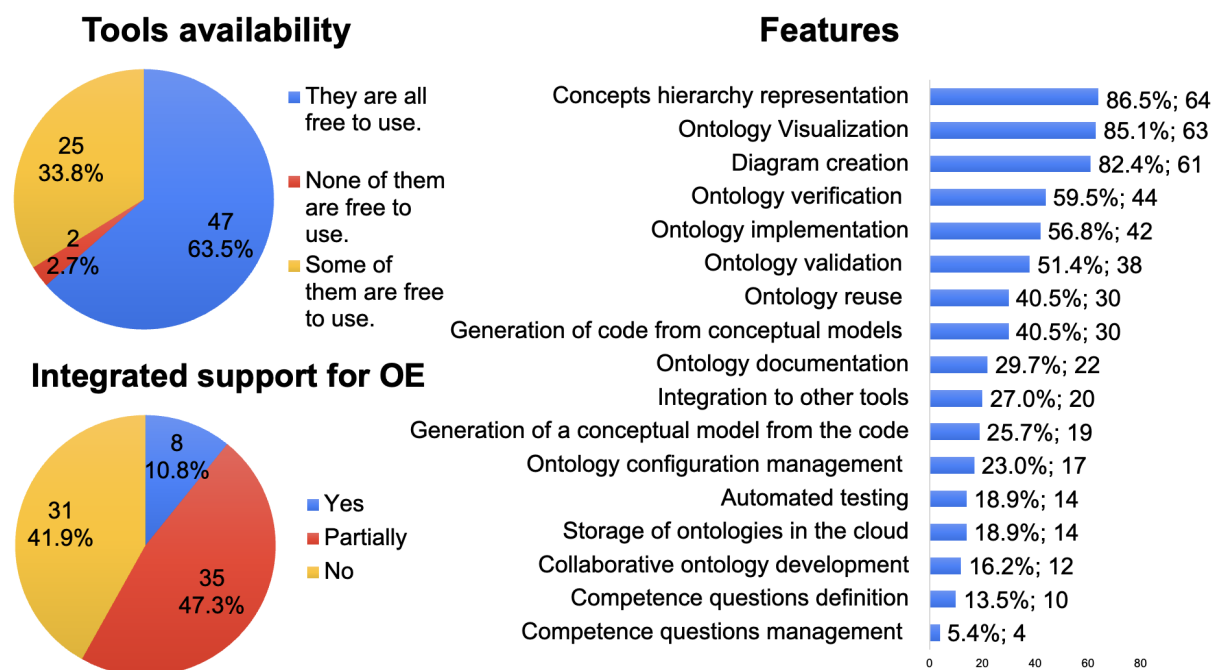


Figura 6 – Características das Ferramentas.

### Percepções sobre o uso de ferramentas de EO

Neste tópico, perguntamos aos participantes sobre suas percepções sobre o uso de ferramentas de EO em relação à utilidade das ferramentas, aos benefícios que elas proporcionaram e às dificuldades enfrentadas. Em relação à utilidade das ferramentas, 43,2% dos participantes declararam que elas ajudaram muito; 39,2% responderam que ajudaram pouco; 16,2% declararam que ajudaram muito pouco; e um participante respondeu que as ferramentas não ajudaram em nada. Foi pedido aos participantes que justificassem suas respostas. Agrupamos as respostas com o mesmo significado e as categorizamos. Em relação aos motivos pelos quais os participantes consideraram as ferramentas úteis, eles destacaram que as ferramentas ajudaram a simplificar o desenvolvimento de ontologias complexas (13,5%), avaliar a ontologia (12,2%), automatizar tarefas repetitivas (10,8%), visualizar e entender a ontologia como um todo (8,1%), aumentar a produtividade (8,1%), comunicar e interagir com as partes interessadas (2,7%), gerenciar o conhecimento (2,7%) e reutilizar (2,7%). Quanto aos motivos pelos quais os participantes não consideraram as ferramentas úteis, eles apontaram que as ferramentas são muito “fragmentadas” e todo o processo de EO não pode ser suportado por uma única ferramenta ou um conjunto de ferramentas integradas (29,7%). Além disso, houve falta de suporte, documentação, tutoriais ou outros recursos educacionais para ajudá-los a usar as ferramentas (17,6%), a automação foi limitada (5,4%), houve problemas de desempenho (5,4%), falta de acesso livre (1,4%) e interface de usuário complexa (1,4%).

Também pedimos aos participantes que indicassem os benefícios e dificuldades que perceberam ao utilizar as ferramentas. Quanto aos benefícios, eles destacaram que as

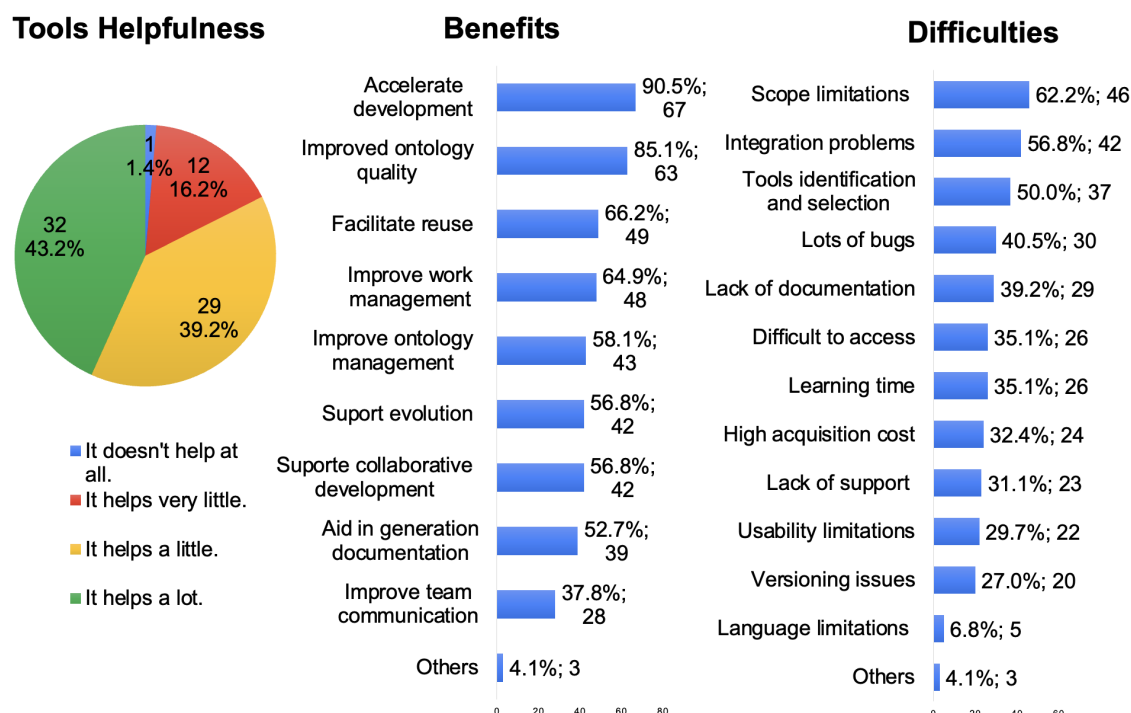


Figura 7 – Percepção do Uso de Ferramentas em EO.

ferramentas aceleram o processo de desenvolvimento de ontologias (90,5%) e melhoram a qualidade das ontologias produzidas (85,1%). Dentre as respostas, 66,2% reconhecem que facilitam a reutilização, 64,9% acham que oferecem melhoria da gestão do trabalho, dentre outras. Quanto às dificuldades, as mais proeminentes foram limitações de escopo (por exemplo, as ferramentas suportam um conjunto limitado de atividades de EO ou as funcionalidades fornecidas têm limitações) (62,2%) e problemas de integração (por exemplo, as ferramentas não permitem integração com outras, o resultado produzido em uma ferramenta não pode ser exportado para outra) (56,8%). Identificar e selecionar as ferramentas foi difícil para metade dos participantes. A Figura 7 resume as percepções dos participantes.

## Propriedades da ferramenta EO

Para investigar quais propriedades os engenheiros de ontologia esperam encontrar em uma ferramenta de suporte EO, apresentamos 10 propriedades (Adequação Funcional, Eficiência, Usabilidade, Confiabilidade, Segurança, Manutenibilidade, Portabilidade, Disponibilidade de uso, Colaboração e Cobertura) e pedimos aos participantes que indicassem o nível de relevância de cada uma delas. As propriedades são baseadas na ISO 25010 (ISO, 2011) (adaptamos algumas delas para melhor se adequarem às ferramentas EO). Adequação Funcional, Confiabilidade, Usabilidade e Disponibilidade de Uso foram consideradas pelos participantes como as propriedades mais relevantes. Em contraste, Manutenibilidade e Segurança foram consideradas menos relevantes (o número de respos-

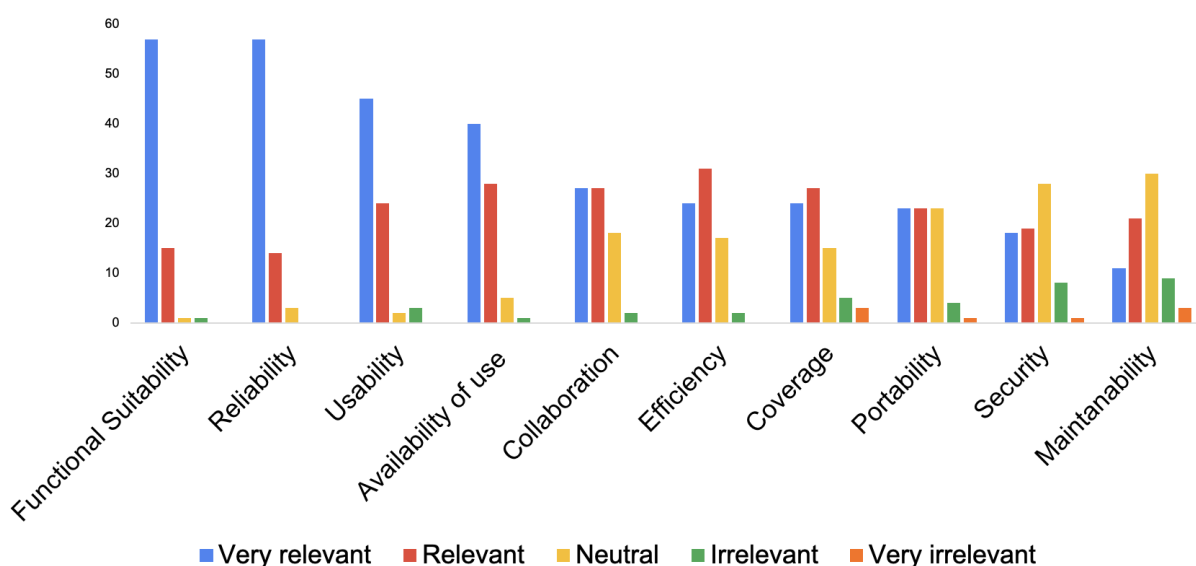


Figura 8 – Relevância das Propriedades das Ferramentas em EO.

tas “neutras” somadas à “irrelevantes” e “muito irrelevantes” é maior do que nas outras propriedades). Os participantes também foram solicitados a indicar outras propriedades que consideram importantes em uma ferramenta de suporte EO, pergunta respondida por 31 participantes. Agrupamos as propriedades equivalentes. As principais propriedades citadas foram Interoperabilidade, Documentação e Suporte, Controle de Versão e Reúso. A Figura 8 apresenta os resultados.

## 3.2 Discussão dos Resultados

A seguir é apresentada a discussão dos resultados encontrados, seguindo a mesma estrutura de tópicos da seção anterior.

Em relação ao **uso de ferramentas em OE**, cerca de 94,6% dos participantes declararam que usaram ferramentas sistematicamente, o que demonstra o papel fundamental que as ferramentas desempenham em OE. As ferramentas mais usadas foram Protégé (85,1%), Visual Paradigm (41,9%) e Astah (36,5%). Elas dão suporte à implementação de ontologias operacionais (Protégé) e à criação de modelos conceituais de ontologia (Visual Paradigm e Astah). Essas são as atividades mais específicas de EO e o suporte de ferramentas é crucial para realizá-las. Portanto, não é surpresa que as ferramentas mais usadas se relacionem a essas atividades. Ao analisar os tipos de ontologia, notamos que 79,7% dos participantes desenvolveram ontologias de referência e 67,6% ontologias operacionais. Embora tenha havido uma predominância de ontologias de referência, o número de participantes que usaram o Protégé (85,1%) é maior do que os que usaram o Visual Paradigm ou o Astah (59,5% dos participantes usaram apenas um deles ou ambos). Acreditamos que isso ocorre porque o Protégé é gratuito, muito popular entre engenheiros

de ontologia e apoiado por uma forte comunidade de usuários acadêmicos, governamentais e corporativos. Em contraste, embora o Astah e o Visual Paradigm sejam populares, eles têm recursos semelhantes, o que pode dividir os usuários (por exemplo, 23% dos participantes usaram o Visual Paradigm, enquanto 17,6% preferiram o Astah) e não são gratuitos (o Astah tem uma versão para estudantes e o Visual Paradigm tem uma versão para a comunidade com menos recursos do que a versão paga). O OntoUML-vp-plugin também foi citado por muitos participantes (36,5%). Ele ajuda a validar modelos conceituais criados no Visual Paradigm e representados no OntoUML e gera código que implementa a ontologia operacional correspondente. Assim, ele fornece suporte integrado às atividades de modelagem e implementação e faz a ponte entre o modelo conceitual da ontologia e sua versão operacional. Acreditamos que ferramentas como essa contribuem para a qualidade incorporada e devem ser mais exploradas pela comunidade EO. Quatro participantes (5,4%) declararam que raramente usaram ferramentas. Dado que desenvolver ontologias sem suporte de ferramentas pode ser inviável, levantamos a hipótese de que os participantes podem ter pensado que a questão se referia apenas a ferramentas específicas do EO. Essa foi a primeira questão sobre ferramentas apresentada aos participantes, portanto, eles não foram avisados de que ambas, ferramentas genéricas e específicas do EO, deveriam ser consideradas.

Ao analisar as 10 ferramentas mais citadas, notamos que seis delas são ferramentas específicas que dão suporte à modelagem conceitual e à implementação de ontologias, enquanto quatro são ferramentas genéricas. Ao analisar em quais fases do EO os diferentes tipos de ferramentas foram usados, notamos que quanto mais específica a tarefa é, mais específica a ferramenta precisa ser. Portanto, houve uma predominância de ferramentas genéricas para dar suporte às atividades de Especificação de Requisitos. Dado que os requisitos são geralmente registrados em formato de texto, ferramentas genéricas, como editores de texto e planilhas, podem ser úteis. A conceitualização e formalização de ontologias, design, implementação e avaliação, por sua vez, dependem principalmente de ferramentas específicas do EO. No entanto, o uso de ferramentas genéricas para dar suporte à conceitualização e formalização de ontologias (por exemplo, registrar definições de conceitos e axiomas) também foi expressivo. Embora ferramentas genéricas tenham sido úteis, pode ser difícil mapear informações registradas em tais ferramentas (por exemplo, editores de texto) para a ontologia armazenada em ferramentas específicas do (por exemplo, editores de ontologia). Portanto, avanços em ferramentas específicas de EO para dar suporte às fases iniciais do EO e conectá-las a outras ferramentas seriam bem-vindos.

Ao analisar o suporte de ferramentas para cada fase do EO desconsiderando os tipos de ferramentas, notamos que cerca de 92% dos participantes utilizaram ferramentas para dar suporte à Especificação de Requisitos, 86% para dar suporte à Conceitualização e Formalização, 84% para dar suporte ao Design, 76% para auxiliar na Implementação e 66% na Verificação, Validação e Teste. É possível notar valores mais altos nas três primeiras

fases quando comparadas às duas últimas fases, mesmo que as ferramentas sejam essenciais para implementar ontologias operacionais. Esses resultados provavelmente ocorrem porque muitos participantes (32,4%) desenvolveram apenas ontologias de referência e, portanto, não implementam ou testam ontologias operacionais.

Não houve uma fonte estruturada comum de ferramentas (por exemplo, um catálogo, um repositório). A maioria dos engenheiros de ontologia (mais de 80%) descobriu as ferramentas principalmente por meio de recomendações de outras pessoas. Isso sugere uma falta de conhecimento do panorama das ferramentas existentes, suas principais características, recursos, pontos fortes, fracos e assim por diante. Alguns engenheiros de ontologia (cerca de 17%) usaram ferramentas indicadas por abordagens sistemáticas de EO. Dado que cerca de 80% dos participantes declararam ter usado tais abordagens, esse resultado sugere que os métodos de EO não sugeriram ferramentas ou sugeriram ferramentas limitadas ou indisponíveis, o que está de acordo com as observações feitas em (Falbo, 2014).

Em relação às **características das ferramentas de EO**, as ferramentas livres têm sido predominantes. Esse resultado corrobora a literatura, que afirma que o livre acesso ao código-fonte e a economia de custos estão entre as principais características desejadas em ferramentas de software (Sanders, 1998; Feller, 2005; Fitzgerald, 2005). Os recursos oferecidos pelas ferramentas têm sido diversos. Os seis recursos mais citados estão diretamente relacionados às fases principais da EO: criação de diagramas, representação de hierarquia de conceitos, visualização de ontologias, implementação de ontologias, verificação de ontologias e validação. Isso demonstra que as ferramentas têm se preocupado em fornecer recursos para abordar as tarefas principais da EO. Outros recursos cobriram atividades de suporte, como reutilização, documentação automática e gerenciamento de configuração. Recursos relacionados a requisitos (definição e gerenciamento de questões de competência) têm sido os menos cobertos pelas ferramentas. Isso é consistente com os achados discutidos anteriormente, indicando uma falta de suporte de ferramentas específicas de EO para especificação de requisitos. Fornecer avanços para dar suporte às atividades de especificação de requisitos e conectar os artefatos produzidos aos resultados das fases subsequentes seria benéfico para a comunidade EO.

Apenas cerca de 10% dos participantes informaram que as ferramentas que eles usaram fornecem suporte integrado ao processo de EO. Isso sugere que, apesar da gama de ferramentas de EO existentes, elas ainda não deram suporte ao ciclo de vida da ontologia de forma abrangente e integrada. Na verdade, a maioria das ferramentas de EO se concentra em algumas atividades do ciclo de vida da ontologia, principalmente design e implementação (Corcho; Fernández-López; Gómez-Pérez, 2003). Além disso, as ferramentas geralmente funcionam de forma isolada ou com pouca integração com outras. Assim, os engenheiros de ontologia as executam individualmente, alternando entre diferentes

ferramentas e se esforçando para integrar resultados de diferentes ferramentas (Zhang *et al.*, 2004). Isso torna as atividades de EO incômodas e demoradas (Alobaid *et al.*, 2019), dificultando a eficiência do suporte à ferramenta. Mesmo algumas ferramentas de EO amplamente difundidas cobrem apenas algumas tarefas de EO e não podem ser usadas no sentido de um ambiente de desenvolvimento integrado (IDE - *Integrated development environment*) completo (Dziwis *et al.*, 2022; Corcho; Fernández-López; Gómez-Pérez, 2003). Portanto, há uma falta de conjuntos integrados de ferramentas que abordem todo o processo de EO. De forma análoga às ferramentas que fazem a ponte entre o modelo conceitual da ontologia e sua versão operacional, é importante produzir soluções que ofereçam ambientes de suporte EO integrados ou outras alternativas (por exemplo, *plugins*, *APIs*) para facilitar a integração das ferramentas existentes.

Quanto às **percepções do uso de ferramentas de EO**, cerca de 82% dos participantes percebem as ferramentas de EO como úteis. Embora esse resultado fosse de alguma forma esperado, quase metade desses participantes afirmou que as ferramentas ajudam apenas um pouco. Além disso, cerca de 18% dos participantes indicaram que as ferramentas ajudam muito pouco ou não são úteis. Os participantes declararam que os principais problemas que dificultam a utilidade das ferramentas estão relacionados à integração, reforçando a questão discutida acima, e à documentação. Sem uma integração adequada entre as várias ferramentas usadas para dar suporte às atividades de EO e documentação e suporte claros para usar as ferramentas, as vantagens de usá-las são prejudicadas (Alobaid *et al.*, 2019; Vigo *et al.*, 2014). Por outro lado, o suporte das ferramentas no desenvolvimento de ontologias complexas, na visualização e avaliação de ontologias, bem como na automação de tarefas repetitivas contribuíram para que os engenheiros de ontologia percebessem as ferramentas como úteis.

As razões pelas quais os engenheiros de ontologia percebem as ferramentas de EO como úteis (ou não) foram declaradas pelos participantes em uma pergunta aberta. Depois disso, eles foram questionados sobre os benefícios e dificuldades do uso de ferramentas de EO. Em geral, os resultados foram consistentes com as razões apresentadas por eles, fornecendo uma visão geral dos prós e contras do uso de ferramentas de EO. Acelerando o processo de desenvolvimento de ontologia e melhorar a qualidade das ontologias produzidas foram os benefícios percebidos por mais participantes. Esses benefícios também foram defendidos na literatura — por exemplo, (Mendonça *et al.*, 2021; Corcho; Fernández-López; Gómez-Pérez, 2003; Kotis; Vouros; Spiliotopoulos, 2020). Portanto, os resultados reforçam e fornecem evidências da utilidade das ferramentas. Embora os engenheiros de ontologia tenham várias ferramentas de EO disponíveis, eles ainda enfrentam dificuldades ao usá-las, principalmente devido a limitações de escopo e problemas de integração. Além disso, selecionar as ferramentas tem sido desafiador e as ferramentas apresentaram muitos bugs. Esses problemas também foram apontados na literatura — por exemplo (Vigo *et al.*, 2014; Katifori *et al.*, 2007; Dzbor *et al.*, 2006; Corcho; Fernández-López; Gómez-Pérez,

2003; Dziwis *et al.*, 2022; Zouaq; Nkambou, 2010; Alobaid *et al.*, 2019; Zhang *et al.*, 2004) — revelando que, apesar dos avanços nas ferramentas de EO, há necessidade de mais melhorias. Em relação aos problemas de integração, que é uma questão proeminente revelada nos resultados da pesquisa, é importante avançar nos trabalhos que visam criar uma bancada de trabalho comum para EO que facilite o desenvolvimento de ontologias durante todo o ciclo de vida da ontologia, incluindo também suporte ao gerenciamento de configuração e evolução de ontologias (Corcho; Fernández-López; Gómez-Pérez, 2003).

Em relação às **propriedades das ferramentas de EO**, as quatro consideradas mais relevantes pelos participantes (Adequação Funcional, Confiabilidade, Usabilidade e Disponibilidade) mostram que os engenheiros de ontologia esperam ferramentas com a capacidade de fazer o que foram criadas para fazer, ou seja, ter as funções necessárias para atender às tarefas e objetivos alvo (Puspaningrum; Rochimah; Akbar, 2017), produzindo resultados corretos e confiáveis, comunicando-se claramente com os usuários e sendo facilmente acessadas (por exemplo, ferramentas gratuitas). Em outras palavras, as ferramentas devem fazer corretamente o que se pretende fazer e deve ser fácil acessá-las e usá-las. Essas propriedades são percebidas principalmente da perspectiva externa (ou seja, quando os engenheiros de ontologia são os usuários das ferramentas) e são consistentes com outros resultados. Por exemplo, bugs, acesso limitado e problemas de usabilidade foram apontados como dificuldades enfrentadas pelos participantes ao usar as ferramentas, indicando a necessidade de melhorias em confiabilidade, disponibilidade e usabilidade e a relevância de tais propriedades. Apesar das limitações de escopo terem sido apontadas pela maioria dos participantes como um obstáculo a ser superado ao usar as ferramentas, a Cobertura não estava entre as cinco principais propriedades. Segurança e Manutenibilidade foram as propriedades consideradas menos relevantes. Segurança se refere à capacidade da ferramenta de proteger dados e informações. Acreditamos que os participantes perceberam essa propriedade como menos relevante porque, em geral, ao desenvolver ontologias, não registramos informações confidenciais ou sensíveis nas ferramentas. No entanto, proteger os modelos, conceitos, instâncias, etc. da ontologia ainda é necessário, o que foi reconhecido por 50% dos participantes, que consideraram essa propriedade relevante ou muito relevante. Manutenibilidade, por sua vez, diz respeito à perspectiva interna, ou seja, de quem desenvolve a ferramenta. Assim, sua relevância seria percebida apenas pelos engenheiros de ontologia que interagem com as ferramentas desempenhando papéis diferentes, usuário (engenheiro de ontologia) e desenvolvedor (engenheiro de software). Os resultados sugerem que esse não é o caso da maioria dos participantes.

A lista de propriedades que fornecemos aos participantes não foi exaustiva. Para capturar outras propriedades relevantes, pedimos que eles citassem outras que considerassem relevantes. Interoperabilidade, Documentação e Suporte, Controle de Versão e (Suporte para) Reutilização se destacam. Todas essas propriedades estão relacionadas a aspectos apontados pelos participantes em outras questões, o que reforça sua relevância. Problemas

de integração, falta de documentação e suporte e problemas de controle de versão foram indicados como dificuldades enfrentadas ao usar as ferramentas. A reutilização, por sua vez, ajuda a acelerar o processo de desenvolvimento, que é o principal benefício do uso de ferramentas percebido pelos participantes. Para atender adequadamente às expectativas e necessidades dos engenheiros de ontologia, as ferramentas devem manifestar propriedades referentes a atributos de qualidade desejáveis. Os resultados da pesquisa indicam que algumas delas não foram adequadamente abordadas pelas ferramentas existentes. Portanto, há uma oportunidade de explorar como essas propriedades devem ser manifestadas e desenvolver estratégias para alcançá-las.

### 3.3 Considerações finais do capítulo

Este capítulo apresentou uma investigação sobre o uso de ferramentas no suporte à Engenharia de Ontologias (EO), buscando responder a três questões principais: (RQ1) Quais ferramentas são utilizadas e quais suas características? (RQ2) Quais benefícios e dificuldades são percebidos no uso dessas ferramentas? (RQ3) Quais propriedades são consideradas importantes para ferramentas de EO?

Os resultados mostraram que em relação a RQ1 as ferramentas específicas, como Protégé, Visual Paradigm e Astah, são predominantes no apoio à modelagem e implementação, enquanto ferramentas genéricas são mais comuns nas fases iniciais, especialmente na Especificação de Requisitos. A escolha das ferramentas se deu majoritariamente por recomendações de outros profissionais, havendo preferência por opções gratuitas. Contudo, identificou-se a ausência de um suporte integrado adequado que cubra todo o processo de EO. Relacionado à RQ2 entre os benefícios percebidos, destacam-se o aumento do desempenho no desenvolvimento e a melhoria da qualidade das ontologias, enquanto os principais desafios foram problemas de integração e limitações de escopo. E, quanto a RQ3 as propriedades mais valorizadas foram Adequação Funcional, Confiabilidade, Usabilidade e Disponibilidade.

Esses achados fornecem um panorama inicial e evidenciam lacunas que podem ser exploradas em estudos futuros. Em especial, aponta-se a necessidade de criar soluções que promovam integração efetiva entre diferentes ferramentas, conectando artefatos e etapas do processo de forma fluida, algo que beneficiaria iniciantes, que muitas vezes enfrentam dificuldades para saber quais ferramentas utilizar em cada fase do processo de desenvolvimento de ontologias. Oferecer recursos que guiem o usuário desde a concepção até a implementação e melhorar o suporte para a especificação de requisitos, pode aumentar o desempenho no processo e qualidade dos resultados.

Nesse sentido, no Capítulo 4 propomos a ampliação desta pesquisa com o objetivo de integrar as ferramentas identificadas a um método estruturado — a saber, o método

SABiOx (Aguiar; Souza, 2024) —, fornecendo uma proposta de uso das ferramentas mais citadas na pesquisa nas diferentes fases do método, percorrendo todo o processo de desenvolvimento da ontologia. Para exemplificar, será realizada a aplicação de uma ontologia, a Object-Oriented Code Ontology (OOC-O) (Aguiar; Falbo; Souza, 2019), em cada etapa do método SABiOx (Especificação de Requisitos, Modelagem e Implementação). Essa exemplificação permitirá demonstrar, na prática, como as ferramentas já conhecidas podem ser utilizadas em cada fase, mostrando que, mesmo sem uma total interoperabilidade entre elas, é possível integrá-las ao método e apoiar todo o processo de Engenharia de Ontologias.

## 4 Aplicação de ferramentas no método SABiOx

Neste capítulo são apresentadas as ferramentas que podem ser usadas nas fases e atividades do método SABiOx, mostrando o que é realizado em cada uma, e quais as demandas cada ferramenta deve atender em cada fase, auxiliando assim no processo de desenvolvimento de ontologias. Para exemplificar, será realizada a aplicação de uma ontologia, a Object-Oriented Code Ontology (OOC-O) (Aguiar; Falbo; Souza, 2019), em cada etapa do método SABiOx (Especificação de Requisitos, Modelagem e Implementação). Essa exemplificação permitirá demonstrar, na prática, como as ferramentas já conhecidas podem ser utilizadas em cada fase, mostrando que, mesmo sem uma total interoperabilidade entre elas, é possível integrá-las ao método e apoiar todo o processo de Engenharia de Ontologias.

Diante da pesquisa realizada e com os resultados apresentados pelo *survey*, o objetivo deste detalhamento envolvendo ferramentas são:

1. Servir de base para iniciantes na construção de ontologias, oferecendo um caminho do início ao fim do ciclo de desenvolvimento;
2. Mostrar quais recursos são atendidos por cada ferramenta, como um meio de auxiliar desenvolvedores de ontologia iniciantes a saber o que pode procurar e o que vai achar em cada ferramenta em cada fase;
3. Mostrar para os usuários que o método embasado por ferramentas que atendam os seus propósitos pode ser um diferencial no desenvolvimento de ontologias e que isso agiliza o processo e pode trazer mais qualidade para as ontologias construídas.

Selecionar as ferramentas mais adequadas que fornecem os recursos necessários é uma decisão importante no processo de Engenharia de Ontologias e pode não ser fácil (Abburu; Babu, 2013). Na literatura é apontado que há uma grande dificuldade em saber qual a ferramenta usar na construção de ontologias, principalmente por iniciantes na área, fato que foi confirmado na aplicação do questionário. No *survey* foram listadas diversas ferramentas encontradas na literatura que podem ser usadas na construção de ontologias independente de sua fase, sendo solicitado aos participantes que apontassem quais ferramentas utilizam no desenvolvimento de ontologias, para o qual obtivemos 74 respostas. A Tabela 3 apresenta o resultado da pesquisa com o nome da ferramenta e número de participantes que indicou seu uso.

Tabela 3 – Ferramentas utilizadas para construção de ontologias e número de usuários.

<b>Ferramentas</b>	<b>Número de Participantes</b>
Astah	27
Crowd	3
Chowlk	2
Draw.io	5
Menthor	8
Note-taking software (e.g., Microsoft OneNote, Google Keep)	6
OLED	12
OntoEdit	3
OntoUML-vp-plugin	20
OOPS!	4
OWLGrEd	3
Presentation software (e.g., Microsoft Power Point, Google Slides)	15
Protegé	63
Robot	2
Spreadsheet software (e.g., Microsoft Excel, Google Sheets)	20
Visual Paradigm	31
Visual Studio Code with specific plugins (for writing Turtle)	2
WebODE	2
Word processing software (e.g., Microsoft Word, Google Docs)	18
Outras <sup>1</sup>	42

Para atender o objetivo deste estudo, as seções a seguir abordam cada uma das fases do método SABiOx, com uma breve descrição sobre as atividades que são englobadas e quais características/funcionalidades uma ferramenta deve ter para auxiliar naquela determinada fase. A cada etapa (que pode englobar mais de uma fase) a ferramenta indicada foi baseada nas que já são utilizadas pela comunidade. Seguindo o critério de respostas levantadas no *survey*, foi selecionada a ferramenta mais votada que atende as necessidades da fase.

<sup>1</sup> Ferramentas citadas por eles além das listadas, dentre as 42 respostas as ferramentas citadas foram: Prolog, Ontoprolog, PlantUML, Bloco de notas, Coding tool, Hermit, Tools for Mind Maps, Coogole.it, CmapTools, Ontofox, OOPS!, FOOPS, Diagram.io, Miro, ICON CROWD, Cameo Concept Modeler, Enterprise Architect, Apache Jena, FluentEditor, Ptyhon libraries for ontology development, yEd Graph Editor, GLUON (Thales internal tool), OnToology, Widoco, Dzeta.

## 4.1 Fase de Requisitos

No método SABiOx a 1ª fase é a de Requisitos, na qual o Engenheiro de Ontologias irá definir e descrever o propósito da ontologia, identificar o domínio a ser representado e dimensionar seus limites, elicitare os requisitos funcionais e não funcionais para a ontologia, identificar os subdomínios relacionados às questões de competência definidas para o domínio, documentar todas essas especificações da ontologia criando um Documento de Especificação da Ontologia, que deve ter um gerenciamento de configuração a fim de controlar as alterações, versões e entregas das informações registradas neste documento e também avaliar se todas informações extraídas e registradas atendem às expectativas do *ontology owner*.

Nesta fase é necessária uma ferramenta de anotação que permita registrar dados, alterar, corrigir e controlar as versões ao longo do desenvolvimento. Dentre as ferramentas indicadas pelos respondentes do survey, as mais utilizadas são ferramentas de propósito geral (por exemplo: *Microsoft Excel*, *Google Sheets*, *Microsoft Word*, *Google Docs*). Embora essas ferramentas atendam a necessidade de anotação, edição e alteração, serem amplamente conhecidas e acessíveis, elas exigem uma dedicação e atenção maior para manter a consistência e padronização do conteúdo ao longo do tempo.

Nesse contexto, recomenda-se o uso de editores baseados em  $\text{\LaTeX}$  que oferecem uma escrita estruturada e com suporte nativo à padronização, o que permite a criação de documentos com mais consistência e qualidade. Dentre editores disponíveis, destacam-se opções como Overleaf, TeXstudio, TeXmaker e VS Code com extensões  $\text{\LaTeX}$ . Diante disso, os editores  $\text{\LaTeX}$  se mostram particularmente adequados para a fase de Requisitos do método SABiOx ao oferecerem precisão, organização e padronização ao processo de documentação.

Além das vantagens já citadas, o uso de editores  $\text{\LaTeX}$  pode ser ainda mais eficiente quando aliado a sistemas de controle de versão como o Git. O Git permite registrar todo o histórico de alterações, recuperar versões anteriores, acompanhar a evolução dos documentos e manter um controle rigoroso sobre as modificações realizadas, algo útil quando há múltiplos colaboradores ou revisões frequentes. A integração com o Git amplia as possibilidades de versionamento, backup externo e integração com fluxos de desenvolvimento mais complexos. Essa combinação fortalece o gerenciamento da documentação, oferecendo maior segurança, rastreabilidade e controle ao longo do processo. Assim, a utilização conjunta de editores  $\text{\LaTeX}$  e sistemas de controle de versão como o Git se mostra altamente recomendada nessa primeira fase do método SABiOx, contribuindo para a confiabilidade, qualidade e organização da documentação da ontologia.

Outro ponto importante é a utilização de um *template* com as seções e elementos essenciais para compor o Documento de Especificação da Ontologia, que defina previamente

a estrutura, formatação e padrões a serem seguidos, reduzindo tempo gasto com ajustes visuais e facilitando a adaptação do conteúdo para diferentes finalidades. A proposta é que, a cada atualização do documento, seja salva uma nova versão seguindo o *template* e que as alterações sejam registradas por meio de *commits* no Git. Esse procedimento permite um controle de versão completo, o rastreamento preciso das modificações, o compartilhamento do documento com todos os envolvidos no projeto e a o histórico de desenvolvimento da ontologia. A combinação de editores  $\text{\LaTeX}$ , controle de versão com Git e uso de um *template* estruturado, fortalece o gerenciamento da documentação, oferecendo maior segurança, rastreabilidade, padronização e controle ao longo do processo. Assim, essa prática se mostra altamente recomendada nessa primeira fase do método SABiOx, contribuindo para a confiabilidade, qualidade e organização da documentação da ontologia.

A Figura 9 apresenta a elaboração do documento no editor *Overleaf*, utilizando como exemplo a ontologia OOC-O (Aguiar; Falbo; Souza, 2019) refeita com o *template* para o Documento de Especificação de Requisitos disponibilizado gratuitamente em <https://github.com/vitorsouza/latex-templates>. Esse exemplo demonstra, de forma prática, a aplicação do modelo e o registro estruturado das informações e requisitos definidos na etapa de Requisitos. O código-fonte da implementação da OOC-O realizada no contexto deste trabalho encontra-se no repositório [https://github.com/ClareteADGomes/Projeto\\_OOC-O\\_Final](https://github.com/ClareteADGomes/Projeto_OOC-O_Final) e também arquivada na página do SABiOx no site do NEMO, em <https://nemo.inf.ufes.br/projetos/sabiox/>.

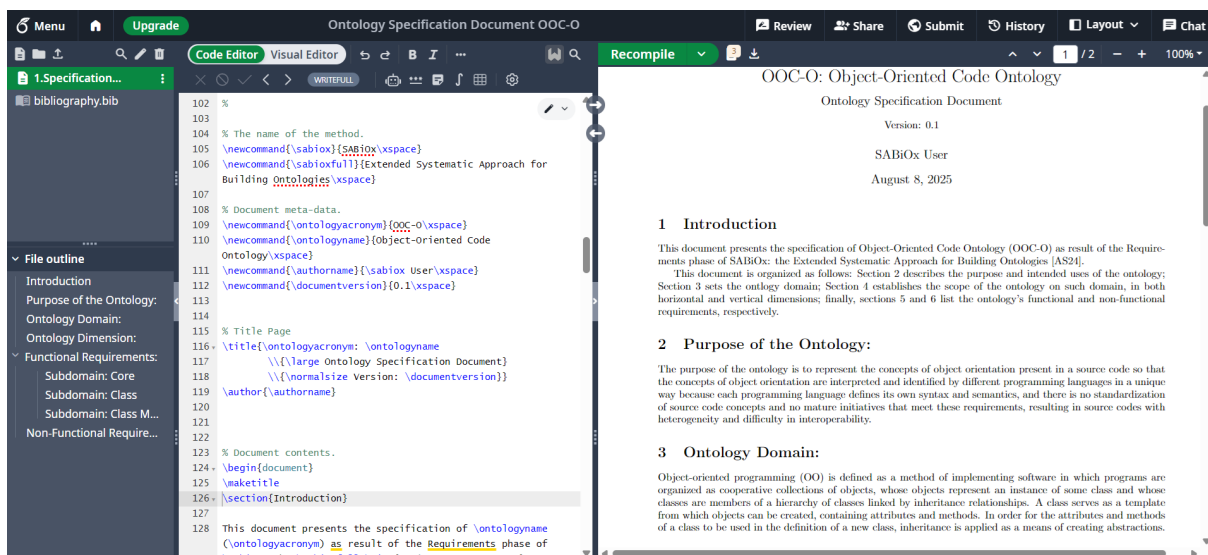


Figura 9 – Documento de Especificação de Requisitos da OOC-O elaborado num editor  $\text{\LaTeX}$ .

## 4.2 Fase de Configuração

Na Fase de Configuração o objetivo é definir questões que irão guiar a elaboração da ontologia de referência, que ocorre na próxima etapa. Desta forma, aqui são definidas questões que implicam na modelagem conceitual. Nesta fase serão definidas a linguagem de representação a ser utilizada na ontologia de referência, a ontologia de fundamentação a ser adotada, os critérios de identificação de aderência de conceitos que serão adotados, e as ontologias de referência que serão reusadas. Além disso, esta fase prevê a documentação das premissas da ontologia de referência, o controle das alterações, versões e entregas das informações registradas neste documento e a avaliação se todas informações registradas atendem as expectativas do *ontology owner* e do especialista no domínio.

Nessa fase continua sendo necessária uma ferramenta de anotação que permita registrar dados, alterar, corrigir e controlar as versões ao longo do desenvolvimento, assim como a usada na fase anterior. Ferramentas como editores baseados em  $\text{\LaTeX}$  em conjunto com o Git, continuam sendo uma alternativa viável, pois possibilitam o registro colaborativo das decisões, o versionamento dos documentos e o compartilhamento transparente entre os envolvidos. O uso de um *template* estruturado, como o disponibilizado na fase de Requisitos, pode ser estendido para essa etapa, agora com campos voltados às decisões conceituais e técnicas da configuração.

O exemplo usando a ontologia OOC-O refeita com o *template* para o Documento de Referência da Ontologia no editor  $\text{\LaTeX}$  Overleaf é mostrado na Figura 10. O modelo é também disponibilizado gratuitamente no mesmo endereço eletrônico mencionado anteriormente. Esse exemplo demonstra, de forma prática, a aplicação do modelo e das decisões conceituais e técnicas definidas na etapa de Configuração.

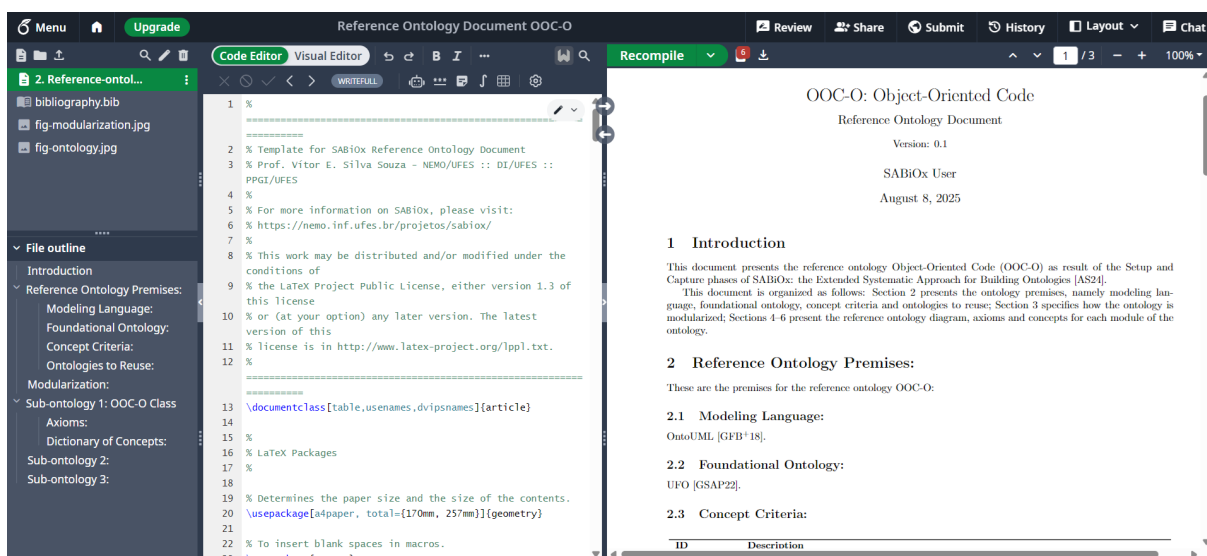


Figura 10 – Documento de Referência da Ontologia OOC-O elaborado no editor  $\text{\LaTeX}$ .

### 4.3 Fase de Captura

A Fase de Captura, é onde a conceituação do domínio será representada de forma bem fundamentada tendo como base as especificações elaboradas na Fase de Requisitos e questões definidas na Fase de Configuração. Nesta fase primeiramente o Engenheiro de Ontologias identifica quais conceitos do domínio fazem parte do propósito da ontologia e então cataloga esses conceitos, de acordo com os critérios já definidos. Em seguida extrai a visão das ontologias de referência que serão reusadas na ontologia em construção, depois identifica os axiomas de restrição e inferência que devem ser considerados no modelo conceitual e então modela os conceitos e relações abrangidos pelo propósito da ontologia em uma linguagem de representação de ontologia independente de tecnologia e, a partir daí, integra as visões da ontologia em construção e as das ontologias reusadas. Agora pode identificar os módulos em que se pode decompor a ontologia e aqueles que podem ser considerados separadamente enquanto estão interligados com outros módulos. Nesta fase deve-se também documentar o modelo conceitual da ontologia e para isso será utilizado o mesmo documento usado na fase anterior, somente complementando as informações, e então controlar as alterações, versões e entregas das novas informações registradas no documento, como também avaliar se todas informações registradas atendem às expectativas da ontologia a ser construída. Por fim, publicar a Ontologia de Referência.

Nessa fase, é necessário o uso de ferramentas que permitam representar graficamente os conceitos e suas relações, também documentar decisões de modelagem, integrar visões de ontologias reusadas com a ontologia em construção e manter controle sobre as alterações realizadas ao longo do processo. Além disso, é necessário que a ferramenta possua suporte à linguagem de modelagem escolhida, em todo a sua expressividade. Para isso, serão utilizadas duas abordagens complementares: a utilização do Documento de Referência da Ontologia, que registra sistematicamente os conceitos identificados do domínio com seus critérios de inclusão e possíveis reusos, e a modelagem conceitual no Visual Paradigm com o plugin OntoUML, que possibilita uma representação visual e formal da estrutura da ontologia.

O uso do Documento de Referência da Ontologia, gerenciado com sistemas de controle de versão como o Git, tem como objetivo organizar e registrar de forma estruturada todos os conceitos do domínio, suas definições, origem (se própria ou reusada), critérios de aceitação e observações relevantes. Nessa fase as anotações são feitas no mesmo template da fase anterior, preenchendo outros campos. Embora não seja possível ainda enviar as anotações do documento de forma automatizada para a ferramenta de modelagem, esse recurso funciona como base auxiliar para a modelagem visual e também como instrumento de rastreabilidade conceitual.

Para modelagem conceitual, a ferramenta que recebeu o maior número de indicações foi o Visual Paradigm usado junto com o *plugin* OntoUML. Essa ferramenta permite a

representação estruturada e visual da ontologia, além de possibilitar a criação de módulos e a visualização das relações entre conceitos com estereótipos específicos. O *plugin* OntoUML adiciona ao Visual Paradigm recursos projetados para modeladores OntoUML. Esses recursos variam desde a ativação de estereótipos OntoUML em diagramas de classes até a verificação e transformação de modelos.

Além disso, o Visual Paradigm permite a organização modular da ontologia por meio de pacotes (*packages*), favorecendo a decomposição em visões distintas e a modelagem por partes. Cada elemento modelado pode conter documentação embutida, como descrições, notas e referências, o que facilita a rastreabilidade e a justificativa das decisões de modelagem. A ferramenta também permite exportações em formatos variados, como imagem e arquivo XML em todas versões e projeto (extensão .vpp, XMI e Excel) somente nas versões pagas e com funções restritas também. Quando usado com o *plugin* OntoUML tem a possibilidade de exportar no formato JSON e para a linguagem OWL com o suporte de gUFO (gentle UFO), que é uma implementação leve da ontologia de fundamentação UFO em OWL (Almeida *et al.*, 2019). Essa última permite a tradução direta dos estereótipos OntoUML gerando um arquivo em formato Turtle (.ttl) que pode ser importado na ferramenta de implementação de ontologia Protégé. A exportação para gUFO viabiliza a continuidade do processo de desenvolvimento ontológico, permitindo que o modelo construído seja importado e estendido, validado ou instanciado no Protégé, que será utilizado na próxima fase. Isso estabelece uma ponte entre a modelagem conceitual e a formalização ontológica, promovendo interoperabilidade entre ferramentas e fortalecendo a consistência da ontologia construída ao longo do método SABiOx.

O Visual Paradigm pode ser adquirido e baixado diretamente do site oficial em <<https://www.visual-paradigm.com/>>. O *plugin* OntoUML, que adiciona suporte para a modelagem ontológica no Visual Paradigm, está disponível no repositório oficial do projeto OntoUML em <<https://github.com/OntoUML/ontouml-vp-plugin>>, onde os usuários podem encontrar instruções para instalação e uso.

Como exemplo prático, a Figura 11 apresenta a modularização da ontologia OOC-O integrada ao Documento de Referência da Ontologia, enquanto a Figura 12 ilustra a subontologia OOC-O Class modelada na ferramenta Visual Paradigm. Essa integração demonstra como as informações documentadas e a modelagem visual se complementam nesse processo.

A Figura 13 apresenta o código gerado automaticamente em formato Turtle (.ttl) pela ferramenta e pelo *plugin*, ilustrando essa integração e a geração do modelo formal a partir da modelagem conceitual. No trecho exibido, é possível observar a ontologia OOC-O, utilizada como exemplo prático, assim como nas etapas anteriores e também pode ser notado que os conceitos presentes são originados da ontologia fundamental gUFO.

Apesar de seus benefícios, o uso do Visual Paradigm também apresenta alguns

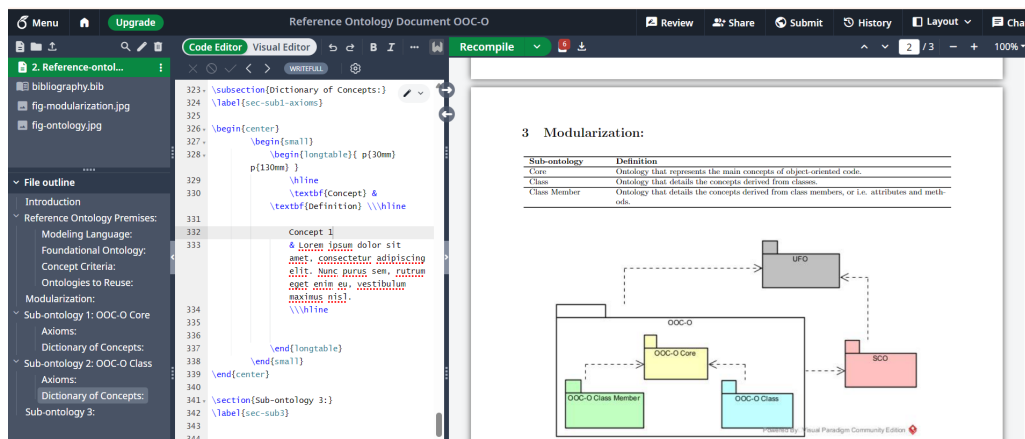


Figura 11 – Modularização feita no Visual Paradigm e integrada ao Documento de Referência da Ontologia no editor  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ .

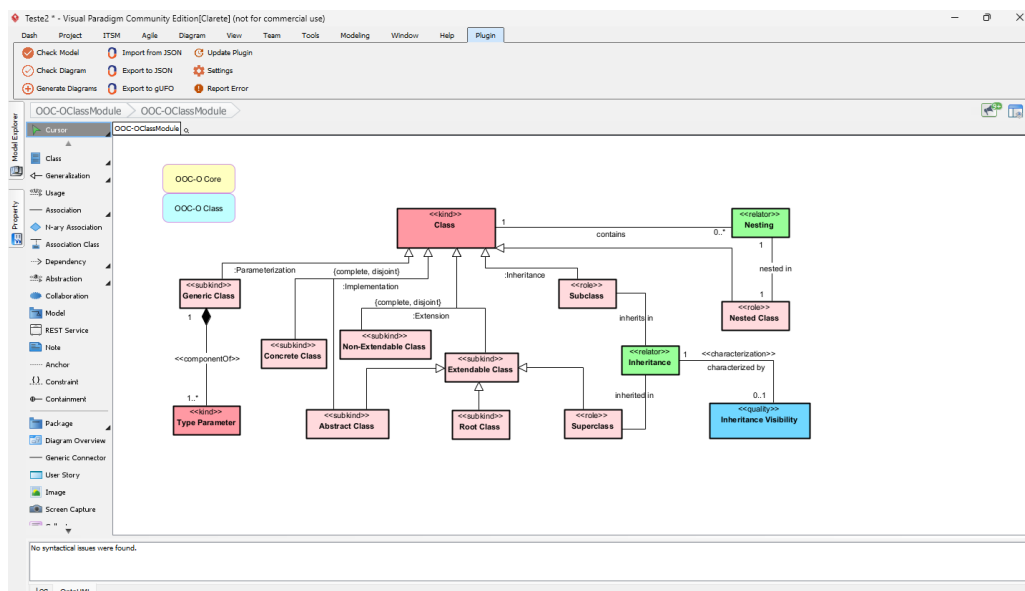


Figura 12 – Diagrama OntoUML do Módulo Class da OOC-O no Visual Paradigm.

desafios. A ferramenta, mesmo em sua versão *Community*, possui certa complexidade de uso, o que pode exigir uma curva de aprendizado inicial, especialmente para usuários não familiarizados com modelagem conceitual. Outro ponto a considerar é que a modelagem gráfica, embora ofereça um modelo bem estruturado e visualmente amigável, pelo fato de não ter uma forma automatizada de importação dos dados/informações da planilha de conceitos, precisa ser cuidadosamente sincronizada com os registros textuais e tabulares do projeto realizado, para evitar inconsistências. O uso de múltiplas ferramentas editor  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , Visual Paradigm e Git, demanda uma boa organização da equipe e disciplina nos registros e versões.

Ainda assim, quando bem integradas, essas ferramentas oferecem um ambiente adequado para a captura, modelagem e documentação da ontologia de referência, atendendo às exigências da terceira fase do método SABiOx.

```

@prefix : <http://example.com#>.
@prefix gufo: <http://purl.org/nemo/gufo#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.

<http://example.com> rdf:type owl:Ontology;
    owl:imports gufo: .
:Class rdf:type owl:Class, gufo:Kind, owl:NamedIndividual;
    rdfs:subClassOf gufo:FunctionalComplex;
    rdfs:label "Class"@en.
:ConcreteClass rdf:type owl:Class, gufo:SubKind, owl:NamedIndividual;
    rdfs:label "Concrete Class"@en.
:AbstractClass rdf:type owl:Class, gufo:SubKind, owl:NamedIndividual;
    rdfs:label "Abstract Class"@en.
:ExtendableClass rdf:type owl:Class, gufo:SubKind, owl:NamedIndividual;
    rdfs:label "Extendable Class"@en.
:NonExtendableClass rdf:type owl:Class, gufo:SubKind, owl:NamedIndividual;
    rdfs:label "Non-Extendable Class"@en.
:Superclass rdf:type owl:Class, gufo:Role, owl:NamedIndividual;
    rdfs:label "Superclass"@en.
:Subclass rdf:type owl:Class, gufo:Role, owl:NamedIndividual;
    rdfs:label "Subclass"@en.
:RootClass rdf:type owl:Class, gufo:SubKind, owl:NamedIndividual;
    rdfs:label "Root Class"@en.
:NestedClass rdf:type owl:Class, gufo:Role, owl:NamedIndividual;
    rdfs:label "Nested Class"@en.
:GenericClass rdf:type owl:Class, gufo:SubKind, owl:NamedIndividual;
    rdfs:label "Generic Class"@en.
:TypeParameter rdf:type owl:Class, gufo:Kind, owl:NamedIndividual;
    rdfs:subClassOf gufo:FunctionalComplex;
    rdfs:label "Type Parameter"@en.
:Nesting rdf:type owl:Class, gufo:Kind, owl:NamedIndividual;
    rdfs:subClassOf gufo:Relator;
    rdfs:label "Nesting"@en.
:InheritanceVisibility rdf:type owl:Class, gufo:Kind, owl:NamedIndividual;
    rdfs:subClassOf gufo:Quality;
    rdfs:label "Inheritance Visibility"@en.
:Inheritance rdf:type owl:Class, gufo:Kind, owl:NamedIndividual;
    rdfs:subClassOf gufo:Relator;

```

Figura 13 – Código no formato Turtle gerado pelo plugin OntoUML no Visual Paradigm.

## 4.4 Fase de Design

Na Fase de Design serão definidas questões tecnológicas para guiar a implementação da ontologia operacional, logo, incidem sobre o ambiente tecnológico. Primeiramente define a linguagem de representação na construção da ontologia operacional (codificação) e identifica os vocabulários a serem adotados, tanto na construção da ontologia como nas ontologias reusadas. Logo após, define a codificação a ser aplicada na ontologia operacional, indicando como a arquitetura da ontologia operacional será derivada da ontologia de referência. As premissas da ontologia operacional devem então ser documentadas utilizando o Documento de Ontologia Operacional colocando as questões definidas nessa fase, e então controlar as alterações, versões e entregas das informações registradas no documento, como também avaliar se todas as informações registradas atendem às expectativas da ontologia a ser construída, e se requisitos e qualidade estão de acordo com definições feitas na Fase de Concepção.

Assim como nas fases anteriores, nesta etapa é fundamental dispor de uma ferramenta de anotação que permita registrar informações, realizar ajustes, correções e manter o controle das versões ao longo do desenvolvimento. O uso de um editor  $\text{\LaTeX}$  continua sendo recomendado, pois oferece suporte à padronização, organização e qualidade na documentação. A combinação desse modelo com o controle de versão por meio do Git permite a construção colaborativa dos registros, o versionamento dos documentos e o compartilhamento entre todos os envolvidos, garantindo maior confiabilidade e organização

na documentação da ontologia operacional.

Como exemplo prático, a ontologia OOC-O será utilizada para ilustrar a aplicação do *template* estruturado em  $\text{\LaTeX}$  denominado Documento da Ontologia Operacional, que também faz parte do conjunto de *templates* já apresentado anteriormente e já está pronto e adaptado para essa fase do método SABiOx. Esse modelo é apresentado na Figura 14.

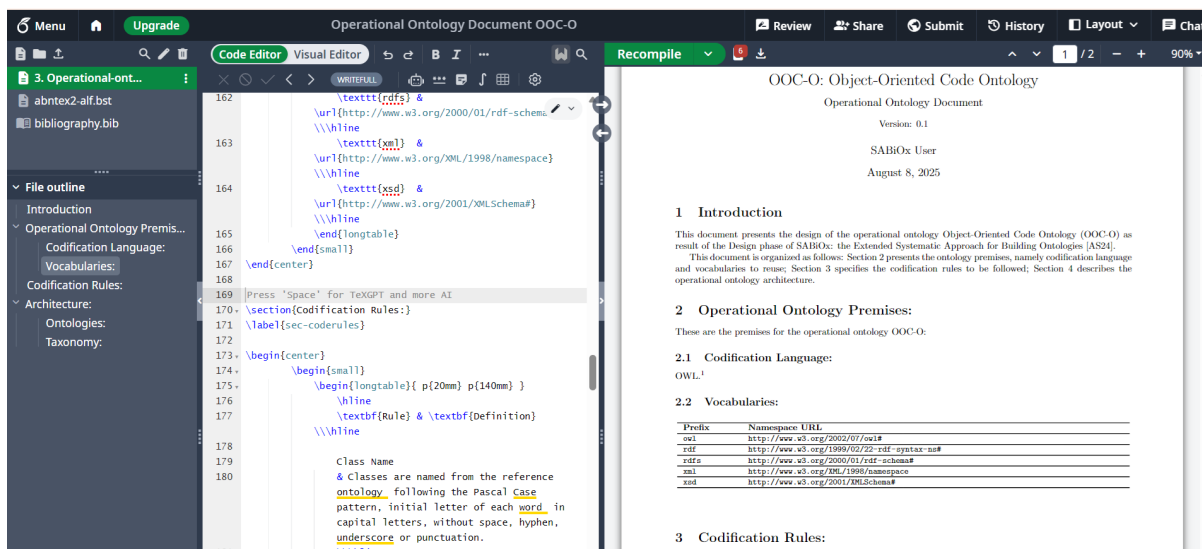


Figura 14 – Documento da Ontologia Operacional OOC-O elaborado no editor  $\text{\LaTeX}$ .

## 4.5 Fase de Implementação

Na Fase de Implementação a ontologia de referência será implementada em uma ontologia operacional, ou seja, o Engenheiro de Ontologias deverá transformar a ontologia de referência em uma ontologia interpretável por máquina. Inicia com a codificação dos conceitos da ontologia de referência como elementos da linguagem formal de ontologia operacional e a codificação das relações da ontologia de referência como propriedades e restrições em uma linguagem formal de ontologia operacional, seguindo com a codificação dos axiomas de acordo com a linguagem operacional. Então a ontologia operacional deve ser documentada na linguagem de codificação definida, enquanto são controladas as alterações, versões e entregas das informações registradas na ontologia operacional, bem como se avalia se todas informações registradas atendem às expectativas da ontologia a ser construída e se requisitos e qualidade estão de acordo com definições feitas na Fase de Requisitos. Por fim, deve ser feita a publicação da Ontologia Operacional.

Nesta fase, a ferramenta utilizada precisa oferecer suporte à linguagem formal adotada (como OWL), permitir a codificação de conceitos, propriedades e axiomas, além de contar com recursos de validação semântica e documentação estruturada. Também é desejável que ela possibilite a importação de modelos existentes, gerados em etapas anteriores, e que se integre com ferramentas de controle de versão para manter a rastreabilidade

do desenvolvimento da ontologia.

Para apoiar nessa fase, é indicado o uso da ferramenta Protégé, amplamente reconhecida para a construção e edição de ontologias formais e também a indicada no *survey* como a mais usada pelos respondentes. Nesta fase, o modelo conceitual elaborado no Visual Paradigm, que foi exportado em formato turtle (.ttl), é então, importado no Protégé, possibilitando a continuidade do desenvolvimento da ontologia dentro de um ambiente que oferece recursos específicos para a definição de propriedades, restrições, axiomas e metadados em conformidade com a linguagem formal adotada. Para importar o arquivo gerado pelo Visual Paradigm em formato turtle (.ttl) no Protégé, é necessário utilizar um *plugin* compatível. Um exemplo está disponível em <<https://github.com/nemo-ufes/ufo-protege-plugin>>.

Como parte da exemplificação prática, é apresentada a Figura 15, que mostra a tela de importação, no Protégé, do código da ontologia OOC-O gerado em gUFO pelo Visual Paradigm e importado no Protégé. Essa figura demonstra, de forma visual, o processo que conecta o modelo conceitual criado no Visual Paradigm à sua implementação formal no Protégé.

O Protégé permite a edição detalhada dos elementos da ontologia, como classes, propriedades, restrições lógicas e regras, além de oferecer suporte à validação semântica por meio de raciocinadores automáticos, como Hermit e Pellet. Também possibilita a adição de anotações e comentários. Com essa abordagem, a fase de Implementação assegura que a ontologia de referência construída nas fases anteriores seja convertida em uma ontologia operacional interpretável por máquina, formalmente codificada, validada e documentada, pronta para ser reutilizada, estendida ou integrada em aplicações.

Durante o desenvolvimento, a ontologia implementada será salva em arquivos no formato OWL e versionada em um repositório Git. As atualizações são realizadas por meio de *commits*, nos quais cada alteração feita no Protégé é registrada com mensagens descritivas, permitindo rastrear o histórico de mudanças. Dessa forma todos os envolvidos no projeto têm acesso à versão mais recente e podem colaborar.

O uso do Git, garante não apenas o armazenamento seguro dos artefatos, mas também a rastreabilidade completa de todas as modificações realizadas ao longo do ciclo de desenvolvimento. Esse controle de versão possibilita a recuperação de estados anteriores da ontologia, facilita o acompanhamento das alterações e assegura a integridade dos arquivos. Além disso, o compartilhamento do repositório com a equipe promove um trabalho colaborativo estruturado e a manter a consistência do projeto. Assim, o Git se mostra como uma ferramenta que traz um diferencial para o processo de construção, oferecendo confiabilidade e organização, automaticamente auxiliando na qualidade da ontologia.

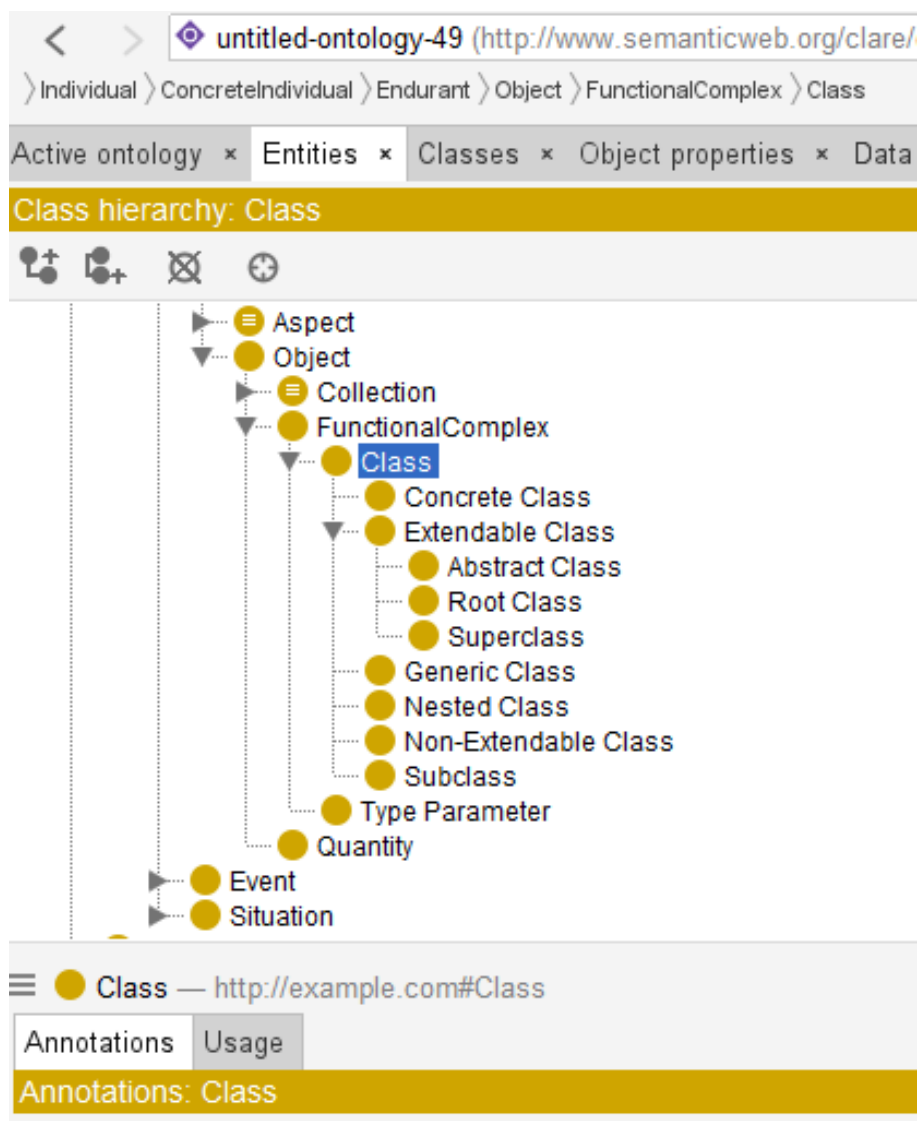


Figura 15 – Representação da Ontologia OOC-O importada no Protégé.

Como parte da exemplificação prática, a Figura 16 mostra a tela do repositório Git no GitHub com o histórico completo de commits, registrando todas as alterações realizadas desde a primeira fase do método SABiOx. Nessa visão consolidada, é possível observar todos os artefatos da atividade de refazer a ontologia OOC-O reunidos em um único repositório, evidenciando a evolução contínua do projeto.

Para publicar a ontologia foi seguido o tutorial de publicação de ontologia do professor João Paulo Almeida (*A simple Semantic Web ontology publication how-to*), disponível em <<https://github.com/jpalmeida/ontology-pub-bp>>. Nesse tutorial são apresentadas considerações sobre boas práticas de publicação de ontologias, é um tutorial simples e prático. O resultado da publicação da OOC-O seguindo os passos do tutorial podem ser vistos em <[https://clareadgomes.github.io/Projeto\\_OOC-O\\_Final/](https://clareadgomes.github.io/Projeto_OOC-O_Final/)>.

A aplicação das ferramentas ao longo das fases do método SABiOx demonstrou que é possível estruturar um fluxo contínuo do início ao fim do ciclo de desenvolvimento

Projeto-Dissertacao---Refazendo-Ontologia-OOC-O Public

Pin Watch 0 Fork 0 Star 0

master had recent pushes 24 seconds ago Compare & pull request

master 2 Branches 0 Tags Go to file Add file Code

This branch is 6 commits ahead of, 1 commit behind main. Contribute

File Name	Description	Commit Time
Class Diagram1.jpg	Modelo conceitual OOC-O Visual paradigm	15 minutes ago
OOC-OCClassModule.jpg	Módulos da Ontologia OOC-O	20 minutes ago
OntologiaOOC-O.ttl	Código turtle gerado Visual Paradigm Plugin OntoUML	13 minutes ago
Ontology_Specification_Document_OOC_O.zip	Documento de Especificação da Ontologia	33 minutes ago
Operational_Ontology_Document_OOC_O.zip	Documento Operacional da Ontologia	7 minutes ago
Reference_Ontology_Document_OOC_O.zip	Documento de Referência da Ontologia	11 minutes ago

About: No description, website, or topics provided. Activity: 0 stars, 0 watching, 0 forks.

Releases: No releases published. Create a new release.

Packages: No packages published. Publish your first package.

Figura 16 – Github com documentação e arquivos da Ontologia OOC-O

de ontologias. A utilização integrada de editores  $\text{\LaTeX}$ , Visual Paradigm com plugin OntoUML, Protégé e sistemas de controle de versão como o Git respondeu a uma das principais dificuldades apontadas no survey: a ausência de integração entre as ferramentas utilizadas em cada etapa. Embora existam outras ferramentas que poderiam ser empregadas, optou-se por adotar as mais votadas no *survey* ou uma opção mais dinâmica (como o editor  $\text{\LaTeX}$  ao invés de editores de texto e planilhas) e o Git (via plataforma GitHub), considerando a familiaridade já existente com este repositório.

Ainda que a interoperabilidade entre as ferramentas não seja completamente automatizada, exigindo, em alguns momentos, esforço manual na transição e adequação dos artefatos entre as etapas, a sequência proposta garante que os artefatos gerados em uma fase possam orientar e alimentar as fases seguintes, mantendo rastreabilidade e consistência das informações. O uso do Git possibilitou o versionamento sistemático de todos os documentos e modelos, o registro do histórico de alterações e o compartilhamento dos materiais com todos os envolvidos, favorecendo o trabalho em equipe ao permitir acesso simultâneo e colaboração contínua.

A exemplificação com a Ontologia OOC-O demonstrou que essa integração, mesmo parcial, contribui para reduzir esforços manuais, padronizar processos e reforçar a confiabilidade e a qualidade da ontologia construída. A proposta apresenta um uso de ferramentas que cobre todas as fases do método, oferecendo suporte durante todo o processo de desenvolvimento da ontologia, resultando em um fluxo de trabalho organizado, colaborativo e alinhado às necessidades da Engenharia de Ontologia.

## 4.6 Avaliação

A avaliação das ferramentas utilizadas nesta dissertação foi conduzida em duas etapas complementares. Na primeira, analisou-se de forma geral o suporte de cada ferramenta às fases e atividades do método SABiOx — Requisitos, Configuração, Captura, Design e Implementação — verificando se ofereciam suporte total, parcial ou inexistente a cada etapa do processo. Na segunda, aprofundou-se a análise por meio da aplicação de critérios de qualidade, baseados em propriedades reconhecidas em padrões de software e adaptadas ao contexto das ferramentas de apoio, de modo a avaliar não apenas a cobertura funcional, mas também permitindo uma visão mais ampla de suas contribuições para o desenvolvimento.

O objetivo dessa primeira análise é fornecer uma visão geral do alcance funcional de cada ferramenta em relação ao método, permitindo identificar quais etapas são atendidas por cada ferramenta. A Tabela 4 apresenta de forma resumida essa avaliação, indicando para cada ferramenta as fases do método em que foi utilizada e o nível de suporte fornecido em cada fase (Total, Parcial ou Sem suporte).

Tabela 4 – Ferramentas utilizadas e suporte geral nas fases do método SABiOx

Ferramenta	Fases SABiOx atendidas	Suporte
Editor L <sup>A</sup> T <sub>E</sub> X	Requisitos, Configuração, Captura, Design	Total
Visual Paradigm	Captura	Parcial
OntoUML-vp-plugin	Captura	Total
Protégé	Implementação	Total
Git	Requisitos, Configuração, Captura, Design, Implementação	Total

A análise das ferramentas mostra que o Editor L<sup>A</sup>T<sub>E</sub>X oferece suporte completo às fases de Requisitos, Configuração, Captura e Design. O Visual Paradigm apoia parcialmente apenas a fase de Captura, sendo parcial porque depende do plugin OntoUML para realizar a exportação em OWL, recurso ideal para ontologias. O OntoUML-vp-plugin cobre totalmente a fase de Captura, funcionando exclusivamente em conjunto com o Visual Paradigm e instalado apenas para esse objetivo. O Protégé é totalmente adequado para a fase de Implementação. Já o Git apresenta cobertura total em todas as fases do método, atuando não como ferramenta de construção, mas garantindo suporte ao versionamento e controle das alterações. Podemos observar, portanto, quais fases cada ferramenta consegue atender integralmente, quais demandam complementação e como a combinação dessas ferramentas pode garantir a cobertura completa do método SABiOx.

Para complementar a análise, a Tabela 5 apresenta uma síntese detalhada dos resultados obtidos a partir da aplicação de um conjunto de critérios baseado em propriedades de

que são esperadas de se encontrar em ferramentas de suporte de Engenharia de Ontologias (EO): adequação funcional, eficiência, usabilidade, confiabilidade, segurança, facilidade de uso, colaboração, cobertura. A avaliação foi feita pela autora desta dissertação e as propriedades são baseadas na ISO 25010 (ISO, 2011) (adaptadas para melhor se adequarem às ferramentas de EO), já mencionadas na Seção 3. A partir dessas propriedades, foram formuladas perguntas norteadoras, apresentadas a seguir, que guiaram a avaliação de cada ferramenta quanto ao atendimento de cada atributo, garantindo uma análise sistemática e consistente.

- **C1 - Adequação Funcional:** A ferramenta oferece os recursos necessários para apoiar a execução das atividades previstas na fase em que é utilizada?
- **C2 - Eficiência:** A ferramenta contribui para agilizar a execução da fase em que é utilizada?
- **C3 - Usabilidade:** A interface da ferramenta é clara e intuitiva, permitindo que usuários com diferentes níveis de experiência consigam utilizá-la sem dificuldades?
- **C4 - Confiabilidade:** A ferramenta apresenta comportamento estável e consistente, produzindo resultados corretos sem falhas frequentes?
- **C5 - Segurança:** A ferramenta dispõe de mecanismos adequados para proteger dados e informações durante o desenvolvimento de ontologias, quando aplicável?
- **C6 - Manutenibilidade:** A ferramenta mantém continuidade no desenvolvimento por meio de atualizações ou novas versões?
- **C7 - Portabilidade:** A ferramenta pode ser utilizada em diferentes ambientes ou sistemas operacionais sem restrições significativas?
- **C8 - Disponibilidade de Uso:** A ferramenta está amplamente acessível, com versões gratuitas ou facilmente obtidas, permitindo sua utilização sempre que necessário?
- **C9 - Colaboração:** A ferramenta possibilita o trabalho colaborativo, permitindo que múltiplos usuários atuem de forma integrada nas atividades?
- **C10 - Cobertura:** A ferramenta atende de maneira abrangente às necessidades de desenvolvimento de ontologias, oferecendo funcionalidades suficientes para as fases em que será utilizada?
- **C11 - Interoperabilidade:** A ferramenta permite integração com outras ferramentas ou exportação/importação de dados em formatos compatíveis com padrões de ontologia?

- **C12 - Documentação e Suporte:** A ferramenta oferece documentação adequada, tutoriais ou recursos de suporte que auxiliem os usuários na execução das atividades?
- **C13 - Controle de Versão:** A ferramenta possibilita rastrear alterações, manter histórico de versões e gerenciar mudanças durante o desenvolvimento das ontologias?
- **C14 - Reúso:** A ferramenta permite reaproveitar modelos, diagramas ou componentes de ontologia, contribuindo para acelerar o processo de desenvolvimento?

Tabela 5 – Avaliação das ferramentas em relação às propriedades.

Propriedade	$\LaTeX$	Visual Pa- radigm	OntoUML- vp-plugin	Protégé	Git
Adequação Funcional	AT	AT	AT	AT	AT
Eficiência	AT	AT	AT	AT	AT
Usabilidade	AT	AP	AP	AP	AP
Confiabilidade	AT	AT	AT	AT	AT
Manutenibilidade	AT	AT	AP	AT	AT
Segurança	AP	AT	NA	AT	AT
Portabilidade	AT	AT	AT	AT	AT
Disponibilidade de Uso	AT	AT	AT	AT	AT
Colaboração	AT	AP	NA	AP	AT
Cobertura	AT	AT	AT	AT	AT
Interoperabilidade	AP	AP	AT	NA	NA
Documentação e Suporte	AT	AT	AT	AT	AT
Controle de Versão	AP	NA	NA	AP	AT
Reúso	AT	AT	NA	AT	NA

A análise da Tabela 5 evidencia o comportamento de cada ferramenta em relação às propriedades de qualidade. Para facilitar a compreensão da avaliação, descrevemos de que forma cada propriedade foi atendida (total – AT, parcial – AP ou não atendida – NA) apresentando as características específicas da ferramenta que justificam esse resultado. Na sequência, são descritas as ferramentas selecionadas, bem como os resultados da avaliação conduzida a partir das propriedades de qualidade estabelecidas como critérios de análise.

## Editor $\LaTeX$

Na primeira ferramenta, o **Editor  $\LaTeX$** , observa-se atendimento total ou parcial em todas as propriedades. Em relação à **adequação funcional**, a ferramenta se destaca pela possibilidade de anotação e edição de texto, sendo essencial para realizar as atividades em que ela está sendo utilizada. Quanto à **eficiência**, a edição de textos estruturados é

realizada de forma rápida, poder rever e alterar o documento é prático também. A **usabilidade** apresenta um desafio inicial levando em consideração que a interface baseada em código exige aprendizado prévio, dificultando a adoção por usuários iniciantes, mas com uma dedicação inicial para aprendizado ele se mostra simples de usar. No que se refere à **confiabilidade**, o ambiente é estável.

A **manutenibilidade** pode ser observada pela atualização de pacotes que podem ser atualizados e em versão online sofrem atualizações constantes, sem o usuário dar conta. A **segurança** pode ser assegurada através de senha e compartilhamento somente com os envolvidos. A **portabilidade** é atendida, visto que a ferramenta pode ser utilizada em diferentes ambientes. No critério **disponibilidade**, o editor  $\text{\LaTeX}$  é gratuito e amplamente acessível, apresentando algumas limitações mas que não interferem na sua utilização para o propósito aqui descrito. No que tange à **colaboração**, permite a edição de mais de um usuário por vez, que varia de versão gratuita para paga. A **cobertura** é atendida quando a ferramenta consegue entregar o artefato sugerido pelo método SABiOx.

Quando indicamos que a **interoperabilidade** é atendida de forma parcial, isso se deve ao fato de que não ocorre de maneira automática. Contudo, é possível integrar dados de outras fases e atividades, bem como inserir imagens e códigos gerados em outras ferramentas. Além disso, destaca-se a possibilidade de retornar ao documento e encontrar todas as informações devidamente registradas. Em contrapartida, a ferramenta é bastante robusta em **documentação e suporte**, contando com extensa documentação e uma comunidade ativa. Sobre o **controle de versão** pode se dizer que atende parcialmente ao possuir um controle de alterações com modificações feitas, com data e nome de quem realizou tais alterações. Por fim, destaca-se o *reúso*, viabilizado pela utilização de *templates* e pacotes que podem ser aplicados em diferentes projetos.

## Visual Paradigm

Na segunda ferramenta, o **Visual Paradigm**, observa-se atendimento total ou parcial em quase todas propriedades avaliadas. Em relação à **adequação funcional**, a ferramenta se destaca pela modelagem de diagramas e pela representação visual de conceitos, sendo adequada para essa fase. Quanto à **eficiência**, as operações dentro da ferramenta agilizam a construção do modelo conceitual de forma organizada. No critério de **usabilidade**, apresenta uma interface gráfica que facilita a interação, embora exija certa familiarização inicial por parte do usuário, por isso foi avaliado como atendendo parcialmente. No que se refere à **confiabilidade**, o ambiente é estável, atendendo de forma consistente às operações de modelagem.

A **manutenibilidade** é observada nos lançamentos de novas versões documentadas na página oficial da ferramenta. Em relação à **segurança**, pode-se dizer que no uso local depende do usuário e no online está protegido por senha, portanto o acesso só será

permitido com autorização. No que diz respeito à **portabilidade**, pode ser utilizada em diferentes sistemas operacionais, atendendo de forma satisfatória a esse critério. No quesito **disponibilidade**, possui versão gratuita e disponível que atende ao propósito dessa fase e para versões mais completas somente a versão paga.

Quanto à **colaboração**, é possível na versão online e na versão desktop essa opção não é possível. Em relação à **cobertura**, a ferramenta entrega o artefato pronto sugerido por essa atividade dentro da fase. Sobre a **interoperabilidade**, não há integração automática com outras ferramentas, o que limita sua utilização, mas oferece alguns formatos de exportação que permitem que sejam anexadas em documentos gerados em outra ferramenta, como por exemplo no Editor L<sup>A</sup>T<sub>E</sub>X. Nesse quesito ainda o Visual Paradigm conta com instalação de plugins que atende essa propriedade, como é o caso do OntoUML-vp-plugin indicado nesse trabalho. No critério de **documentação e suporte**, a ferramenta apresenta documentação oficial e materiais de apoio, além de tutoriais que auxiliam no aprendizado. No que diz respeito ao **controle de versão** não há nenhum suporte. Por fim, com relação ao **reúso**, é possível reutilizar modelos já feitos anteriormente.

## OntoUML VP Plugin

O **OntoUML-vp-plugin** oferece atendimento total ou parcial nas propriedades. Em relação à **adequação funcional**, atende totalmente ao propósito pelo qual foi utilizado, que é permitir a exportação para gUFO. Quanto à **eficiência**, apresenta bom desempenho ao permitir a exportação do modelo construído.

No critério de **usabilidade**, observa-se que a interface gráfica pertencente a ele é bem amigável e intuitiva. Em termos de **confiabilidade**, a ferramenta se mostra estável no ambiente em que está inserida. A **manutenibilidade** é atendida parcialmente pois a ferramenta não recebe uma nova versão desde o ano de 2021. No quesito **portabilidade**, pode ser utilizada em diferentes plataformas, acompanhando a compatibilidade do Visual Paradigm. Em relação à **disponibilidade**, o plugin é gratuito e amplamente utilizado.

Quanto à **cobertura**, atende de forma satisfatória. Sobre a **interoperabilidade**, destaca-se pela possibilidade de exportar os modelos para formatos compatíveis com outras ferramentas. Em relação à **documentação e suporte**, conta com materiais de apoio e publicações associadas à sua comunidade de desenvolvimento, mas de maneira mais restrita quando comparada a ferramentas consolidadas, porém se justifica pelo fato de não ser uma ferramenta propriamente dita, e sim um *plugin* que deve ser usado juntamente com uma outra ferramenta. Por fim, com relação aos critérios **segurança**, **colaboração**, **controle de versão** e ao **reúso**, podemos dizer que não temos como avaliar essas propriedades, pois não é uma ferramenta que funciona sozinha.

## Protégé

Na ferramenta **Protégé**, observa-se atendimento total ou parcial em todos os critérios. Em relação à **adequação funcional**, destaca-se como um dos principais editores de ontologias, permitindo a criação, edição e gerenciamento de classes, propriedades e instâncias, atendendo de forma satisfatória às necessidades propostas por essa fase. Quanto à **eficiência**, apresenta bom desempenho na execução das operações proporcionando agilidade no desenvolvimento de ontologias. No critério de **usabilidade**, oferece uma interface gráfica que facilita a interação, embora exija conhecimento prévio em modelagem ontológica, o que pode representar uma barreira inicial para usuários iniciantes, por isso esse critério foi avaliado como atendendo parcialmente. No que diz respeito à **confiabilidade**, a ferramenta é estável e com consistência nos resultados gerados.

A **manutenibilidade** é considerada total por estar sempre recebendo atualizações. No quesito **segurança**, no uso local depende da segurança do usuário e na versão online é protegida por senha. Em relação à **portabilidade**, pode ser utilizada em diferentes sistemas operacionais. Quanto à **disponibilidade**, o Protégé é gratuito e de código aberto, o que facilita seu acesso e utilização sem restrições de licença.

No que se refere à **colaboração**, o suporte é limitado, não permitindo edição simultânea de forma nativa. Já em relação à **cobertura**, atende às demandas da fase de implementação do método SABiOx totalmente. Sobre a **interoperabilidade**, ele permite importação de arquivos em diversos formatos, neste trabalho utilizamos a importação do arquivo em turtle (.ttl) com o código em owl, tendo obtido sucesso nessa atividade. Em relação à **documentação e suporte**, possui extensa documentação, fóruns e uma comunidade ativa que auxilia na resolução de dúvidas e no aprendizado. Quanto ao **controle de versão**, atende parcialmente ao inserir data e horário quando salva um arquivo. Finalizando, no quesito **reúso**, o Protégé se destaca pela possibilidade de utilizar ontologias já desenvolvidas em novos projetos de construção.

## Git

Por fim, foi avaliado o **Git**, mesmo sendo usado como ferramenta de apoio e não de fato de construção. Em relação à **adequação funcional**, destaca-se por oferecer mecanismos completos de versionamento, permitindo o registro detalhado de todas as alterações realizadas nos artefatos do projeto. Esse recurso garante rastreabilidade e controle sobre cada modificação realizada em todas as fases do método. Quanto à **eficiência**, a ferramenta possibilita operações rápidas e eficazes de atualização, sincronização e recuperação de versões, o que é importante em um trabalho de equipe.

No que se refere à **usabilidade**, o Git apresenta uma curva de aprendizado acentuada, especialmente para usuários iniciantes, visto que sua utilização é baseada em

comandos ou em plataformas gráficas que exigem familiarização. Contudo, após essa fase inicial, a ferramenta se mostra simples e prática para se usar. Em termos de **confiabilidade**, é reconhecida como uma solução estável.

A **manutenibilidade** é atendida totalmente com atualizações constantes. No aspecto da **segurança**, o Git permite a configuração feita pelo usuário de recursos relacionados. Quanto à **portabilidade**, pode ser utilizado em diferentes sistemas operacionais. Já no quesito **disponibilidade**, trata-se de uma ferramenta gratuita e de código aberto, amplamente acessível e consolidada.

Em relação à **colaboração**, o Git se mostra bastante eficiente, pois permite que múltiplos usuários trabalhem simultaneamente em diferentes versões e ramificações, consolidando alterações de maneira organizada. No critério de **cobertura**, atende satisfatoriamente às necessidades do método SABiOx, na medida em que possibilita manter e gerenciar todos os artefatos gerados durante todo o processo de construção. No que se refere à **documentação e suporte**, a ferramenta conta com extensa documentação oficial, além de uma comunidade ativa que contribui com guias e tutoriais. Quanto à **interoperabilidade** e ao **reúso**, podemos dizer que são propriedades que não temos como avaliar com o uso para o qual foi destinada nesse projeto.

## Conclusões

A análise realizada demonstra que cada ferramenta avaliada apresenta pontos fortes específicos, mas também limitações quando considerada isoladamente. As ferramentas genéricas, como o **Editor L<sup>A</sup>T<sub>E</sub>X** e o **Git**, mostraram-se adequadas para atender às demandas de documentação, versionamento e colaboração, garantindo confiabilidade e rastreabilidade ao processo. Já as ferramentas específicas, como o **Visual Paradigm** e o **OntoUML Plugin**, revelaram-se essenciais para a modelagem e representação formal, ainda que dependam do suporte das ferramentas genéricas para assegurar a completude do método.

O maior desafio identificado refere-se à **interoperabilidade**, uma vez que a integração entre as ferramentas não ocorre de forma automática. Apesar disso, observou-se que a combinação dos recursos disponíveis possibilitou a troca de informações entre fases, ainda que de maneira manual, garantindo que os artefatos produzidos alimentassem as etapas subsequentes.

Dessa forma, conclui-se que, embora nenhuma ferramenta individual atenda integralmente às necessidades do método SABiOx, o uso combinado das soluções analisadas mostrou-se **viável, consistente e aplicável**, assegurando o desenvolvimento das ontologias durante todo o ciclo de vida.

## 5 Considerações Finais

Nos últimos anos surgiram muitos métodos e ferramentas para apoiar o desenvolvimento de ontologias, apesar de alguns já serem consolidados ainda há uma lacuna no que se refere à integração de ferramentas que possam sustentar todas as fases do processo. Baseado nisso, este trabalho teve como objetivo principal instrumentar as fases do método SABiOx utilizando ferramentas de apoio para a construção de ontologias. Ao longo do trabalho, buscou-se não apenas compreender o potencial de cada ferramenta isoladamente, mas também avaliar de que forma poderia ser integrada ao método e interoperar com outras ferramentas, a fim de conseguir suporte em todas etapas do método e assegurar a qualidade dos artefatos gerados.

A pesquisa foi estruturada em etapas bem definidas. Inicialmente, realizou-se uma revisão teórica que permitiu compreender os fundamentos das ontologias, sua importância, características e classificações e os desafios relacionados à sua construção. Em seguida foram detalhados diferentes métodos de construção, tendo o método SABiOx se destacado por oferecer um ciclo de vida estruturado em fases (Requisitos, Configuração, Captura, Design e Implementação) cada uma com um conjunto de atividades e acrescido de atividades de suporte para auxiliar nas fases principais (aquisição de conhecimento, documentação, gerenciamento de configuração, avaliação, reutilização e publicação). Em seguida, investigamos e relacionamos diversas ferramentas de apoio ao desenvolvimento de ontologias, primeiro buscando na literatura ferramentas existentes, características e utilizações. Em complemento com os dados apresentados na literatura foi realizado um *survey* onde investigamos quais ferramentas foram usadas, suas principais características, benefícios e dificuldades percebidos ao usá-las e propriedades consideradas importantes em ferramentas de Engenharia de Ontologias (EO). Os resultados indicaram que as ferramentas genéricas continuam sendo usadas, especialmente na fase de Requisitos, demonstrando que, para certas atividades, soluções simples ainda prevalecem. Outro resultado do *survey* foi a constatação de que nas ferramentas específicas não existe interoperabilidade plena, o que leva a esforços manuais para transferência de artefatos e adaptação de formatos. As ferramentas foram amplamente adotadas por ontologistas, mas algumas questões, como problemas de integração e limitações de escopo, ainda precisam ser abordadas.

A aplicação prática do método SABiOx, utilizando a ontologia OOC-O como exemplo, permitiu verificar na prática a instrumentação do método proposto. Cada fase foi associada a um conjunto de ferramentas específicas, escolhidas por suas funcionalidades e adequação às demandas do método. Essa implementação mostrou que, mesmo sem integração técnica completa, é possível estabelecer uma integração lógica coerente, em que os artefatos gerados em cada fase alimentam a fase seguinte com o mínimo de retrabalho

possível. Esse alinhamento conceitual contribuiu para garantir consistência, rastreabilidade e clareza na evolução da ontologia.

A avaliação dessas ferramentas foi conduzida em duas etapas complementares. A primeira buscou verificar em quais fases do método cada ferramenta oferece suporte total, parcial ou inexistente. A segunda dimensão consistiu na aplicação de propriedades definidas pela ISO 25010 adaptadas, propriedades estas que são esperadas por ontologistas em ferramentas de apoio ao desenvolvimento de ontologias. Essa abordagem integrada possibilitou uma visão abrangente da contribuição das ferramentas, destacando características, funcionalidades e limitações.

Os resultados obtidos indicam que nenhuma ferramenta, isoladamente, atende a todas as fases e atividades definidas pelo método SABiOx. No entanto, observou-se que a utilização combinada das soluções analisadas garante a execução completa do processo, mesmo que algumas etapas ainda dependam de esforços manuais ou de complementações. Um dos principais desafios identificados diz respeito à interoperabilidade, visto que a integração entre as ferramentas ainda não ocorre de maneira automatizada. Apesar disso, o fluxo de informações entre as fases foi viabilizado, permitindo que os artefatos gerados fossem reutilizados ou adaptados conforme a necessidade, garantindo a consistência do processo.

Em relação aos objetivos específicos propostos, considera-se que foram alcançados da seguinte forma:

- O primeiro objetivo, que consistiu em identificar e selecionar ferramentas existentes que pudessem ser utilizadas de forma eficaz nas diferentes fases do método SABiOx, foi atendido por meio da revisão da literatura e da aplicação do *survey*, que permitiram levantar as ferramentas mais utilizadas por ontologistas, suas características, benefícios e limitações.
- O segundo objetivo, que visou analisar as ferramentas indicadas e apresentar formas de integrá-las ao método SABiOx, foi atendido por meio do mapeamento das ferramentas às fases do método, evidenciando como cada uma pode contribuir para os resultados esperados em cada etapa e como os artefatos gerados podem ser utilizados nas fases seguintes, mesmo sem integração totalmente automatizada.
- Por fim, o terceiro objetivo, que consistiu em avaliar a utilização das ferramentas integradas ao método SABiOx por meio de aplicação prática, foi atendido por meio da reconstrução da ontologia OOC-O com o apoio das ferramentas selecionadas, permitindo observar, na prática, o funcionamento da instrumentação proposta, bem como as dificuldades, limitações e contribuições das ferramentas ao longo do processo.

Uma das principais limitações deste trabalho e que deve ser mencionada foi a impossibilidade de avaliar todas as ferramentas existentes, bem como ao fato de a aplicação ter sido realizada em um único cenário e pela própria autora do trabalho. Embora tenha sido possível demonstrar a utilização das ferramentas no contexto analisado, não é possível afirmar que, em todos os cenários, os resultados serão semelhantes ou que não haverá dificuldades adicionais, considerando a ampla variedade de domínios, métodos e ferramentas existentes. Para amenizar essa limitação foi adotada a utilização das ferramentas mais indicadas no *survey*, pois assim possibilitou apresentar como ferramentas já conhecidas e consolidadas podem se complementar e atender a um método completo. Além disso, optou-se pela estratégia de refazer uma ontologia já existente, o que possibilitou a comparação entre a ontologia previamente desenvolvida e a ontologia reconstruída com o apoio das ferramentas analisadas, permitindo uma avaliação mais consistente do processo e dos resultados obtidos.

O estudo indica que a utilização do método SABiOx pode ser potencializada pela seleção e uso de ferramentas adequadas a cada fase, mostrando que sua instrumentalização, conforme o objetivo principal da pesquisa, não apenas reforçou sua utilidade, mas também ampliou seu valor prático. A associação do método a ferramentas já consolidadas e amplamente conhecidas na área o tornou mais completo e prático, fazendo que se apresente como um referencial não apenas funcional, mas também com aplicabilidade, o que o torna mais interessante para ser utilizado no desenvolvimento de ontologias.

## 5.1 Contribuições

As principais contribuições deste trabalho são:

- Apresentação de uma seleção de ferramentas de apoio ao desenvolvimento de ontologias, apontados pela literatura e utilizados por especialistas em ontologias por meio da aplicação de um *survey*;
- Instrumentação do método SABiOx com ferramentas conhecidas e já consolidadas, apresentando uma possível integração entre as fases do ciclo de vida da ontologia, destacando que o processo não ocorre de forma totalmente automática, mas com o apoio e intervenção do especialista ao longo das etapas;
- Validação prática por meio da ontologia OOC-O, que demonstrou a viabilidade do método apesar de limitações de interoperabilidade técnica;
- Adaptação de critérios de avaliação da ISO 25010 ao contexto de Engenharia de Ontologias, permitindo uma análise sistemática de ferramentas;
- Indicação de que, embora nenhuma ferramenta cubra todas as fases do processo

isoladamente, a combinação delas pode contribuir para a consistência, rastreabilidade e abrangência ao processo.

## 5.2 Trabalhos Futuros

Ao realizar o trabalho descrito nesta dissertação, foram identificadas algumas oportunidades de trabalhos futuros, a saber:

- Desenvolver *plugins*, APIs ou outras soluções automáticas para aprimorar a interoperabilidade entre ferramentas, reduzindo a necessidade de retrabalho manual e aumentando a eficiência do processo;
- Ampliar o escopo de testes para ontologias de diferentes domínios e níveis de complexidade, de modo a verificar a robustez, a flexibilidade e a adaptabilidade das soluções propostas;
- Investigar o uso de tecnologias emergentes, como por exemplo inteligência artificial, para apoiar a captura de conhecimento, a avaliação automática e a integração de ferramentas de forma mais eficiente;
- Desenvolver ambientes unificados que concentrem as funcionalidades necessárias em uma única solução ou em um número reduzido de soluções interoperáveis, proporcionando maior usabilidade e simplificação do processo de modelagem;
- Explorar padrões de representação e interoperabilidade que facilitem a troca, o reaproveitamento e a padronização de artefatos entre diferentes ferramentas;
- Aprimorar mecanismos de colaboração em tempo real, permitindo que múltiplos especialistas participem da construção de ontologias de forma síncrona e integrada.

## Referências

- ABBURU, S.; BABU, G. S. Survey on ontology construction tools. *International Journal of Scientific & Engineering Research*, v. 4, n. 6, p. 1748–1752, 2013. Available on: <<https://ufes.es/abburu-babu-injasr2013>>. Citado 5 vezes nas páginas 14, 15, 25, 32 e 49.
- AGUIAR, C. Z. d.; SOUZA, V. E. S. SABiOx: the Extended Systematic Approach for Building Ontologies. In: *Proc. of the 17th Seminar on Ontology Research in Brazil (ONTOBRAS 2024)*. Vitória, ES, Brazil: CEUR, 2024. Citado 5 vezes nas páginas 14, 16, 23, 24 e 48.
- AGUIAR, C. Z. de; FALBO, R. de A.; SOUZA, V. E. S. Ooc-o: A reference ontology on object-oriented code. In: SPRINGER. *International Conference on Conceptual Modeling*. [S.l.], 2019. p. 13–27. Citado 4 vezes nas páginas 17, 48, 49 e 52.
- ALMEIDA, J. *et al.* *gUFO: A Lightweight Implementation of the Unified Foundational Ontology (UFO)*. 2019. Available on: <<http://purl.org/nemo/doc/gufo>>. Citado na página 55.
- ALMEIDA, M. B. Roteiro para a construção de uma ontologia bibliográfica através de ferramenta automatizada. *Perspectivas em ciência da informação*, v. 8, n. 2, 2003. Citado na página 31.
- ALOBAID, A. *et al.* Automating ontology engineering support activities with ontology. *Journal of Web Semantics*, Elsevier, v. 57, p. 100472, 2019. Citado 2 vezes nas páginas 45 e 46.
- ARPÍREZ, J. C. *et al.* Webode: a scalable workbench for ontological engineering. In: *Proceedings of the 1st international conference on Knowledge capture*. [S.l.: s.n.], 2001. p. 6–13. Citado na página 15.
- ARRIETA, A. B. *et al.* Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, Elsevier, v. 58, p. 82–115, 2020. Citado na página 19.
- BLOMQUIST, E.; HAMMAR, K.; PRESUTTI, V. Engineering ontologies with patterns-the extreme design methodology. *Ontology Engineering with Ontology Design Patterns*, v. 25, p. 23–50, 2016. Citado na página 14.
- CARDOSO, J. The semantic web vision: Where are we? *IEEE Intelligent systems*, IEEE, v. 22, n. 5, p. 84–88, 2007. Citado 2 vezes nas páginas 15 e 16.
- CONFALONIERI, R.; GUIZZARDI, G. On the multiple roles of ontologies in explainable ai. *arXiv preprint arXiv:2311.04778*, 2023. Citado na página 19.
- CORCHO, O.; FERNÁNDEZ-LÓPEZ, M.; GÓMEZ-PÉREZ, A. Methodologies, tools and languages for building ontologies. where is their meeting point? *Data & knowledge engineering*, Elsevier, v. 46, n. 1, p. 41–64, 2003. Citado 6 vezes nas páginas 15, 16, 31, 44, 45 e 46.

- CORRENDO, G.; ALANI, H. Survey of tools for collaborative knowledge construction and sharing. In: IEEE. *2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Workshops*. [S.l.], 2007. p. 7–10. Citado na página 32.
- D'AQUIN, M. *et al.* Designing core ontologies. *Applied Ontology*, SAGE Publications Sage UK: London, England, v. 6, n. 3, p. 177–221, 2011. Citado na página 20.
- DING, Y. A review of ontologies with the semantic web in view. *Journal of information science*, Sage Publications Sage CA: Thousand Oaks, CA, v. 27, n. 6, p. 377–384, 2001. Citado na página 19.
- DUDÁŠ, M.; ZAMAZAL, O.; SVÁTEK, V. Roadmapping and navigating in the ontology visualization landscape. In: SPRINGER. *Knowledge Engineering and Knowledge Management: 19th International Conference, EKAW 2014, Linköping, Sweden, November 24-28, 2014. Proceedings 19*. [S.l.], 2014. p. 137–152. Citado na página 32.
- DZBOR, M. *et al.* Developing ontologies in owl: An observational study. 2006. Citado 2 vezes nas páginas 45 e 46.
- DZIWIŚ, G. *et al.* Ontoflow: A user-friendly ontology development workflow. In: *SemIIM*. [S.l.: s.n.], 2022. Citado 2 vezes nas páginas 45 e 46.
- EASTERBROOK, S. *et al.* Selecting empirical methods for software engineering research. *Guide to advanced empirical software engineering*, Springer, p. 285–311, 2008. Citado na página 35.
- FALBO, R. A. SABiO: Systematic Approach for Building Ontologies. In: GUIZZARDI, G. *et al.* (Ed.). *Proc. of the Proceedings of the 1st Joint Workshop ONTO.COM / ODISE on Ontologies in Conceptual Modeling and Information Systems Engineering*. [S.l.]: CEUR, 2014. Citado 6 vezes nas páginas 14, 20, 21, 23, 24 e 44.
- FALBO, R. de A. *et al.* Organizing ontology design patterns as ontology pattern languages. In: SPRINGER. *The Semantic Web: Semantics and Big Data: 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings 10*. [S.l.], 2013. p. 61–75. Citado 2 vezes nas páginas 10 e 20.
- FELLER, J. *Perspectives on free and open source software*. [S.l.]: MIT Press, 2005. Citado na página 44.
- FERNÁNDEZ-LÓPEZ, M.; GÓMEZ-PÉREZ, A.; JUZGADO, N. J. Methontology: from ontological art towards ontological engineering. American Association for Artificial Intelligence, 1997. Citado na página 21.
- FITZGERALD, B. Has open source software a future? 2005. Citado na página 44.
- GANGEMI, A. Some tools and methodologies for domain ontology building. *Comparative and Functional Genomics*, v. 4, n. 1, p. 104, 2003. Citado na página 31.
- GANGEMI, A.; PRESUTTI, V. Ontology design patterns. In: *Handbook on ontologies*. [S.l.]: Springer, 2009. p. 221–243. Citado na página 14.
- GOMES, C. A. D. *et al.* Investigating tool usage in ontology engineering: a survey. 2024. Citado 2 vezes nas páginas 17 e 35.

- GOMES, C. A. D. *et al.* Investigating tool usage in ontology engineering: a survey - protocol & data extraction. 2024. Citado na página 36.
- GÓMEZ-PÉREZ, A.; FERNÁNDEZ-LÓPEZ, M.; CORCHO, O. *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. [S.l.]: Springer, 2004. Citado 2 vezes nas páginas 28 e 29.
- GÓMEZ-PÉREZ, A.; FERNÁNDEZ-LÓPEZ, M.; CORCHO, O. Ontology tools. *Ontological Engineering: With Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web*, Springer, p. 293–362, 2004. Citado na página 21.
- GRUBER, T. R. A translation approach to portable ontology specifications. *Knowledge acquisition*, Elsevier, v. 5, n. 2, p. 199–220, 1993. Citado na página 19.
- GUARINO, N. *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*. [S.l.]: IOS press, 1998. v. 46. Citado 4 vezes nas páginas 10, 14, 19 e 20.
- GUARINO, N. *et al.* Towards a Methodology for Ontology Based Model Engineering. *Proceedings of International Workshop on Model Engineering (IWME-2000): Nice, France: 2000, June, 13*, 2000. Citado na página 21.
- GUERSON, J. *et al.* Ontouml lightweight editor: a model-based environment to build, evaluate and implement reference ontologies. In: IEEE. *2015 IEEE 19th international enterprise distributed object computing workshop*. [S.l.], 2015. p. 144–147. Citado na página 38.
- GUIZZARDI, G. *et al.* On ontology, ontologies, conceptualizations, modeling languages, and (meta) models. *Frontiers in artificial intelligence and applications*, IOS Press, v. 155, p. 18, 2007. Citado na página 20.
- IQBAL, R. *et al.* An analysis of ontology engineering methodologies: A literature review. *Research journal of applied sciences, engineering and technology*, v. 6, n. 16, p. 2993–3000, 2013. Citado na página 15.
- ISO, I. O. for S. Iso/iec 25010:2011 - systems and software engineering – systems and software quality requirements and evaluation (square) – system and software quality models. *ISO Standards*, Geneva, Switzerland, 2011. ISSN N/A. Available on: <<https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>>. Citado 3 vezes nas páginas 18, 41 e 63.
- JÚNIOR, P. S. S. *et al.* From a scrum reference ontology to the integration of applications for data-driven software development. *Information and Software Technology*, Elsevier, v. 136, p. 106570, 2021. Citado na página 19.
- KATIFORI, A. *et al.* Ontology visualization methods—a survey. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 39, n. 4, p. 10–es, 2007. Citado 2 vezes nas páginas 45 e 46.
- KHONDOKER, M. R.; MUELLER, P. Comparing ontology development tools based on an online survey. In: WORLD CONGRESS ON ENGINEERING 2010 (WCE 2010), LONDON, UK. [S.l.], 2010. Citado na página 33.

- KITCHENHAM, B. A.; BUDGEN, D.; BRERETON, O. P. Using mapping studies as the basis for further research—a participant-observer case study. *Information and Software Technology*, Elsevier, v. 53, n. 6, p. 638–651, 2011. Citado na página 25.
- KOTIS, K. I.; VOUIROS, G. A.; SPILIOPOULOS, D. Ontology engineering methodologies for the evolution of living and reused ontologies: status, trends, findings and recommendations. *The Knowledge Engineering Review*, v. 35, p. e4, 2020. Citado na página 45.
- LAERA, L.; TAMMA, V. „the hitchhiker’s guide to ontology editors”. *AgentLink News*, n. 18, 2005. Citado na página 25.
- MENDONÇA, F. M.; CASTRO, L. P. de. Ontoforinfoscience e onto4alleditor: metodologia e editor de ontologias como facilitadores na construção de ontologias por especialistas do domínio e cientistas da informação. *Advances in Knowledge Representation/Avanços em Representação do Conhecimento*, v. 1, n. 1, p. 145–173, 2021. Citado 2 vezes nas páginas 21 e 29.
- MENDONÇA, F. M. *et al.* Onto4alleditor: a graphic web ontology editor for information science: a graphic web ontology editor for information science. *Frontiers of Knowledge Representation*, v. 1, n. 2, p. 70–94, 2021. Citado 4 vezes nas páginas 11, 19, 29 e 45.
- MENDONÇA, F. M.; SOARES, A. L. Construindo ontologias com a metodologia ontoforinfoscience: uma abordagem detalhada das atividades do desenvolvimento ontológico. *Ciência da Informação*, v. 46, n. 1, 2017. Citado na página 23.
- MOREIRA, J. L. *et al.* Menthor editor: an ontology driven conceptual modeling platform. 2016. Citado na página 38.
- NEUHAUS, F. *et al.* Towards ontology evaluation across the life cycle: The ontology summit 2013. *Applied Ontology*, SAGE Publications Sage UK: London, England, v. 8, n. 3, p. 179–194, 2013. Citado na página 33.
- NICOLA, A. D.; MISSIKOFF, M.; NAVIGLI, R. A proposal for a unified process for ontology building: Upon. In: SPRINGER. *International conference on database and expert systems applications*. [S.l.], 2005. p. 655–664. Citado na página 21.
- NIETO, M. A. M. An overview of ontologies. *Universidad De Las Américas Puebla, Interactive and Cooperative Technologies Lab*, Citeseer, 2003. Citado 2 vezes nas páginas 29 e 31.
- NOPPENS, O.; LIEBIG, T. Ontology patterns and beyond: towards a universal pattern language. In: *Proceedings of the 2009 International Conference on Ontology Patterns-Volume 516*. [S.l.: s.n.], 2009. p. 179–186. Citado na página 14.
- NOY, N. F.; MUSEN, M. A. Evaluating ontology-mapping tools: Requirements and experience. In: *EON*. [S.l.: s.n.], 2002. p. 1–14. Citado na página 25.
- OBERLE, D. How ontologies benefit enterprise applications 5 (6)(2014), 473–491. doi: 10.3233. *SW-130114*. Citado na página 19.
- PERONI, S. A simplified agile methodology for ontology development. In: SPRINGER. *OWL: Experiences and Directions—Reasoner Evaluation: 13th International Workshop*,

OWLED 2016, and 5th International Workshop, ORE 2016, Bologna, Italy, November 20, 2016, Revised Selected Papers 13. [S.l.], 2017. p. 55–69. Citado na página 14.

PINTO, H. S.; TEMPICH, C.; STAAB, S. Ontology engineering and evolution in a distributed world using diligent. *Handbook on ontologies*, Springer, p. 153–176, 2009. Citado na página 22.

POVEDA-VILLALÓN, M. *et al.* Lot: An industrial oriented ontology engineering framework. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 111, p. 104755, 2022. Citado 2 vezes nas páginas 14 e 15.

PUSPANINGRUM, A. S.; ROCHIMAH, S.; AKBAR, R. J. Functional suitability measurement using goal-oriented approach based on iso/iec 25010 for academics information system. *Journal of Information Systems Engineering and Business Intelligence*, v. 3, n. 2, p. 68–74, 2017. Citado na página 46.

RAUTENBERG, S. *et al.* Uma metodologia para o desenvolvimento de ontologias. *RECEN-Revista Ciências Exatas e Naturais*, v. 10, n. 2, p. 237–262, 2008. Citado na página 14.

REGINATO, C. C. *et al.* A goal-oriented framework for ontology reuse. *Applied ontology*, IOS Press, v. 17, n. 3, p. 365–399, 2022. Citado 2 vezes nas páginas 14 e 19.

SANDERS, J. Linux, open source, and software's future. *IEEE software*, IEEE, v. 15, n. 5, p. 88–91, 1998. Citado na página 44.

STAAB, S. *et al.* Knowledge processes and ontologies. *IEEE Intelligent systems*, IEEE, v. 16, n. 1, p. 26–34, 2001. Citado na página 15.

SUÁREZ-FIGUEROA, M. C.; GÓMEZ-PÉREZ, A.; FERNÁNDEZ-LÓPEZ, M. The neon methodology for ontology engineering. In: *Ontology engineering in a networked world*. [S.l.]: Springer, 2011. p. 9–34. Citado na página 14.

SUÁREZ-FIGUEROA, M. C. *et al.* *Ontology Engineering in a Networked World*. [s.n.], 2012. v. 53. 9–34 p. ISSN 1098-6596. ISBN 9788578110796. Available on: <file:///C:/Users/User/Downloads/fvm939e.pdf>. Citado na página 22.

SUBHASHINI, R.; AKILANDESWARI, J. A survey on ontology construction methodologies. *International Journal of Enterprise Computing and Business Systems*, Citeseer, v. 1, n. 1, p. 60–72, 2011. Citado na página 32.

SURE, Y.; ANGELE, J.; STAAB, S. Ontoedit: Multifaceted inferencing for ontology engineering. In: *Journal on Data Semantics I*. [S.l.]: Springer, 2003. p. 128–152. Citado na página 15.

TUDORACHE, T. Ontology engineering: Current state, challenges, and future directions. *Semantic Web*, IOS Press, v. 11, n. 1, p. 125–138, 2020. Citado na página 14.

VIGO, M. *et al.* Overcoming the pitfalls of ontology authoring: Strategies and implications for tool design. *International Journal of Human-Computer Studies*, Elsevier, v. 72, n. 12, p. 835–845, 2014. Citado 4 vezes nas páginas 15, 33, 45 e 46.

WIERINGA, R. *et al.* Requirements engineering paper classification and evaluation

criteria: a proposal and a discussion. *Requirements engineering*, Springer, v. 11, n. 1, p. 102–107, 2006. Citado na página 26.

WOHLIN, C. *et al.* *Experimentation in software engineering*. [S.l.]: Springer Science & Business Media, 2012. Citado na página 35.

ZHANG, L. *et al.* Orient: Integrate ontology engineering into industry tooling environment. In: SPRINGER. *The Semantic Web–ISWC 2004: Third International Semantic Web Conference, Hiroshima, Japan, November 7-11, 2004. Proceedings 3*. [S.l.], 2004. p. 823–837. Citado 2 vezes nas páginas 45 e 46.

ZOUAQ, A.; NKAMBOU, R. A survey of domain ontology engineering: Methods and tools. In: *Advances in intelligent tutoring systems*. [S.l.]: Springer, 2010. p. 103–119. Citado 2 vezes nas páginas 45 e 46.

# Apêndices

# APÊNDICE A – Questionário aplicado

Questionário aplicado na pesquisa para levantamento do uso de ferramentas no desenvolvimento de ontologias.

# Study on tools used for ontology development

Developing ontologies is a complex process and the use of tools can help the ontology engineer in the various activities that must be carried out.

This study aims to investigate tools used in the development of ontologies and some issues related to the use of these tools, such as the characteristics considered relevant. The study is related to master's and doctoral research carried out at the Ontology & Conceptual Modeling Research Group (NEMO), of the Postgraduate Program in Computer Science (PPGI) at the Federal University of Espírito Santo (UFES), Brazil.

The questionnaire used in this study consists of eight questions to characterize the participant's profile and fourteen questions related to tools to support ontology development.

The time to answer the questions is around 15 minutes.

Anonymity is guaranteed and the data will be used exclusively for academic purposes.

Thank you in advance for your cooperation.

Sincerely,

Clarete Aparecida Diniz Gomes (clarete.gomes@edu.ufes.br)

Isauflânia Suelen Ribeiro Timóteo (isauflania.timoteo@edu.ufes.br)

Jordana Sarmenghi Salamon (jssalamon@inf.ufes.br)

Glaice Kelly da Silva Quirino Monfardini (glaice.monfardini@ifes.edu.br)

Rosiane da Silva Sangali (rosiane.sangali@edu.ufes.br)

Monalessa Perini Barcellos (monalessa@inf.ufes.br)

Vítor E. Silva Souza (vitor.souza@ufes.br)

---

\* Indicates required question

1. **E-mail (Your e-mail is captured for management purposes only. We will not share it and your responses will be kept anonymous):** \*
- 

**Participant Profile:**

2. 1. What is your highest level of education? \*

Select only one option.

*Mark only one oval.*

- Incomplete Bachelor's degree
- Complete Bachelor's degree
- Incomplete Post Graduate Specialization
- Complete Post Graduate Specialization
- Incomplete Master's degree
- Complete Master's degree
- Incomplete Doctorate
- Complete Doctorate

3. 2. In which area is your highest level of training? \*

Select only one option.

*Mark only one oval.*

- Computer Science
- Computer Engineering
- Information Systems
- Electrical Engineering
- Information Science
- Other: \_\_\_\_\_

4. 3. What is your level of knowledge about ontologies? \*

Select only one option.

*Mark only one oval.*

- Superficial (you did not take any course or class on ontologies and obtained some knowledge through books, papers, videos or other materials).
- Low (you have taken a course or class on ontologies lasting at least 4 hours or have worked on the subject during your undergraduate degree).
- Average (you have taken more than one course or class on ontologies or have worked on the subject during your master's or doctoral research).
- High (you are an expert on the subject or have worked on multiple projects involving ontologies).

5. 4. In what context do you work with ontologies? \*

Check all that apply.

*Check all that apply.*

- Academia, as a student
- Academia, as a professor/researcher
- Government
- Industry

6. 5. How long have you been working with ontologies? \*

*Mark only one oval.*

- Less than 1 year
- 1 to 3 years
- 3 to 5 years
- 5 to 10 years
- More than 10 years

7. 6. What type(s) of ontology do you usually develop? \*

Select only one option.

*Mark only one oval.*

- Operational ontologies (e.g., OWL/RDF)
- Reference ontologies (conceptual models)
- Both

8. 7. Do you use a systematic approach (e.g., a method) to develop ontologies? \*

*Mark only one oval.*

- Always
- Most of the time
- Rarely
- Never

9. 8. If you didn't answer "Never" in the previous question, please indicate which approach(es) you use:

---

*Skip to question 10*

### **Tools to support ontology development**

10. 9. How often do you use tools to support ontology development? \*

*Mark only one oval.*

- Always
- Most of the time
- Rarely
- Never

11. 10. Which tool(s) do you use to support ontology development? \*

Check all that apply.

*Check all that apply.*

- Astah
- AVOnEd
- CQchequer
- Eddy
- Growl
- Hozo
- Medina
- Menthor
- Note-taking software (e.g., Microsoft OneNote, Google Keep)
- Robot
- ROO
- OLED
- OntoEdit
- OntoTrack
- Ontouml-vp-plugin
- Onto4AllEditor
- OWLAX
- OWLGrEd
- Protegé
- Presentation software (e.g., Microsoft PowerPoint, Google Slides)
- Spreadsheet software (e.g., Microsoft Excel, Google Sheets)
- Visual Paradigm
- WebODE
- Word processing software (e.g., Microsoft Word, Google Docs)
- Other: \_\_\_\_\_

12. 11. In which phases of the ontology development process do you use GENERIC \* support tools (e.g., word processing, note-taking, spreadsheet, etc.)?

Check all that apply.

*Check all that apply.*

- Ontology Requirements Specification
- Ontology Conceptualization and Formalization
- Ontology Design
- Ontology Implementation
- Ontology Verification, Validation & Testing

13. 12. In which phases of the ontology development process do you use SPECIFIC \* support tools (e.g., ontology editor, UML editor, ontology verification & validation tool, etc.)?

Check all that apply.

*Check all that apply.*

- Ontology Requirements Specification
- Ontology Conceptualization and Formalization
- Ontology Design
- Ontology Implementation
- Ontology Verification, Validation & Testing

14. 13. What is the availability of the tool(s) you use? \*

*Mark only one oval.*

- They are all free to use.
- None of them are free to use.
- Some of them are free to use.

15. 14. How did you discover the tools you use? \*

Check all that apply.

*Check all that apply.*

- Referral by someone else (friends, work/study colleagues, professors, etc.)
- Developed by the organization (e.g., company, research group) in which you work.
- Used in the organization (e.g., company, research group) in which you work, but not developed by it.
- Indicated by the systematic approach (e.g. method, framework) for ontology development that you adopt
- Found by searching the Internet.
- Suggested in a course or class you took.
- Other: \_\_\_\_\_

16. 15. What features are present in the tools you use? \*

Check all that apply.

*Check all that apply.*

- Creation of diagrams to represent the conceptual model of the ontologies (e.g., creation of class diagrams to represent the conceptual models of the ontology).
- Representation of the ontology's concept hierarchy (taxonomy).
- Visualization of ontologies (e.g., presentation of the graph with the concepts and relations of an ontology, navigation in the conceptual model of the ontology).
- Ontology configuration management (i.e., version control, change control, etc.).
- Integration with a repository for storing ontologies in the cloud.
- Automatic generation of code (operational ontology) from conceptual models (reference ontology) (e.g., generation of OWL code from conceptual models represented in graphical language, e.g., OntoUML).
- Coding/implementation (not automatically) of operational ontology (e.g., in OWL, RDF).
- Generation of a conceptual model from the operational ontology (e.g., automatic representation of a diagram with the ontology's concepts and relationships from the ontology's OWL code).
- Verification of the ontology (e.g., assessing whether the ontology is represented correctly, whether the concepts and relationships in the ontology are sufficient and necessary to answer the defined competency questions).
- Validation of the ontology (e.g., instantiation of the ontology to assess whether it is capable of instantiating real-world situations).
- Preparation of ontology documentation (automatic or not).
- Integration with other tools.
- Definition of competency questions.
- Management of competency questions (e.g., change control, traceability between competency questions and ontology).
- Collaborative ontology development (e.g., several people can develop an ontology simultaneously).
- Automated testing (e.g., testing the operational ontology code, automatic instantiation and validation of axioms).
- Reuse of ontologies (e.g., selection of ontologies or ontology fragments for reuse in the development of a new ontology).
- Other: \_\_\_\_\_

17. 16. Do the tools you use provide integrated support for the entire ontology engineering process? \*

*Mark only one oval.*

- Yes (the tools I use are integrated and provide support from the conception of the ontology (requirements identification) to making it available for use).
- Partially (some of the tools I use are integrated and provide support for some of the phases of the ontology engineering process).
- No (the tools I use are not integrated and therefore provide fragmented support for phases of the ontology engineering process).

18. 17. What is your perception of the support provided by the use of tools in the development of ontologies? \*

*Mark only one oval.*

- It doesn't help at all.
- It helps very little.
- It helps a little.
- It helps a lot.

19. 18. Please justify: \*

---

---

---

---

---

19. What difficulties do you see in using tools to support the development of ontologies?

Check all that apply.

*Check all that apply.*

- Difficulties in identifying existing tools and selecting the most appropriate ones for my context.
- Difficulties accessing online tools or installing tools locally.
- It takes a long time to learn how to use the tools.
- Language limitations (e.g., the tools can only be found in a language on which I'm not proficient).
- Scope limitations (e.g., the tools support a limited set of activities in the ontology development process or the functionalities provided have limitations).
- The tools lack consistency between different versions (e.g., results produced in one version cannot be used in an earlier or later version).
- Integration problems (e.g., the tools do not allow integration with others, the result produced in one tool cannot be exported to another).
- The tools are difficult to use.
- The tools have many bugs.
- The cost of acquiring the tools is high.
- Lack of documentation on how to use the tools.
- Lack of support for questions about the tools (e.g., there is no communication channel with those responsible for the tool to ask questions about its use).
- Other: \_\_\_\_\_

21. 20. What benefits do you see in using tools to support ontology development? \*

Check all that apply.

*Check all that apply.*

- Improved quality of the produced ontologies.
- Accelerated ontology development process.
- Easier generation of ontology documentation.
- Better management of ontology development work.
- Better management of ontologies (e.g., control of different versions, authors, identification of problems).
- Possibility of collaboration in the development of ontologies (i.e., development of the same ontology by several people).
- Easier evolution of the ontologies developed.
- Improved communication between members of the development team.
- Easier reuse of existing ontologies.
- Other: \_\_\_\_\_

**Relevant properties in ontology development support tools**

22. 21. For each of the properties identified below, indicate your perception of the level of relevance of their presence in ontology development support tools. \*

- **Functional Suitability:** The tool's ability to do what it sets out to do.
- **Efficiency:** Performance of the tool in relation to the amount of resources used
- **Usability:** The tool's ability to achieve its objectives satisfactorily from the user's point of view.
- **Reliability:** The tool's ability to produce correct results.
- **Security:** The tool's ability to protect data and information.
- **Maintainability:** Ability to modify the tool.
- **Portability:** Ability for the tool to be used in different environments (e.g. different operating systems)
- **Availability of use:** Ability of the tool to be available for use (e.g. free access, trial version)
- **Collaboration:** The tool's ability to support the collaborative work of several users.
- **Scope:** The tool's ability to cover the application domain.

Mark only one oval per row.

	Very irrelevant	Irrelevant	Neutral	Relevant	Very relevant
<b>Functional Suitability</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Efficiency</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Usability</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Reliability</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Security</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Maintainability</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Portability</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Availability of use</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Collaboration</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Scope</b>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

23. 22. What other properties/characteristics do you think are important in a tool to support ontology development?

---

---

---

---

---

24. 23. If you have any additional comments on the subject of support tools for ontology development or any suggestions about the questionnaire, please use the space below to register them.

---

---

---

---

---

---

This content is neither created nor endorsed by Google.

Google Forms



Universidade Federal do Espírito Santo  
Pró-Reitoria de Pesquisa e Pós-Graduação  
Centro Tecnológico  
Programa de Pós-Graduação em Informática (PPGi)



## DECLARAÇÃO DE VERSÃO FINAL

Eu, então discente do Programa de Pós-Graduação em Informática (PPGi) da Universidade Federal do Espírito Santo (UFES) e autor desta dissertação, e o meu orientador, declaramos que a versão aqui apresentada incorpora as correções exigidas pelos membros examinadores da banca no dia da defesa.

Assinatura do autor

Assinatura do orientador