



UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS  
DEPARTAMENTO DE ELETRÔNICA E SISTEMAS  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

AÊDA MONALLIZA CUNHA DE SOUSA

**ONTOLOGIA PARA ESPECIFICAÇÃO DE REQUISITOS EM SISTEMAS  
EMBARCADOS**

Recife

2021

AÊDA MONALLIZA CUNHA DE SOUSA

**ONTOLOGIA PARA ESPECIFICAÇÃO DE REQUISITOS EM SISTEMAS  
EMBARCADOS**

Tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutora em Engenharia Elétrica.

Área de concentração: Eletrônica.

Orientadora: Profa. Dra. Fernanda Maria Ribeiro de Alencar.

Coorientador: Prof. Dr. Cleyton Mário de Oliveira Rodrigues.

Recife

2021

Catálogo na fonte  
Bibliotecária Margareth Malta, CRB-4 / 1198

S725o Sousa, Aêda Monalliza Cunha de.  
Ontologia para especificação de requisitos em sistemas embarcados /  
Aêda Monalliza Cunha de Sousa. - 2021.  
167 folhas, il., gráfs., tabs.

Orientadora: Profa. Dra. Fernanda Maria Ribeiro de Alencar.  
Coorientador: Prof. Dr. Cleyton Mário de Oliveira Rodrigues.  
Tese (Doutorado) – Universidade Federal de Pernambuco. CTG.  
Programa de Pós-Graduação em Engenharia Elétrica, 2021.  
Inclui Referências e Apêndices.

1. Engenharia Elétrica. 2. Ontologia. 3. Engenharia de requisitos.  
4. Sistemas embarcados. I. Alencar, Fernanda Maria Ribeiro de  
(Orientadora). II. Rodrigues, Cleyton Mário de Oliveira (Coorientador). III.  
Título

UFPE

621.3 CDD (22. ed.)

BCTG/2021-217

AÊDA MONALLIZA CUNHA DE SOUSA

**ONTOLOGIA PARA ESPECIFICAÇÃO DE REQUISITOS EM SISTEMAS  
EMBARCADOS**

Tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, como requisito parcial para a obtenção do título de Doutora em Engenharia Elétrica. Área de concentração: Eletrônica.

Aprovada em: 29/ 04/ 2021.

**BANCA EXAMINADORA**

---

Prof<sup>ª</sup>. Dr<sup>ª</sup>. Fernanda Maria Ribeiro de Alencar (Orientadora)  
Universidade Federal de Pernambuco

---

Prof. Dr. Cleyton Mário de Oliveira Rodrigues (Coorientador)  
Universidade de Pernambuco

---

Prof. Dr. Giancarlo Guizzardi (Examinador Externo)  
University of Twente

---

Prof. Dr. Johnny Cardoso Marques (Examinador Externo)  
Instituto Tecnológico de Aeronáutica

---

Prof<sup>ª</sup>. Dr<sup>ª</sup>. Marcia Jacyntha Nunes Rodrigues Lucena (Examinadora Externa)  
Universidade Federal do Rio Grande do Norte

---

Prof<sup>ª</sup>. Dr<sup>ª</sup>. Natasha Correia Queiroz Lino (Examinadora Externa)  
Universidade Federal da Paraíba

Dedico este trabalho primeiramente a Deus, por ser essencial em minha vida,  
em segundo aos meus pais e meu irmão.

## AGRADECIMENTOS

Agradeço primeiramente a Deus, por ser meu guia em tudo que faço e para onde vou. Obrigada por me conceder mais uma graça alcançada e pelas oportunidades concedidas durante a realização deste trabalho, as quais me permitiram chegar até aqui.

Agradeço aos meus pais, Antônio e Margareth, por todo o amor, carinho e conselhos dados durante toda a minha vida. Obrigada por sempre me proporcionar totais condições para a realização dos meus estudos. Que Deus continue os abençoando. Amo muito vocês.

Ao meu irmão, Matias Aidan, por ser uma pessoa maravilhosa e inteligente, por sua dedicação à família e pelo grande irmão que é. Obrigada por sempre me motivar e acreditar em mim. Te amo muito, meu galeguinho!

As minhas amigas, que no momento mais difícil que já enfrentei me deram todo apoio e sempre me motivaram para finalização deste trabalho. Obrigada por compartilharem suas experiências, ideias e conhecimentos. Fica aqui meu muito obrigada à Amora Sukar, Kadna Vale, Manuella Figueiras e Victória Nascimento.

Nessa jornada como docente tenho o prazer de conhecer alunos(as) maravilhosos(as). Agradeço aos meus queridos alunos(as) por todo carinho e confiança. Em especial quero agradecer a duas alunas/amigas, Letícia Lima e Nathiele Lins, pela força que convenceram no meu pensamento para não desistir e o conforto de saber que nunca estava só. Vocês me incentivaram a superar todas as dificuldades. Amo vocês!

Um agradecimento especial a minha orientadora, Fernanda Alencar, por toda paciência e contribuições durante a elaboração deste trabalho. Sou eternamente grata por tê-la como orientadora.

Durante a realização deste trabalho tive o prazer de conhecer Cleyton Rodrigues, ele foi essencial para realização desta pesquisa. Obrigada por ter aceitado ser meu co-orientador e por todo apoio e correções realizadas.

Um agradecimento aos professores, Giancarlo Guizzardi, Johnny Marques, Marcia Lucena e Natasha Lino, que aceitaram o convite para fazerem parte da banca de avaliação deste trabalho.

## RESUMO

Devido à particularidade existente nos diversos domínios dos Sistemas Embarcados, muitas vezes torna-se difícil entender os requisitos que devem ser identificados e traduzi-los nas funções e limitações esperadas do sistema. A complexidade inerente de diferentes domínios, bem como a necessidade de encontrar defeitos ou falhas com rapidez e precisão, são fatores importantes a serem considerados ao desenvolver esses sistemas. Apesar dos avanços em desenvolvimento de sistemas embarcados, ainda é uma área que merece atenção dos pesquisadores devido a sua complexidade e às poucas iniciativas de engenharia de requisitos que atendem às particularidades desses sistemas. A cada dia algumas abordagens para o desenvolvimento de Sistema Embarcados são propostas, considerando várias áreas do conhecimento, tais como ontologia. Gerenciar conhecimento na área de sistemas embarcados é um desafio para engenheiros e pesquisadores, pois a representação desses tipos de conhecimento é fonte de diversos problemas. O principal problema está relacionado à linguagem natural usada para elicitación e especificación de requisitos, e essas linguagens não são totalmente formalizadas. Com o advento da Web Semântica e das tecnologias para sua realização, as possibilidades de aplicação de ontologias como meio de definir a semântica da informação e do conhecimento tornam-se cada vez mais aceitas em diferentes domínios. Nesse contexto, esta tese visa, a partir dos padrões abertos da Web Semântica (o formalismo da lógica descritiva), definir uma abordagem ontológica para auxiliar a elicitación e especificación de requisitos em Sistemas Embarcados. Assim, através das Ontologias construídas, destacamos simulações de especificações de requisitos em cenários reais, extraindo os requisitos utilizados, normas e os impactos positivos e negativos que eles podem causar entre si. O método de pesquisa utilizado foi um mapeamento sistemático da literatura, uma revisão sistemática da literatura, survey e entrevistas com especialistas do domínio. Como contribuições desta tese, assinalamos o avanço no uso de ontologias logicamente formalizadas para a representação não-ambígua e compartilhada do conhecimento de requisitos em Sistemas Embarcados. Espera-se com isso auxiliar os desenvolvedores de sistemas embarcados, criando-se um núcleo comum de requisitos. É essencial impulsionar a condução das pesquisas na área da engenharia de requisitos, forma a ter-se qualidade e precisão na especificación no desenvolvimento de sistemas embarcados.

Palavras-chave: ontologia; engenharia de requisitos; sistemas embarcados.

## ABSTRACT

Due to the particularity existing in the various domains of Embedded Systems, it is often difficult to understand the requirements that must be identified and translate them into the system's expected functions and limitations. The inherent complexity of different domains and the need to find defects or flaws quickly and accurately are essential factors for developing these systems. Despite advances in the development of embedded systems, it is still an area that deserves attention from researchers due to its complexity and the few requirements engineering initiatives that meet these systems' particularities. Every day some approaches to the development of Embedded Systems are proposed, considering several knowledge areas, such as ontology. Managing knowledge in the area of embedded systems is a challenge for engineers and researchers, as the representation of these types of knowledge is the source of several problems. The main problem is related to the natural language used for eliciting and specifying requirements, and these languages are not fully formalized. With the advent of the Semantic Web and technologies for its realization, the possibilities of applying ontologies as a means of defining the semantics of information and knowledge become increasingly accepted in different domains. In this context, this thesis aims, based on the Semantic Web's open standards (the formalism of descriptive logic), to define an ontological approach to assist the elicitation and specification of requirements in Embedded Systems. Thus, through the built Ontologies, we highlight simulations of requirements specifications in real scenarios, extracting the requirements used, standards, and the positive and negative impacts that they can cause among themselves. The research method used was a systematic mapping of the literature, a systematic review of the literature, a survey, and interviews with experts in the field. As contributions to this thesis, we note the progress in using logically formalized ontologies for unambiguous and shared representation of requirements knowledge in Embedded Systems. This is expected to assist developers of embedded systems, creating a common core of requirements. It is essential to boost the conduct of research in requirements engineering to have quality and precision in the specification in the development of embedded systems.

**Keywords:** ontology; requirements engineering; embedded systems.

## LISTA DE FIGURAS

Figura 1 – Processo de engenharia de requisitos.....	27
Figura 2 – Estrutura de um Sistema Embarcado. ....	38
Figura 3 – Arquitetura da Web Semântica. ....	45
Figura 4 – Hierarquia das Ontologias.....	47
Figura 5 – Visão Geral da Methontology. ....	49
Figura 6 – Sintaxe X Semântica da <i>SR01Q</i> .....	54
Figura 7 – Fragmentos da UFO-A.....	56
Figura 8 – Fragmentos da UFO-B. ....	57
Figura 9 – Fragmentos da UFO-C. ....	58
Figura 10 – Processo de condução da revisão sistemática. ....	61
Figura 11 – Etapas para seleção dos trabalhos. ....	65
Figura 12 – Etapas do Processo de Wolf.....	72
Figura 13 –Etapas do Survey.....	83
Figura 14 – Características e Identificação dos Respondentes.....	86
Figura 15 – Definição dos Requisitos.....	87
Figura 16 – Utilização de Abordagem.....	88
Figura 17 – Abordagens para Elicitação.....	89
Figura 18– Inconsistências, Ausências e Integridade dos Requisitos. ....	90
Figura 19– Priorização de Requisitos.....	91
Figura 20– Documentação dos Requisitos. ....	92
Figura 21– Ferramentas e Gerenciamento de Versões. ....	93
Figura 22– Validação de Requisitos.....	94
Figura 23– Gerenciamento dos Requisitos.....	96
Figura 24– Prazo e Custo do Projeto. ....	96
Figura 25– Abordagens X Prazo e Custo. ....	97
Figura 26 – Nuvem de Palavras.....	103
Figura 27 – Ontologia com as três classificações.....	104
Figura 28 – Modelagem Agentes e Objetos da OntoREES.....	107
Figura 29 – Modelagem Requisitos da OntoREES. ....	115
Figura 30 – Condições de falha de sistemas pela RTCA DO-178C.....	118
Figura 31 – Modelagem da OntoREAV.....	121
Figura 32 – Modelo Empírico de Transferência de Tecnologia.....	128

Figura 33 – Padrão Role.....	132
Figura 34 – Padrão RoleMixin.....	132
Figura 35 – Padrão Category.....	133

## LISTA DE GRÁFICOS

Gráfico 1 – Quantidade de trabalhos relevantes por base dados. ....	69
Gráfico 2 – Porcentagem de trabalhos relevantes por base de dados.....	69
Gráfico 3 – Quantidade de estudos relevantes por ano. ....	70
Gráfico 4 – Porcentagem de estudos relevantes por ano. ....	71
Gráfico 5 – Tipos de Contribuição do Estudo. ....	75

## LISTA DE QUADROS

Quadro 1 – Resumo das escolhas metodológicas.....	21
Quadro 2 – Diferenças entre sistemas embarcados e sistemas de propósito geral.....	41
Quadro 3 - Expressões em <i>SR0IQ</i> .....	51
Quadro 4 – Construtos Booleanos de Conceitos <i>SR0IQ</i> .....	52
Quadro 5 – Construtores para restrições de relações <i>SR0IQ</i> .....	52
Quadro 6 – Perfil dos Tipos Universais da UFO.....	55
Quadro 7 – Fases, atividades e comparativo do processo de revisão sistemática e revisão da literatura comum.....	61
Quadro 8 – <i>String</i> de busca. ....	63
Quadro 9 – Bases de dados utilizadas. ....	64
Quadro 10 – Critérios de Qualidade. ....	66
Quadro 11 – Trabalhos selecionados.....	67
Quadro 12 – Etapas do processo de desenvolvimento abordados nos estudos. ....	73
Quadro 13 – Domínios dos Sistemas Embarcados apoiados pelos estudos. ....	77
Quadro 14 – Linguagens das Ontologias.....	78
Quadro 15 – Benefícios do uso de ontologia em SE.....	80
Quadro 16 – Axiomas de Agentes e Recursos. ....	108
Quadro 17 – Axiomas dos Requisitos. ....	112
Quadro 18 – Questões da OntoREAV.....	116
Quadro 19 – Axiomas dos Agente, Recursos e Requisitos da OntoREAV.....	119
Quadro 20 – Trabalhos Relacionados.....	123
Quadro 21 – Questões, Respostas e Justificativas do Estudo Empírico.....	129
Quadro 22 – ABox Instâncias Cenário 1. ....	134
Quadro 23 – Requisitos <i>Loadable Airborne System</i> .....	135
Quadro 24 – Inferências ABox 1.....	136
Quadro 25 – Inferências ABox 2.....	137
Quadro 26 – Inferências ABox 3.....	137
Quadro 27 – Inferências ABox 4.....	138
Quadro 28 – Inferências ABox 5.....	138

## LISTA DE ABREVIATURAS E SIGLAS

CAN	Controller Area Network
DL	Description Logic
ECU	Electronic Control Units
ER	Engenharia de Requisitos
MSL	Mapeamento Sistemático da Literatura
NFR	Non-Functional Requirements
OMG	Object Management Group
OntoREAV	Ontologia de Requisitos para Aviônicos
OntoREES	Ontologia de Requisitos para Sistemas Embarcados
OWL	Ontology Web Language
RSL	Revisão Sistemática da Literatura
SE	Sistema Embarcado
SPARQL	Semantic Query Language for Databases
SWRL	Semantic Web Rule Language
SysML	System Modeling Language
UFO	Unified Foundational Ontology
UML	Unified Modeling Language
W3C	World Wide Web Consortium
XML	eXtensible Markup Language

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>17</b>
1.1	CARACTERIZAÇÃO E JUSTIFICATIVA DO PROBLEMA.....	17
1.2	PROBLEMÁTICA DA PESQUISA .....	20
1.3	OBJETIVOS .....	20
1.4	PROCEDIMENTOS METODOLÓGICOS .....	21
<b>1.4.1</b>	<b>Quanto à Natureza da pesquisa.....</b>	<b>21</b>
<b>1.4.2</b>	<b>Quanto ao Procedimento de pesquisa.....</b>	<b>22</b>
<b>1.4.3</b>	<b>Quanto à Abordagem da pesquisa .....</b>	<b>22</b>
<b>1.4.4</b>	<b>Quanto ao Método de abordagem.....</b>	<b>23</b>
1.5	ESTRUTURA DO TRABALHO .....	23
<b>2</b>	<b>A ENGENHARIA DE REQUISITOS E SISTEMAS EMBARCADOS.....</b>	<b>25</b>
2.1	ENGENHARIA DE REQUISITOS .....	25
<b>2.1.1</b>	<b>Requisitos: Uma visão geral.....</b>	<b>25</b>
<b>2.1.2</b>	<b>Classificação dos requisitos.....</b>	<b>26</b>
<b>2.1.3</b>	<b>O processo de engenharia de requisitos.....</b>	<b>27</b>
<b>2.1.4</b>	<b>Elicitação de requisitos.....</b>	<b>28</b>
2.1.4.1	Análise de requisitos.....	29
2.1.4.2	Especificação de Requisitos .....	30
2.1.4.3	Validação dos requisitos .....	31
2.1.4.4	Gerenciamento de requisitos.....	32
2.2	SISTEMAS EMBARCADOS .....	33
<b>2.2.1</b>	<b>Conceitualização de sistemas embarcados .....</b>	<b>33</b>
<b>2.2.2</b>	<b>Histórico.....</b>	<b>34</b>
<b>2.2.3</b>	<b>Características e aplicações de sistemas embarcados.....</b>	<b>34</b>
<b>2.2.4</b>	<b>Famílias de sistemas embarcados.....</b>	<b>35</b>
<b>2.2.5</b>	<b>Estrutura Geral de um Sistema Embarcado.....</b>	<b>36</b>
2.2.5.1	Memórias para sistema embarcado.....	38
2.2.5.2	Processadores para sistemas embarcados .....	39
2.2.5.3	Microcontroladores.....	39
2.2.5.4	Outros componentes de um sistema embarcado.....	40
<b>2.2.6</b>	<b>Engenharia de requisitos para sistemas embarcados.....</b>	<b>40</b>
2.3	CONSIDERAÇÕES DO CAPÍTULO.....	42

<b>3</b>	<b>ONTOLOGIAS E A WEB SEMÂNTICA.....</b>	<b>43</b>
3.1	WEB SEMÂNTICA .....	43
3.2	ONTOLOGIA.....	46
<b>3.2.1</b>	<b>Tipologia das Ontologias.....</b>	<b>47</b>
<b>3.2.2</b>	<b>Componentes de uma Ontologia.....</b>	<b>48</b>
<b>3.2.3</b>	<b>Engenharia Ontológica: Methontology .....</b>	<b>48</b>
<b>3.2.4</b>	<b>Estratégias para Construção de Ontologias.....</b>	<b>50</b>
3.3	LÓGICA DESCRITIVA (DL) .....	50
<b>3.3.1</b>	<b>Sintaxe da Lógica Descritiva <i>SRQIQ</i> .....</b>	<b>51</b>
<b>3.3.2</b>	<b>Semântica da Lógica Descritiva <i>SRQIQ</i>.....</b>	<b>53</b>
3.4	UFO: A ONTOLOGIA DE FUNDAMENTAÇÃO UNIFICADA .....	54
<b>3.4.1</b>	<b>UFO-A: Uma Ontologia para Endurantes .....</b>	<b>56</b>
<b>3.4.2</b>	<b>UFO-B: Uma Ontologia para Perdurantes .....</b>	<b>56</b>
<b>3.4.3</b>	<b>UFO-C: Uma Ontologia para Entidades Sociais .....</b>	<b>57</b>
3.5	PADRÕES E ANTI-PADRÕES .....	58
3.6	SÍNTESE DO CAPÍTULO.....	59
<b>4</b>	<b>REVISÃO SISTEMÁTICA DA LITERATURA.....</b>	<b>60</b>
4.1	INTRODUÇÃO .....	60
4.2	FASE DE PLANEJAMENTO.....	62
<b>4.2.1</b>	<b>Questões de pesquisa .....</b>	<b>62</b>
<b>4.2.2</b>	<b>Estratégia de busca .....</b>	<b>63</b>
<b>4.2.3</b>	<b>Fontes de busca .....</b>	<b>64</b>
<b>4.2.4</b>	<b>Critérios de inclusão e exclusão dos estudos .....</b>	<b>64</b>
4.3	FASE DE EXECUÇÃO .....	65
4.4	AVALIAÇÃO DE QUALIDADE.....	66
4.5	FASE DE ANÁLISE E DIVULGAÇÃO DOS RESULTADOS .....	67
4.6	RESULTADOS E ANÁLISE.....	70
<b>4.6.1</b>	<b>Ano de Publicação .....</b>	<b>70</b>
<b>4.6.2</b>	<b>Q1 – Quais as etapas no processo de desenvolvimento dos sistemas embarcados foram apoiadas pelo uso de ontologias?.....</b>	<b>71</b>
<b>4.6.3</b>	<b>Q2 – Quais os tipos de contribuição do estudo? .....</b>	<b>74</b>
<b>4.6.4</b>	<b>Q3 – Quais os domínios de Sistemas Embarcados que o estudo suporta? .....</b>	<b>75</b>
<b>4.6.5</b>	<b>Q4 – Quais as linguagens de representação de ontologias foram utilizadas? .....</b>	<b>77</b>
<b>4.6.6</b>	<b>Q5 – Quais os requisitos (funcionais e não-funcionais) identificados? .....</b>	<b>79</b>

4.6.7	<b>Q6 – Quais os benefícios, identificados na literatura, relacionados ao uso de ontologia no desenvolvimento de sistemas embarcados? .....</b>	<b>79</b>
4.7	SÍNTESE DO CAPÍTULO .....	81
<b>5</b>	<b><i>SURVEY</i> .....</b>	<b>83</b>
5.1	ASPECTOS METODOLÓGICOS .....	83
5.2	ANÁLISE E SÍNTESE DOS DADOS .....	85
<b>5.2.1</b>	<b>Primeira I – Características dos Respondentes .....</b>	<b>85</b>
<b>5.2.2</b>	<b>Parte II – A Engenharia de Requisitos .....</b>	<b>86</b>
<b>5.2.3</b>	<b>Parte III – Requisitos de Hardware e Software .....</b>	<b>96</b>
5.3	AMEAÇAS À VALIDADE .....	97
5.4	SÍNTESE DO CAPÍTULO .....	99
<b>6</b>	<b>ONTOLOGIA DE REQUISITOS PARA SISTEMA EMBARCADO .....</b>	<b>100</b>
6.1	CONSTRUÇÃO DA ONTOLOGIA .....	100
<b>6.1.1</b>	<b>Especificação .....</b>	<b>101</b>
<b>6.1.2</b>	<b>Aquisição do Conhecimento.....</b>	<b>102</b>
<b>6.1.3</b>	<b>Conceitualização .....</b>	<b>103</b>
<b>6.1.4</b>	<b>Formalização .....</b>	<b>103</b>
6.1.4.1	Classificações dos Recursos e Agentes .....	104
6.1.4.2	Classificações dos Requisitos .....	109
6.2	ONTOLOGIA DE DOMÍNIO .....	116
<b>6.2.1</b>	<b>OntoREAV .....</b>	<b>116</b>
6.3	TRABALHOS RELACIONADOS .....	122
6.4	SÍNTESE DO CAPÍTULO .....	125
<b>7</b>	<b>VALIDAÇÃO DAS ONTOLOGIAS .....</b>	<b>126</b>
7.1	VALIDAÇÃO EMPÍRICA .....	126
7.2	VERIFICAÇÃO – PADRÕES .....	131
7.3	VALIDAÇÃO – UTILIZANDO INFERÊNCIAS ABOX.....	133
<b>7.3.1</b>	<b>Inferências Realizadas.....</b>	<b>133</b>
7.4	SÍNTESE DO CAPÍTULO .....	139
<b>8</b>	<b>CONSIDERAÇÕES FINAIS.....</b>	<b>140</b>
8.1	CONCLUSÕES .....	140
8.2	CONTRIBUIÇÕES .....	141
8.3	LIMITAÇÕES E AMEAÇAS À VALIDADE.....	142
8.4	TRABALHOS FUTUROS .....	142

8.5	PUBLICAÇÕES .....	143
	<b>REFERÊNCIAS.....</b>	<b>145</b>
	<b>APÊNDICE A – PROTOCOLO DE PESQUISA DA REVISÃO</b>	
	<b>SISTEMÁTICA .....</b>	<b>154</b>
	<b>APÊNDICE B – E-MAIL.....</b>	<b>156</b>
	<b>APÊNDICE C – CARTA CONVITE.....</b>	<b>157</b>
	<b>APÊNDICE D – QUESTIONÁRIO .....</b>	<b>158</b>
	<b>APÊNDICE E – GLOSSÁRIO DE TERMOS.....</b>	<b>161</b>

# 1 INTRODUÇÃO

Este capítulo apresenta uma visão geral da tese. Apresentando o contexto da pesquisa, o direcionamento para o problema de pesquisa, a motivação e a justificativa desta proposta de tese, o objetivo geral do trabalho e os objetivos específicos, métodos e procedimentos utilizados para realização desta proposta.

## 1.1 CARACTERIZAÇÃO E JUSTIFICATIVA DO PROBLEMA

Estamos em um cenário no qual, cada vez mais, a tecnologia é necessária para facilitar o cotidiano das pessoas. Por isso, os Sistemas Embarcados (SE) estão se tornando presentes em diversas atividades do dia a dia, trazendo alguns benefícios como conforto e segurança. Atualmente, é muito raro encontrar qualquer dispositivo eletrônico que não seja controlado por um sistema embarcado. Esses sistemas geralmente recebem dados de entrada de sensores, usam um microcontrolador ou microprocessador para processar os dados e enviam os resultados para o atuador e / ou interface do usuário (SAEED et al., 2019).

Grande parte dos fracassos nos projetos de desenvolvimento de sistemas, sejam totais ou parciais, são decorrentes do desconhecimento das boas práticas da engenharia de sistemas (OSHANA; KRAELING, 2019). A Engenharia de Requisitos (ER) tem sido reconhecida como uma etapa importante no processo de engenharia de sistemas. A ER é uma etapa do desenvolvimento de sistemas responsável pela identificação dos objetivos do sistema pretendido, pela operacionalização de tais objetivos em serviços e restrições e pela atribuição da responsabilidade dos requisitos resultantes para agentes, humanos, hardware e software (POHL, 2016). O sucesso do processo de desenvolvimento ou manutenção de um sistema está intrinsecamente associado ao uso das boas práticas sugeridas pela engenharia de requisitos.

No desenvolvimento de sistemas, de forma geral, a elicitação de requisitos é fundamental para que o sistema proposto funcione de acordo com as necessidades dos *stakeholders* e restrições às quais se encontra submetido. Assim, a má especificação ou a ausência de documentação suficiente podem ser fatores de insucesso dos sistemas. Segundo Pereira et al. (2018), no projeto de sistemas embarcados, devido à simplicidade ou necessidade de agilidade no processo de desenvolvimento, a engenharia de requisitos é frequentemente ignorada porque o tempo de lançamento no mercado (time to market) é menor do que o de sistemas gerais.

Os projetos de SE têm aumentado significativamente de tamanho, acompanhando a evolução da complexidade dos componentes eletrônicos que cada vez mais necessitam de confiabilidade, bom desempenho e segurança adequada. Consequentemente, esses aspectos geram novos desafios para a engenharia de requisitos. Engenheiros da área de sistemas embarcados têm buscado abordagens da engenharia de requisitos, na tentativa de adaptar e aproveitar o que já existe, e buscar novas técnicas que levem em consideração as particularidades dos SE (OSSADA et al., 2012). Devido às especificidades desses conceitos e suas relações, muitas vezes se torna complexa a compreensão das necessidades que devem ser identificadas e traduzidas em funcionalidades e restrições do sistema pretendido. A complexidade inerente aos diferentes domínios, aliada à necessidade de encontrar defeitos ou faltas de forma rápida e precisa são fatores importantes a serem considerados no desenvolvimento desses sistemas (PEREIRA et al., 2018).

A compreensão efetiva do domínio e das necessidades dos *stakeholders* é fundamental para que se especifique de forma clara, inequívoca e o mais rigorosa possível, em termos de formalidade, as funcionalidades e não funcionalidades do sistema pretendido. Diante disso, a má especificação podem ser fatores de insucesso dos sistemas em desenvolvimento (STARON, 2019). A garantia da qualidade e a precisão na definição dos requisitos são essenciais para o correto e bom funcionamento dos SEs, bem como para a sua evolução. Isso tem levado a comunidade acadêmica a explorar e estabelecer novas abordagens para o desenvolvimento de SE e integrá-las a diferentes áreas do conhecimento, tais como ontologia.

Pesquisas nos últimos anos mostraram uma fusão entre as áreas de engenharia de requisitos e tecnologias semânticas (DERMEVAL et al., 2016). Em 2001 foram lançadas as bases da Web Semântica. Ela busca facilitar o processo de comunicação entre os diversos participantes de um domínio, de forma a criar um modelo mental comum, minimizando a possibilidade de ambiguidades, e facilitando, assim, o trabalho necessário para o desenvolvimento de aplicações que manipulem as diversas fontes de dados (GRUBER, 2008). As ontologias fazem parte de uma camada da Web Semântica. Segundo Pease (2011), ontologia é um componente importante em diversas áreas, fornecendo uniformidade aos conceitos, no que diz respeito à sintaxe e semântica, facilitando a comunicação em diversos domínios. Com o lançamento do conceito semântico e o avanço das tecnologias semânticas, as oportunidades de aplicação de ontologias como meio de definir a semântica da informação e do conhecimento têm se tornado cada vez mais aceitas em diferentes domínios. Ao mesmo tempo, a implementação da maioria das técnicas de classificação de requisitos não trata dos aspectos semânticos dos requisitos. Se o significado dos requisitos e suas relações podem ser

manipulados, os desenvolvedores de sistemas podem obter classificações de requisitos mais eficazes para produzir especificações de requisitos de qualidade superior.

Administrar o conhecimento no domínio dos sistemas embarcados é um desafio para engenheiros e pesquisadores, pois a representação destes tipos de conhecimento é fonte de vários problemas. Os principais problemas estão relacionados a linguagem natural utilizada para elicitación e especificação dos requisitos, e essas linguagens não são semanticamente formalizadas em sua completude. No desenvolvimento de SE, para uma comunicação ser bem-sucedida, os *stakeholders* envolvidos devem compartilhar um mesmo conjunto de requisitos para que não haja diferentes interpretações durante o processo de elicitación e especificação. Verifica-se, então, que a linguagem natural é passível de diversas formas de ambiguidades. O conteúdo de uma ontologia é geralmente descrito pela utilização de formalismos para representação de conhecimento bem embasado em termos lógico-matemáticos, que podem dotar os sistemas com a habilidade de realizar inferências lógicas (RODRIGUES, 2019). Diante deste contexto, sugere-se que uma ontologia direcionada para o domínio de sistemas embarcados pode contribuir para o sucesso do desenvolvimento de tais sistemas, pois apresenta uma formalidade para representar os conceitos internos do domínio e as relações entre eles. Essa formalidade diminui a ambiguidade e inconsistência entre os requisitos (SOUSA et al., 2016).

Diante disso, é construída uma ontologia central de requisitos para SEs, que poderá ser especializada para domínios específico de SEs, possibilitando uma melhor definição dos requisitos no desenvolvimento desses tipos de sistemas. Nesses sistemas a garantia da qualidade e precisão da definição dos requisitos é essencial para o funcionamento correto dos mesmos e para a sua evolução. Assim, a elicitación e especificação feita de forma coerente e correta, fornece maior controle e qualidade ao projeto. Portanto, a seguinte hipótese de pesquisa foi definida: a construção de Ontologias para Especificação de Requisitos em Sistemas Embarcados através das tecnologias abertas da Web Semântica contribui para que seja formalizado o conhecimento inerente aos requisitos de sistemas embarcados. As ontologias propostas foram construídas utilizando um método sistemático baseado na Menthontology, utilizando lógica descritiva. Além disso, para construção das questões de competências e levantamento dos conceitos e relacionamentos das ontologias, foi realizada uma RSL, Survey e entrevistas com especialistas do domínio de embarcados.

As ontologias desenvolvidas foram avaliadas (verificação e validação) utilizando uma validação empírica, verificação de padrões e anti-padrões, e validação por inferências Abox. A ontologia proposta pode representar uma contribuição significativa para área de requisitos

em sistemas embarcados, pois pode garantir a consistência entre os requisitos e facilitar a comunicação entre os engenheiros, devido a padronização dos conceitos. Além disso, essa ontologia pode ser usada para aumentar a qualidade na especificação de requisitos de sistemas embarcados.

## 1.2 PROBLEMÁTICA DA PESQUISA

Nesta tese pretende-se investigar, como questão de pesquisa (QP), a que trata da seguinte problemática: “**Como formalizar o conhecimento inerente aos requisitos de sistemas embarcados e realizar inferências lógicas para simulação de definição de requisitos de um projeto de SE?**”. Para tanto, pretende-se utilizar os padrões abertos da Web Semântica (o formalismo Description Logic (DL)).

Com a intenção de estudar este problema, foram definidas as seguintes questões de pesquisa (QP):

**QP01-** Quais estudos estão disponíveis na literatura, que tratam sobre ontologias de requisitos em projetos de sistemas embarcados?

**QP02-** Como acontece a engenharia de requisitos em desenvolvimento de sistemas embarcados, e quais abordagens estão sendo utilizadas por especialistas do domínio?

**QP03-** Como é possível formalizar o conhecimento dos requisitos de sistemas embarcados, através de uma notação Ontológica que possa ser usada e reutilizada para simular a definição de requisitos, por intermédio da lógica descritiva?

**QP04-** Como disponibilizar os resultados alcançados nesta pesquisa, para que se possa interagir e realizar simulações de definição de requisitos em projetos de SE?

## 1.3 OBJETIVOS

O objetivo geral nesta tese é construir Ontologias para Especificação de Requisitos em Sistemas Embarcados através das tecnologias abertas da Web Semântica. Portanto, por meio da solução proposta, é possível realizar simulações para definições de requisitos, de SE, em forma de modelos lógicos, de forma a tornar explícitos os requisitos de um determinado sistema, além de checar a consistência desses.

Desdobrando-se o objetivo geral, tem-se questões mais delimitadas e complementares que se quer alcançar, estabelecendo-se estreita relação com a satisfação desse objetivo. Assim, em particular, são propostos cinco objetivos específicos:

- realizar uma revisão sistemática da literatura em busca de encontrar ontologias de requisitos específicas para o domínio de sistemas embarcados.;
- realizar um *survey* com especialista do domínio sobre a Engenharia de Requisitos em Sistemas Embarcados;
- construir a OntoREES, uma ontologia de propósito mais geral para mapear os principais requisitos dos Sistemas Embarcados;
- construir Ontologias de Domínio, isto é, ontologias mais específicas que a OntoREES para formalizar subdomínios de SE; e
- realizar avaliações para verificar a consistência, aplicabilidade e adequação das ontologias para temática proposta.

#### 1.4 PROCEDIMENTOS METODOLÓGICOS

A metodologia científica se propõe a definir regras e procedimentos que darão segurança e validade ao exercício de conhecer, tendo a pesquisa presente nesse processo (SEVERINO, 2017).

Para atender ao objetivo central desta pesquisa, serão utilizados diversos procedimentos e técnicas, apresentados no Quadro 1 e detalhados a seguir.

Quadro 1 – Resumo das escolhas metodológicas

<b>Natureza da pesquisa</b>	Básica e Aplicada
<b>Procedimentos utilizados na pesquisa</b>	Pesquisa bibliográfica e Survey
<b>Abordagem da pesquisa</b>	Quantitativa e Qualitativa
<b>Método de abordagem</b>	Hipotético-dedutivo

Fonte: A Autora (2021).

##### 1.4.1 Quanto à Natureza da pesquisa

As pesquisas científicas podem ser classificadas conforme a perspectiva a que se propõe alcançar, podendo assim ser básica ou aplicada. Em particular, esta pesquisa é de natureza básica e aplicada ao mesmo tempo. É de natureza básica, por buscar conhecimentos que servem de apoio à formulação da hipótese em coerência com o problema de pesquisa estudado e que, de alguma forma, geram novos conhecimentos úteis ao avanço da ciência

(PROVDANOV; FREITAS, 2013). Por outro lado, é de natureza aplicada porque objetiva gerar conhecimentos para aplicação prática dirigidos à solução de problemas específicos.

#### 1.4.2 Quanto ao Procedimento de pesquisa

A pesquisa bibliográfica disponibiliza um resumo das evidências relacionadas a uma estratégia de intervenção específica, mediante a aplicação de métodos explícitos e sistematizados de busca, apreciação, crítica e síntese da informação selecionada (SEVERINO, 2017). Deste modo, esta pesquisa caracteriza-se por ser bibliográfica, pois procura explicar um problema a partir de fontes literárias publicadas em artigos, dissertações e teses advindos de repositórios e indexadores virtuais.

A revisão bibliográfica baseou-se em uma revisão sistemática da literatura (discutida no Capítulo 3). Para Kitchenham et al., (2009), a revisão sistemática, assim como outros tipos de estudo de revisão, é uma forma de pesquisa que utiliza como fonte de dados a literatura sobre determinado tema. Ao usar métodos de pesquisa explícitos e sistemáticos, avaliações-chave e métodos integrados de informações selecionadas, essas pesquisas fornecem um resumo das evidências relacionadas a estratégias de intervenção específicas.

O survey é um tipo de pesquisa que ocorre quando envolve a interrogação direta das pessoas cujo comportamento desejamos conhecer através de algum tipo de questionário (PROVDANOV; FREITAS, 2013). O survey proposto nesta tese, teve como objetivo entender o processo de Engenharia de Requisitos no desenvolvimento de Sistemas Embarcados por especialista na temática de pesquisa.

#### 1.4.3 Quanto à Abordagem da pesquisa

Quanto a forma de abordagem, essa pesquisa caracteriza-se como qualitativa e quantitativa. É considerada quantitativa, pois considera que tudo pode ser quantificável, o que significa traduzir em números opiniões e informações para classificá-las e analisá-las (PROVDANOV; FREITAS, 2013). Requer o uso de recursos e de técnicas estatísticas (percentagem, média, moda, mediana, desvio-padrão, coeficiente de correlação, análise de regressão etc.). Mas também é possível identificar características qualitativas, pois foca em aspectos mais profundos através da análise detalhada, classificação e interpretação do contexto do objeto de pesquisa (SEVERINO, 2017).

#### 1.4.4 Quanto ao Método de abordagem

Quanto ao método de abordagem, esta pesquisa caracteriza-se como hipotético-dedutiva. Segundo Provdanov e Freitas (2013), o método hipotético-dedutivo inicia-se com um problema ou uma lacuna no conhecimento científico, passando pela formulação de hipóteses e por um processo de inferência dedutiva, o qual testa a predição da ocorrência de fenômenos abrangidos pela referida hipótese.

Como hipótese tem-se que a construção de Ontologias para Especificação de Requisitos em Sistemas Embarcados através das tecnologias abertas da Web Semântica contribui para que seja formalizado o conhecimento inerente aos requisitos de sistemas embarcados. Promovendo-se, assim, a garantia da qualidade e a precisão na definição dos requisitos tão essenciais não só ao funcionamento correto dos mesmos, mas à sua evolução. Dessa forma, a elicitação e especificação feita de forma coerente e correta fornecerá maior controle e qualidade ao projeto.

#### 1.5 ESTRUTURA DO TRABALHO

Para melhor compreensão, este trabalho foi organizado em 7 capítulos, da seguinte maneira:

- **Capítulo 1 – Introdução:** é realizada uma apresentação contextualizada do tema de pesquisa, seus objetivos, problemática da pesquisa.
- **Capítulo 2 – A Engenharia de Requisitos e Sistemas Embarcados:** apresenta a base conceitual para o desenvolvimento do trabalho e relaciona as principais áreas dos estudos: Engenharia de Requisitos, Sistemas Embarcados e Engenharia de Requisitos em Sistemas Embarcados.
- **Capítulo 3 – Ontologias e a Web Semântica:** apresenta a base conceitual para o desenvolvimento do trabalho e relaciona as principais áreas dos estudos: Web Semântica, Lógica Descritiva e UFO.
- **Capítulo 4 – Revisão Sistemática da Literatura:** demonstra a aplicação do método de pesquisa bibliográfica proposto, apresentando todos os passos para identificação e seleção dos trabalhos utilizados, incluindo em detalhes o protocolo utilizado para a aplicação da revisão sistemática.
- **Capítulo 5 – Survey:** apresenta um survey sobre a engenharia de requisitos no

desenvolvimento de sistemas embarcados.

- **Capítulo 6 – Ontologias de Requisitos para Sistemas Embarcados (OntoREES e OntoREV):** apresenta o processo de elaboração das duas ontologias propostas nesta tese.
- **Capítulo 7 – Validação das Ontologias:** demonstrar as validações que foram realizadas.
- **Capítulo 8 – Considerações Finais:** apresenta as limitações, contribuições, trabalhos futuros e conclusões.

## 2 A ENGENHARIA DE REQUISITOS E SISTEMAS EMBARCADOS

Neste capítulo, apresentamos o conhecimento de base necessário à compreensão da temática proposta neste trabalho.

### 2.1 ENGENHARIA DE REQUISITOS

A Definição e Análise de Requisitos são reconhecidas como as etapas mais importantes em qualquer modelo de desenvolvimento de *software*, pois mesmo que se tenha um sistema bem projetado e codificado, a má especificação de requisitos ou a ausência de documentação suficiente, causará transtornos aos desenvolvedores e clientes. A especificação de requisitos é fundamental para que o sistema pretendido funcione de acordo com as necessidades dos *stakeholders* sem deixar de atender às restrições sob as quais se encontra submetido.

Nesta seção, apresentamos uma visão geral da engenharia de requisitos apontando algumas definições para os termos Requisitos e Engenharia de Requisitos, a classificação para os requisitos e o processo de engenharia de requisitos.

#### 2.1.1 Requisitos: Uma visão geral

De acordo com Sommerville (2011), um requisito pode ser uma função, uma restrição ou outra característica que deve ser fornecida ou atendida para satisfazer as necessidades do usuário do sistema que será desenvolvido. Os requisitos são necessidades coletadas de clientes e usuários para sabermos “o quê” o sistema a ser idealizado fará, sem se preocupar por enquanto, “como” este será criado.

Os requisitos são todos os requisitos no nível do sistema que descrevem as funções que o sistema como um todo deve cumprir para satisfazer as necessidades e requisitos das partes interessadas (WOLF, 2012).

No processo de desenvolvimento de software e sistema, definir requisitos que possam ser entendidos pelas partes interessadas (clientes, analistas de sistema, desenvolvedores, etc.), é um fator básico, mas também um problema difícil de resolver (POHL, 2016). Portanto, para assegurar que os requisitos dos sistemas sejam bem definidos, é necessário o auxílio de uma metodologia organizada e sistemática, a qual permita que os engenheiros de *software* utilizem

ferramentas, técnicas e abordagens apropriadas para resolver o problema. Nesse sentido é que foi criada a Engenharia de Requisitos.

Existem muitas definições da Engenharia de Requisitos. De acordo com Pohl (2016), a Engenharia de Requisitos corresponde ao processo sistêmico de desenvolvimento de requisitos, analisa os problemas por meio de um processo iterativo e colaborativo, registra os resultados e as observações em várias representações e verifica a precisão do entendimento, que corresponde ao desenvolvimento dos requisitos.

Segundo Sommerville (2011), a Engenharia de Requisitos é o processo de descobrir, analisar, documentar e verificar as funções e restrições do sistema. Ela é responsável pela identificação dos objetivos do sistema pretendido, pela operacionalização de tais objetivos em serviços e restrições e pela atribuição da responsabilidade dos requisitos resultantes para agentes como: humanos, *hardware*, e *software*.

O que observamos é que algumas definições procuram retratar a Engenharia de Requisitos como um processo sistemático de investigação das necessidades do cliente com relação ao sistema a ser desenvolvido, portanto, baseando-se fundamentalmente na percepção do problema para o qual se desenvolverá um sistema (*software* e/ou *hardware*). Ainda assim, recaem na questão da compreensão do problema por parte dos desenvolvedores.

### 2.1.2 Classificação dos requisitos

De forma geral podemos classificar os requisitos de três maneiras: requisitos funcionais, requisitos não funcionais, e de domínio (SOMMERVILLE, 2011). Nesta pesquisa, adotou-se os tipos: Requisitos Funcionais e Requisitos Não Funcionais porque são os principais requisitos considerados em projetos de sistemas embarcados.

Os **requisitos de domínio** são restrições originárias do domínio da aplicação do sistema e refletem características do mesmo.

Os **requisitos funcionais** estão ligados diretamente às funcionalidades do *software* e definem como o sistema deve reagir às entradas específicas de um sistema e como deve comportar-se em determinadas situações, ou seja, expressa o que o sistema deve fazer.

Os **requisitos não-funcionais** expressam as qualidades específicas que o *software* deve ter. Dizem respeito ao desempenho, segurança, confiabilidade, portabilidade entre outras propriedades que o sistema deve possuir. Em Sistemas Embarcados, os requisitos relacionados ao *hardware* também são classificados como requisitos não funcionais. Os

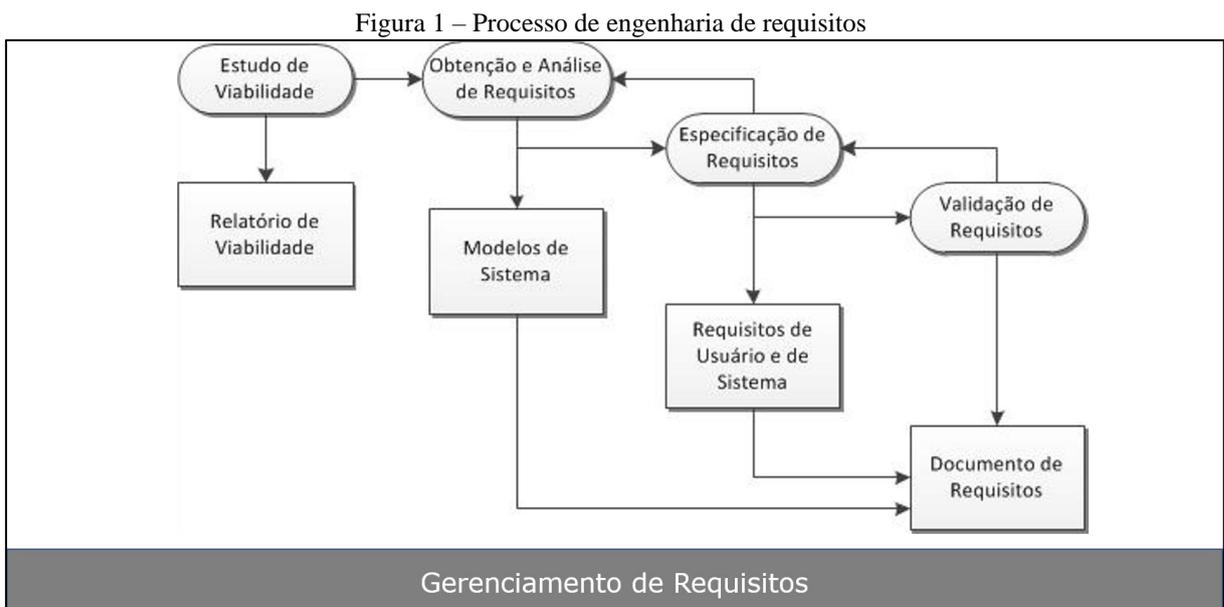
requisitos não funcionais são frequentemente mais críticos que os requisitos funcionais (WOLF, 2012) (SOMMERVILLE, 2011).

De forma geral, a diferença entre requisitos funcionais e não funcionais está no fato dos primeiros descreverem “o quê” o sistema deve fazer, enquanto que os outros fixam restrições sobre “como” os requisitos funcionais serão implementados (POHL, 2016). A ontologia desenvolvida nesta tese aborda os dois tipos de requisitos (funcionais e não funcionais).

Os requisitos funcionais e não funcionais, em princípio, devem ser distinguidos numa especificação de requisitos. Todavia, nem sempre é tão simples estabelecer a diferença entre o que seja um requisito funcional ou não funcional. Fazer uma separação bem definida fica um tanto quanto difícil, pois alguns requisitos podem exprimir necessidades do sistema como um todo e não limitações em funções individuais.

### 2.1.3 O processo de engenharia de requisitos

Para Sommerville (2011), o objetivo do processo de ER é definir as **funcionalidades** que o cliente deseja que o sistema acomode, e as **restrições** que afetam a operação e o desenvolvimento deste sistema. O processo de engenharia de requisitos está exemplificado na Figura 1.



Fonte: SOMMERVILLE (2011).

As entradas para o processo de engenharia de requisitos incluem: informações sobre sistemas existentes, necessidades dos *stakeholders*, padrões organizacionais, regulamentações, domínio da informação, entre outros. Todas essas entradas são utilizadas para realização das atividades do processo. Como saídas temos: relatório de atividades, modelos de sistema, requisitos de usuário e de sistema e documento de requisitos.

No processo de engenharia de requisitos, essas fases não seguem rigorosamente uma certa sequência de realização. Normalmente inicia-se com o estudo de viabilidade, depois a elicitación de requisitos, juntamente com a análise de requisitos e negociações. Paralelamente a todas essas atividades, é realizado o gerenciamento de requisitos, no qual objetiva-se gerenciar mudanças nos requisitos do sistema. Consecutivamente, os requisitos são documentados e validados.

#### 2.1.4 Elicitación de requisitos

A elicitación de requisitos é a primeira atividade no processo de engenharia de requisitos. Nesta atividade, os requisitos são extraídos dos clientes e usuários, determinando-se “o quê” o sistema deverá fazer. Pretende-se, nesta etapa, obter conhecimento, o mais correto e mais completo possível, sobre o domínio do problema, de forma a prover entendimento do que será demandado e possibilitar que esse conhecimento possa ser usado para produzir uma especificação mais precisa (POHL, 2016). A elicitación é umas das primeiras fases no ciclo de vida da engenharia de requisito, seu principal objetivo é obter informações relevantes para o desenvolvimento do sistema. Trata-se de uma tarefa complexa, desafiadora e propensa a erros devido a várias razões (SANTANDER, 2002):

- Os *stakeholders* frequentemente não sabem o que querem do sistema a não ser em termos gerais;
- Os *stakeholders* expressam os requisitos naturalmente em seus próprios termos e com o conhecimento implícito de seu trabalho;
- Diferentes *stakeholders* possuem diferentes requisitos, expressos em formatos diferentes;
- Fatores políticos podem influenciar os requisitos do sistema;
- O ambiente econômico e de negócio sobre o qual a análise é realizada é dinâmico.

Selecionar a técnica mais apropriada para elicitar os requisitos, portanto, não é uma tarefa trivial. As técnicas mais utilizadas são (SOMMERVILLE, 2011):

- **Sessões colaborativas (*Workshop*)** – são as sessões em que se reúnem os interessados, especialmente selecionados, junto a um engenheiro de requisitos, em uma sala na qual todos relatam os problemas e o que se deseja e espera do novo sistema;
- **Entrevista** – técnica comumente utilizada pelos especialistas, particularmente para levantar novas especificações, resolver conflitos ou interesses do grupo, podendo ser aplicada individualmente, quando um entrevistador contata o interessado para sanar suas dúvidas;
- **Etnografia** – considerado um dos métodos mais eficazes. É uma técnica de observação. O elicitor observa o ambiente de trabalho, anotando e acompanhando os processos da empresa;
- **Cenários** – focalizam as atividades que os envolvidos realizam nas organizações, possibilitando uma visão ampla dos problemas atuais, na qual o sistema será inserido;
- **Brainstorming** – é uma técnica para geração de ideias. Ela consiste em uma ou várias reuniões que permitem que as pessoas sugiram e explorem ideias.

A escolha da técnica a ser utilizada dependerá do problema e dos *stakeholders* envolvidos. Essas técnicas de elicitação devem descrever os requisitos de uma forma facilmente entendida por todos.

#### 2.1.4.1 Análise de requisitos

A análise de requisitos é o processo de análise das necessidades dos usuários e cliente do *software* para chegar a uma definição precisa dos requisitos. É uma etapa altamente crítica no processo de desenvolvimento, pois os requisitos devem ser analisados para detectar incompletudes, omissões ou redundâncias, deve ser intercalada com a elicitação, pois, em geral, problemas são descobertos quando os requisitos estão sendo elicitados (SANTANDER, 2002).

Nesta etapa, a preocupação está em descobrir os requisitos que realmente são necessários e que atende a necessidade do *stakeholder*. Algumas técnicas podem ser utilizadas para essa etapa. Entre elas destacam-se (SOMMERVILLE, 2011):

- **Lista de checagem de análise** – é uma lista de questão utilizada, as quais os analistas podem usar para avaliar cada requisito;
- **Matrizes de interação** – é utilizada para descobrir as interações entre os requisitos, apontando possíveis conflitos entre eles;
- **Prototipação** – avaliar as características de um *software* antes que ele venha realmente a ser construído. É uma versão inicial de um sistema para experimentação.

Depois de descobertos os problemas e conflitos entre os requisitos, ocorre o processo de negociação. Esta fase visa discutir e solucionar problemas advindos do conflito entre os diversos *stakeholders*, os quais podem atribuir diferentes prioridades aos requisitos (SANTANDER, 2002). Durante o desenvolvimento de um sistema, conflitos sempre surgirão e não podem ser negligenciados. Diante disso, durante todo o ciclo de desenvolvimento do sistema, deve-se recorrer a negociação com frequência.

#### 2.1.4.2 Especificação de Requisitos

Esta fase está relacionada a uma descrição detalhada dos requisitos do sistema, a qual deve ser facilmente entendida por todos os *stakeholders*. O objetivo dessa etapa é obter um documento formal contendo uma definição oficial do que o sistema deve fazer, o “Documento de Requisitos” e é intercalada muitas vezes com as atividades de elicitação, análise e negociação de requisitos (POHL, 2016). Esse documento assume um papel fundamental no processo de desenvolvimento, trazendo grandes benefícios a todos *stakeholders*, tais como (OSSADA, 2010):

- É o veículo básico de comunicação entre os desenvolvedores e os usuários sobre o que deve ser construído;
- Registra os resultados da análise do problema obtido através da elicitação e da análise dos requisitos;
- Define quais propriedades o sistema deve ter e quais são as restrições impostas em seu projeto e execução;
- É a base para as estimativas de custo e o cronograma;
- É o fundamento para o desenvolvimento do plano de teste do sistema;
- Oferece uma definição padrão do comportamento esperado pelos profissionais envolvidos na manutenção do sistema.

Idealmente, os requisitos declarados no documento devem ser claros, inequívocos, de fácil compreensão, completos e consistentes. Na prática, isso é difícil de conseguir, pois os *stakeholders* interpretam os requisitos de maneiras diferentes e, muitas vezes, notam-se conflitos e inconsistências inerentes aos requisitos (POHL, 2016). Esse documento não deve incluir informações detalhadas sobre a arquitetura do sistema, pois o objetivo do documento é o detalhamento dos requisitos funcionais e não funcionais.

Segundo Sommerville (2011), as principais técnicas utilizadas para especificação de requisitos são:

- Linguagem natural;
- Linguagem natural estruturada;
- Linguagem de descrição de projeto;
- Notações gráficas;
- Especificações matemáticas.

#### 2.1.4.3 Validação dos requisitos

A validação dos requisitos é a etapa onde todas as inconsistências e as questões não resolvidas são removidas. É preciso a certificação de que todos os requisitos analisados, extraídos e negociados sejam validados para que possam ser aceitos por todos (SOMMERVILLE, 2011), ou seja, busca validar se todos os requisitos estejam definidos de forma correta. Essa etapa é responsável por eliminar problemas de ambiguidade e inconsistências no documento de requisitos. Também, ela é importante porque corrigir um erro de definição de requisitos após a instalação de um produto é geralmente mais caro que corrigir um erro na especificação. A razão para isso é que a ocorrência de mudança dos requisitos normalmente significa que o projeto e a implementação do sistema também devem ser alterados.

Algumas técnicas de validação de requisitos podem ser usadas em conjunto ou individualmente (BENNACEUR et al., 2019):

- **Verificação de consistência** – requisitos no documento não devem entrar em conflito. Ou seja, não deve haver restrições contraditórias ou descrições diferentes da mesma função do sistema;
- **Verificação de completude** – o documento de requisitos deve incluir requisitos que definam todas as funções e as restrições pretendidas pelo usuário do sistema;

- **Revisões de requisitos** – consiste na revisão dos requisitos por um grupo de pessoas que analisam, discutem e apontam caminhos para solucionar os problemas encontrados;
- **Prototipação** – é desenvolvida com intuito de permitir uma melhor representação dos requisitos de um sistema. Com a prototipação, os usuários podem ter uma ideia antecipada de como o sistema executável funcionará;
- **Geração de casos de teste** – geralmente são realizados no final do processo do desenvolvimento do sistema. Se surgirem dificuldades para criar os casos de teste, há uma grande possibilidade de existirem problemas nos requisitos.

#### 2.1.4.4 Gerenciamento de requisitos

O gerenciamento de requisitos é um processo para compreender e controlar as mudanças dos requisitos do sistema (SOMMERVILLE, 2011). É uma das fases mais importante no processo de engenharia de requisitos.

É necessário manter o acompanhamento dos requisitos individuais e manter as ligações entre os requisitos dependentes, de modo que seja possível avaliar o impacto das mudanças de requisitos. O gerenciamento de requisitos deve iniciar assim que uma versão inicial do documento de requisitos esteja disponível. Durante o estágio de gerenciamento de requisitos, devem ser realizados (POHL, 2016):

- **Identificação dos requisitos** – cada requisito deve possuir algum tipo de identificação única de modo que possa ser feita a referência entre ele e outros requisitos;
- **Processo de gerenciamento de mudanças** – é o conjunto de atividades que avaliam o impacto e o custo da mudança;
- **Políticas de rastreabilidade** – definem os relacionamentos entre os requisitos e entre os requisitos e o projeto do sistema;
- **Apoio de ferramenta CASE (*Computer Aided Software Engineering*)** – o gerenciamento de requisitos envolve o processamento de grandes quantidades de informações sobre os requisitos. Podem ser usados sistemas especializados de gerenciamento de requisitos, planilhas e banco de dados. Como exemplo de ferramentas existentes atualmente no mercado usadas com este fim podemos citar

*Doors*<sup>1</sup>, *QSSRequireit*, *Caliber-RM* e *Requite Pro*<sup>2</sup>.

## 2.2 SISTEMAS EMBARCADOS

Nesta seção, serão apresentados os principais conceitos sobre sistemas embarcados, como o que se entende por seu surgimento, tipos e suas especificações.

### 2.2.1 Conceitualização de sistemas embarcados

Dada a variedade de aplicações para sistemas embarcados, não existe um consenso na conceitualização, mas existe um conjunto de características que são comuns a esses sistemas. Ribeiro e Ramos (2011) apresentam os sistemas embarcados como sistemas eletrônicos de processamento de informação embutidos em produto de forma transparente para o usuário. Oshana e Kraeling (2019), define os sistemas embarcados como sistemas microprocessados, cuja finalidade é de controlar um sistema ou dispositivo, diferentes, em diversos aspectos, de um computador para propósitos gerais.

Segundo Braun et al. (2010a), sistemas embarcados são sistemas técnicos que operam em um ambiente físico, e têm como objetivo controlar seu ambiente utilizando-se de variáveis que fazem referências às propriedades físicas ou técnicas do ambiente. Já segundo Jimenez et al. (2014), um sistema embarcado é um dispositivo que contém componentes de *hardware* e *software* fortemente acoplados para executar uma única função. Em geral, faz parte de um sistema maior, não se destina a ser programável de forma independente pelo usuário e deve funcionar com o mínimo ou nenhuma interação humana.

Os sistemas embarcados podem ser considerados como todo sistema computacional que não é utilizado para sistemas computacionais de propósito gerais. A palavra embarcada deve-se ao fato de o computador ser totalmente encapsulado ou dedicado ao dispositivo e executar apenas um grupo de programas pré-definidos.

---

<sup>1</sup>[https://www.ibm.com/support/knowledgecenter/pt-br/SSYQBZ\\_9.6.1/com.ibm.doors.requirements.doc/topics/c\\_welcome.html](https://www.ibm.com/support/knowledgecenter/pt-br/SSYQBZ_9.6.1/com.ibm.doors.requirements.doc/topics/c_welcome.html)

<sup>2</sup> <http://www.sinfic.pt/SinficNewsletter//sinfic/Newsletter65/FormacaoFundamentosRRP.html>

### 2.2.2 Histórico

Na década de 1940, os computadores eram por vezes dedicados a uma única tarefa. Eram sistemas muito grandes e não podiam ser considerados embarcados. A ideia de controlador programável foi desenvolvida algum tempo depois.

O primeiro sistema embarcado em massa surgiu em 1961, foi o LGM-30 Minuteman, que era um míssil nuclear formado por um disco rígido com a função de memória principal (OSSADA, 2010). Em 1966 entrou em produção a segunda versão do míssil, onde pela primeira vez utilizou-se grandes volumes de circuitos integrados. Essa tecnologia permitiu a redução do preço de circuitos integrados e, com isso, possibilitou o uso desses circuitos em escala comercial.

O primeiro Sistema Embarcado reconhecido foi o *Apollo Guidance Computer*, desenvolvido por Charles Stark Draper em 1968, através do *Massachusetts Institute of Technology*, junto ao *Instrumentation Laboratory* (YANG, 2008). Esse sistema teve como objetivo levar o homem à lua e operava em tempo real. Utilizava circuitos integrados monolíticos para redução de peso e consumo de energia.

Com o alto grau de diminuição e o constante aumento de processamento, em 1978, a NEMA (*National Engineering Manufacturers Association*), juntamente com a IEC (*International Electrotechnical Commission*), a ISO (*International Organization for Standardization*) e a NFPA (*Fire Protection Association*) publicou a norma ICS 3-1978 para microcontroladores programáveis, substituída em 1995 pela IEC 1131 (OSSADA, 2010).

Em meados da década de 1980, com a evolução da microeletrônica, pôde-se juntar vários componentes externos no mesmo chip do processador, o que ficou conhecido como microcontrolador.

### 2.2.3 Características e aplicações de sistemas embarcados

Os sistemas embarcados podem funcionar de duas maneiras: controle de tempo real e modo reativo (TONIOLO, 2012). Os dois modos de funcionamento são importantes para saber como programar o dispositivo, entender o comportamento da aplicação e como será seu funcionamento.

O modo de **Controle de Tempo Real** deve reagir a mudanças no ambiente e devem fornecer resultados em tempo real. Respeita os limites de tempo para a execução das tarefas, pois depende de resultados vindos de sensores, atuadores, *display*, entre outros (TONIOLO,

2012). Eles são classificados em dois tipos.

- **Soft Real Time:** executa tarefas em um determinado intervalo de tempo, mesmo que esse tempo não seja respeitado. Exemplo: sistema bancário (se uma tarefa não for realizada dentro do tempo determinado, apenas uma mensagem de erro aparecerá);
- **Hard Real Time:** o intervalo de tempo deve ser respeitado, podendo ter consequências graves, caso contrário. Exemplo: sistemas de controle de avião (uma falha pode resultar em queda e perdas de vidas).

No modo **Reativo**, o funcionamento se dá como resposta a eventos externos, havendo um limite de tempo em relação à entrada de sinais. Podem ser periódicos (quando os estímulos podem ser *loops*) e assíncronos (quando o acionamento ocorre através do pressionamento de botões por parte dos usuários). A saída deve ser realizada, exatamente, após os sinais de entradas começarem a atuar.

Segundo Ossada e Martins (2010), os sistemas embarcados podem ser de três tipos: propósito geral, sistemas de controle e processamento de sinais, conforme explicados a seguir:

- **Propósito geral:** semelhantes a computadores de mesa, havendo grande interação com o usuário, através de terminais de vídeos ou monitores. Exemplo: videogames, caixa de banco, outros;
- **Sistemas de controle:** com realimentação em tempo real. Geralmente não tem muita interação com o usuário, realiza-se com sensores e atuadores, e são sistemas mais robustos. Exemplo: sistemas automotivos, aviônicos, usinas nucleares, entre outros;
- **Processamento de sinais:** recebem um grande volume de informações a serem processadas em curto espaço de tempo. Os sinais são recebidos pelos conversores analógicos/digitais, processados e novamente convertidos em sinais analógicos. Exemplos: modems, tratamento de áudio, radares, outros.

#### 2.2.4 Famílias de sistemas embarcados

Sistemas Embarcados estão se tornando cada vez mais generalizados e praticamente já não podemos mais visualizar a sociedade vivendo sem estes sistemas.

Mais de 98% dos processadores do mundo estão hoje operando em sistemas embarcados, na imensa maioria das vezes, invisíveis ao usuário (JIMENEZ; PALOMERA; COUVERTIER, 2014). As pessoas não reconhecem nestes objetos computadores e, muitas vezes, nem percebem que dentro destes produtos escondem-se sistemas bastantes complexos.

Algumas famílias de sistemas embarcados demandam um alto grau de confiabilidade. Uma falha nestes sistemas pode ocasionar perda de vidas, danos financeiros e de imagem à empresa.

Existem muitas famílias de sistemas embarcados, algumas delas são apresentadas a seguir:

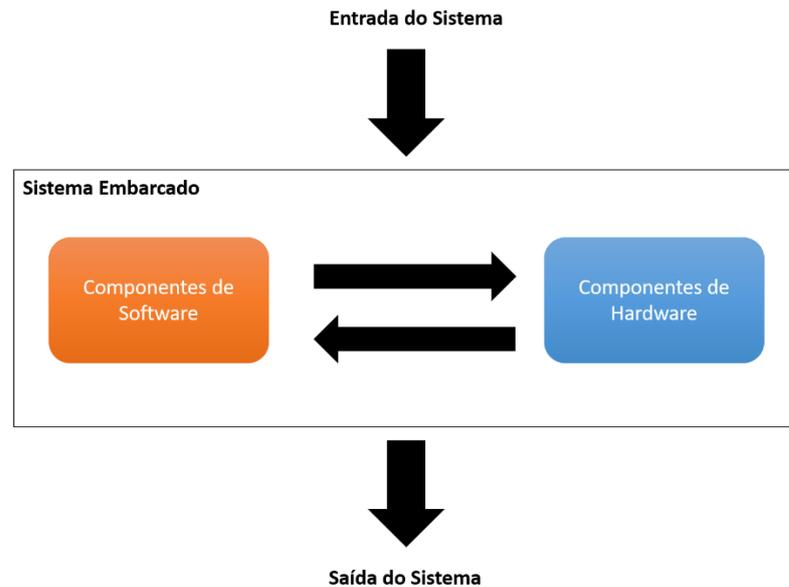
- **Indústria automotiva** – a tecnologia embarcada nos veículos está evoluindo constantemente evoluindo. A eletrônica e o *software* estão substituindo rapidamente a mecânica e a hidráulica (STARON, 2019). São sistemas de navegação automáticos, freios ABS (Anti-lock Braking System), controle de temperatura ambiente, AIBARG, entre outros.
- **Sistemas médicos** – a medicina atual está fortemente dependente de tecnologias. A tecnologia nessa família é caracterizada por intensa inovação, competição cada vez mais acirrada entre os fornecedores e uma consequente redução no ciclo de lançamento de produtos (ANDRÉ; ANDRADE, 2010). São usualmente utilizados em marca passo, bomba de insulina, em desfibrilador, raios X, entre outros. Também são considerados sistemas críticos.
- **Sistemas aviônicos** – esses sistemas devem responder em tempo curtíssimo a eventos, com falha zero (TAN et al., 2016). São altamente complexos e considerados sistemas críticos.
- **Eletrônica de consumo** – são sistemas cada vez mais complexos e únicos, onde cada aplicação deve estar exclusivamente ligada a um determinado aparelho. São produtos como celulares, aparelhos de DVD, câmeras digitais, vídeo games portáteis, entre outros.

### 2.2.5 Estrutura Geral de um Sistema Embarcado

De acordo com Jimenez, Palomera e Couvertier (2014), a estrutura de um sistema embarcado é dividida em dois conjuntos de componentes fortemente acoplados: componentes de *hardware* que inclui uma unidade central de processamento, normalmente na forma de um microcontrolador; e uma série de programas de *software*, normalmente incluídos como *firmware* que dão funcionalidade ao hardware. A

Figura 2 apresenta os dois conjuntos de componentes.

Figura 2 – Estrutura de um Sistema Embarcado



Fonte: Adaptado (JIMENEZ; PALOMERA; COUVERTIER, 2014)

As principais entradas em um sistema embarcado são variáveis de processo e parâmetros que chegam por meio de sensores e/ou portas de entrada/saída (E/S). As suas saídas estão na forma de ações de controle em atuadores do sistema ou informações processadas para usuários ou outros subsistemas dentro do aplicativo.

A estrutura de um sistema embarcado é composta por vários componentes, alguns desses componentes serão discutidos nas subseções seguintes.

#### 2.2.5.1 Memórias para sistema embarcado

Internamente a um sistema embarcado, o item mais evidente é a memória, pois é o local onde está armazenado o *software* embarcado. Há memórias para armazenamento de dados temporários e memórias para armazenar instruções básicas do processador.

Oliveira (2006) define as principais características das memórias: tempo de acesso, utilizado para realizar uma operação de leitura ou gravação; capacidade, quantidade efetiva de dados que podem ser armazenados na memória; não-volatilidade, capacidade de manter os dados armazenados mesmo com falta de energia elétrica; tempo de latência, intervalo mínimo entre cada operação de leitura ou escrita na memória que, se não respeitado, pode acarretar uso de espaços indevidos, podendo gerar perda ou manipulação incorreta dos dados.

### 2.2.5.2 Processadores para sistemas embarcados

Em sistemas embarcados existem algumas arquiteturas de processadores, as quais são desenvolvidas para uma aplicação específica. Sua principal característica é a finalidade para a qual será utilizada. Por exemplo, uma aplicação exige certos requisitos, para o qual arquiteturas diferentes de microprocessadores podem ser capazes de executá-los.

Os microprocessadores para sistemas embarcados são divididos em:

- **Processadores para propósito geral** – projetados para diversas variedades de aplicações, com o fim de maximizar o número de vendas de processadores. A utilização de processadores para propósito geral traz benefícios em métricas de desenvolvimento, como o ganho no tempo de desenvolvimento, uma vez que a arquitetura e as funções dos processadores já estão prontas, cabendo ao desenvolvedor apenas escrever o programa;
- **Processadores para propósito único** – projetados para executar somente um programa, no qual o projetista cria um processador com circuitos customizados para aplicação. Tem um alto rendimento, um baixo consumo de energia e um menor custo ao ser produzido em grande escala;
- **Processadores para aplicação específica** – projetados para uma classe específica de aplicações, como os processadores de sinais digitais (DSP), de telecomunicações e de controle. São consumidos em grande escala, usados em brinquedos, eletrônicos e eletrodomésticos.

### 2.2.5.3 Microcontroladores

São componentes eletrônicos utilizados para controlar circuitos, encontrados regularmente em diversos dispositivos eletrônicos. Eles são projetados para ter todos os componentes (processador, circuitos de memória, periféricos de entrada e saída) em um único circuito integrado, cabendo ao desenvolvedor codificar o programa (*firmware*) necessário para que o dispositivo funcione e, na maioria dos casos, somar alguns componentes externos que terão a função de entrada e saída de dados (OSSADA; MARTINS, 2010).

#### 2.2.5.4 Outros componentes de um sistema embarcado

Nesta seção serão apresentados os outros principais componentes de um sistema embarcado (JIMENEZ; PALOMERA; COUVERTIER, 2014):

- **Clock** – fornece a referência de tempo para o processador, ou seja, ele fornece um sinal que é uma “onda quadrada”, por isso é frequentemente chamado de oscilador. Um *clock* define a velocidade de processamento;
- **Registradores** – capazes de alocar pequena quantidade temporária dos dados. Trabalha diretamente com o processador, armazenando os resultados de suas operações lógicas e aritméticas temporariamente até que esse dado seja reutilizado em outra instrução ou transferido para outra forma de memória;
- **Contador de instrução** – é um registrador cuja função é armazenar o endereço da próxima instrução a ser executada; quando a instrução inicia a execução, seu conteúdo é logo atualizado para a próxima instrução a ser executada.

#### 2.2.6 Engenharia de requisitos para sistemas embarcados

A ER para desenvolvimento de sistemas embarcados ainda é pouco explorada. Projetos são abandonados devido ao fato dos requisitos terem sido mal formulados, evidenciando a necessidade de uma abordagem que auxilie a capturar requisitos de forma eficaz (ou seja, ideal para causar um resultado pretendido) durante a fase inicial de um projeto de sistema embarcado (OSSADA et al., 2012). Infelizmente, em muitos projetos de SE, a percepção de que os requisitos foram mal capturados ocorre tardiamente durante o ciclo de desenvolvimento do sistema embarcado, inviabilizando a implementação do sistema como um todo.

Os projetos de sistemas embarcados têm crescido ao longo dos anos, e a produção de *hardware* e *software* para esses sistemas exige que o tempo de projeto seja cada vez mais curto. Consequentemente, geram novos desafios para engenharia de requisitos. O grande desafio é produzir sistemas embarcados com alto grau de qualidade, que atenda ao mercado antes dele se tornar obsoleto. A questão do tempo de mercado (mais curto em relação a sistemas de propósito geral) faz com que o projeto tenha o tempo de desenvolvimento encurtado, forçando o início do desenvolvimento do sistema o mais rápido possível e obrigando um projeto de sistema embarcado a encontrar uma plataforma de *hardware* existente (OSSADA, 2010).

Geralmente os projetos de sistemas embarcados são conduzidos por engenheiros eletrônicos e mecânicos. O desenvolvimento do *hardware* é o primeiro a ser iniciado devido ao grande tempo para se projetar um processador, dada a dependência de fornecedores externos. Geralmente os engenheiros de *software* não estão comprometidos nas decisões do *hardware*. Por este motivo, o desenvolvimento do *software* se inicia quando o projeto de *hardware* já está pronto (OSSADA, 2010). Isso é ruim, porque muitas vezes o *software* não atende as expectativas do cliente e/ou usuário, um problema que poderia ser resolvido se, na construção do *hardware*, houvesse o envolvimento dos engenheiros de *software*. Diante disso, muitas empresas estão realizando mudanças, na qual os engenheiros de *software* estão sendo mais envolvidos no projeto de desenvolvimento de *hardware*.

A complexidade no desenvolvimento de sistemas embarcados está crescendo nos últimos anos, constituindo-se em um grande estímulo para a área de engenharia de requisitos. Para permitir que os processos de linha de tempo, a produtividade e a qualidade do *software* embarcado estejam otimizados, as empresas precisam empregar técnicas da engenharia de *software* em determinadas situações (GRAAF; LORMANS; TOETENEL, 2003).

Muitas abordagens de engenharia de requisitos não atendem a todas as particularidades de um sistema embarcado. Os SEs são diferentes de sistemas de propósito gerais, pois inúmeras funções a serem implementadas muitas vezes possuem soluções que podem ser programadas por *software* e por *hardware*, simultaneamente (OSSADA; MARTINS, 2010). No Quadro 2 é apresentado um resumo geral das diferenças entre sistemas embarcados e sistemas de propósito geral.

Quadro 2 – Diferenças entre sistemas embarcados e sistemas de propósito geral

Características	Sistemas embarcados	Sistemas de propósito geral
Participação de <i>stakeholders</i> com perfil técnico (engenheiros)	Muito frequente	Pouco frequente
Requisitos de tempo real	Comum, tendo grande impacto no produto.	Incomum
Utilização de <i>hardware</i> padronizado	Incomum. Normalmente, o <i>hardware</i> é projetado especificamente para o produto em desenvolvimento.	É comum a utilização de plataformas padronizadas
Restrições quanto ao tamanho do <i>software</i> executável	Presente na maioria dos projetos de SE	Normalmente não há
<i>Time to market</i>	Janela de tempo curta	Janela de tempo longa
Dimensões físicas do sistema	RFN muito importante a ser	Não se aplica

(volume, peso, ergonomia)	considerado	
Consumo de energia	RNF muito importante a ser considerado.	Não se aplica
Confiabilidade	RNF fundamental para a maioria dos casos, e mandatório em alguns casos	Necessário, mas com impacto menor no projeto global do sistema.
Definição de requisitos de <i>Hardware</i>	Forte impacto no projeto global	Baixo impacto no projeto global
Atores que interagem com o sistema	Dispositivos físicos como sensores e atuadores e, eventualmente, atores humanos.	Normalmente atores humanos

**Fonte: OSSADA (2010).**

### 2.3 CONSIDERAÇÕES DO CAPÍTULO

De maneira geral, esse capítulo apresentou informações relativas ao referencial teórico sobre a temática de pesquisa.

Inicialmente foi apresentado uma visão geral da engenharia de requisitos, mostrando alguns conceitos sobre requisitos e ER. Os requisitos podem ser classificados como funcionais e não-funcionais. Apresentamos as etapas da engenharia de requisito, são elas: elicitação, análise e negociação, especificação, validação e gerenciamento.

Foram apresentadas as definições e conceitos sobre sistemas embarcados, sua evolução histórica e as características comuns. Foram apresentadas as principais famílias de sistemas embarcados, a arquitetura geral de um SE, mostrando os principais componentes e a definição de microcontroladores com suas principais características. Ao final, foi apresentada a importância da engenharia de requisitos no desenvolvimento de sistemas embarcados.

### 3 ONTOLOGIAS E A WEB SEMÂNTICA

Este capítulo apresenta as tecnologias e metodologias da Web Semântica adotados para o desenvolvimento desta tese.

#### 3.1 WEB SEMÂNTICA

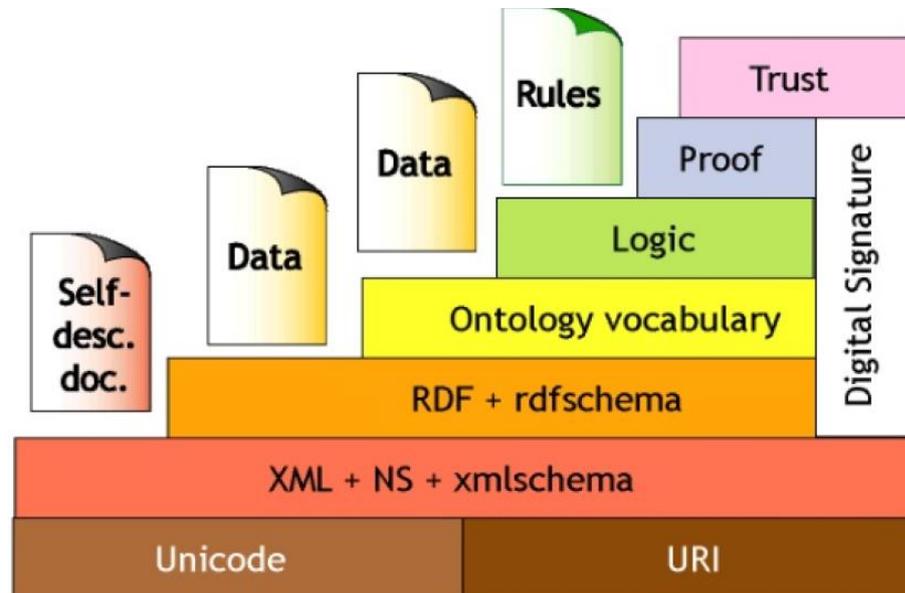
A iniciativa da Web Semântica empreendida pela World-Wide Web Consortium (W3C) tem estado ativa nos últimos anos e tem atraído interesse e dúvidas em igual medida. O W3C é um esforço coletivo onde as organizações membros, uma equipe em tempo integral e o público trabalham juntos para desenvolver padrões da web (YU, 2011). Liderada pelo inventor e diretor da Web Tim Berners-Lee e por Jeffrey Jaffe, a missão do W3C é levar a Web a todo o seu potencial. Ela é vista como a nova geração da Web atual, e se fundamenta em um conjunto de princípios de design e uma gama de tecnologias.

A Web Semântica se concretiza através de uma Web baseada em serviços, construída por aplicações baseadas em conhecimento. Desta forma, surge a noção de que a semântica é parte da Web, capturando a razão pela qual os recursos estão lá. “O conhecimento dito relevante para o homem é identificado e adicionado a um sistema computacional. Este, por sua vez, possui uma base de conhecimento, que uma vez explícita e bem estruturada, concebe os componentes de conhecimentos” (RODRIGUES, 2019). Portanto, o objetivo central da Web Semântica é criar uma extensão da Web atual onde o conhecimento recebe um significado bem definido, permitindo que pessoas e máquinas trabalhem de forma cooperativa.

**A partir do ano 2000, o consórcio W3C vem definindo um conjunto de tecnologias abertas sobre a Web atual, criando uma arquitetura em camadas (apresentada na**

Figura 3).

Figura 3 – Arquitetura da Web Semântica



Fonte: w3.org (2021).

A camada mais inferior na arquitetura (e que fornece subsídios para as demais camadas) compreende o padrão UNICODE<sup>3</sup> e o *Uniform Resource Indicator* (URI)<sup>4</sup>. Unicode é o padrão para representação de caracteres de computador, enquanto que a URIs é o padrão para identificar e localizar recursos (como páginas na Web). Estes fornecem uma linha de base para representar caracteres usados na maioria dos idiomas do mundo e para identificar recursos, respectivamente.

A segunda camada apresenta a linguagem de marcação *eXtensible Markup Language* (XML) e o *XMLSchema* (definição da estrutura do documento XML). O XML fornece uma maneira simples e padronizada de serializar informações representáveis como árvores rotuladas com anotações e referências cruzadas. Ela tem uma sintaxe útil tanto para humanos, quanto para todos os tipos de máquina. O XMLSchema é usado para descrever e validar a estrutura e o conteúdo dos dados XML. Ele define os elementos, atributos e tipos de dados.

A terceira camada da arquitetura abrange o *Resource Description Framework* (RDF) e o *Resource Description Framework Schema* (RDFS). RDF é a primeira camada da Web

<sup>3</sup> <https://unicode.org/>

<sup>4</sup> <http://www.asciitable.com/>

Semântica propriamente dita. Ela é uma estrutura de representação de metadados simples, usando URIs para identificar recursos baseados na Web e um modelo gráfico para descrever relacionamentos entre recursos. RDF especifica a semântica por intermédio de triplas: Recurso — Atributo — Valor (ou Sujeito — Predicado — Objeto). A RDFS estabelece as primitivas que podem ser usadas no documento RDF, como classes, subclasses, propriedades, subpropriedades e restrições, mas não impõe nenhuma ordem a ser seguida no documento.

A quarta camada (ontologias) é considerada a mais importante da Web Semântica. A *Web Ontology Language* (OWL) é uma linguagem da Web Semântica projetada para representar um conhecimento rico e complexo sobre coisas, grupos de coisas e relações entre coisas.

Na quinta camada, nível lógico, é possível descrever regras entre os conceitos mapeados na camada de ontologias, pois a camada de ontologias não oferece suporte para esse fim. Portanto, algumas linguagens são propostas na literatura, sendo elas: *Rule Markup and Modelling Language* (RuleML), *Semantic Web Rule Language* (SWRL), *Semantic Query Language for Databases* (SPARQL).

A últimas camadas tratam questões de Prova e Confiança que a Web Semântica pode suportar. A camada de Prova faculta a execução das regras do nível lógico e a camada de Confiança permite que as pessoas façam perguntas sobre a confiabilidade das informações na Web, a fim de fornecer uma garantia de sua qualidade.

### 3.2 ONTOLOGIA

Uma ontologia é uma especificação formal de uma conceptualização de um domínio (Gruber, 1993). Uma ontologia providencia um vocabulário que permite a representação e a comunicação do conhecimento relativo ao domínio da conceitualização (Gruber, 1993).

Ontologia é um formalismo para representar conceitos e relacionamentos de um domínio específico, ou seja, busca tornar explícito e interpretável o conhecimento deste domínio. A ontologia é organizada hierarquicamente em um número específico de termos e especificações de seus significados (Pease, 2007). A criação da ontologia contribui para resolver a heterogeneidade semântica na concepção de sistemas embarcados, entre outros domínios (Sousa, 2016). A heterogeneidade semântica surge de fontes de dados com diferentes formatos e linguagens de modelagem. Essa heterogeneidade é resultado de fontes de informação sendo construídas por diferentes pessoas com diferentes objetivos em mente (Yacout, 2014). Neste contexto, a ontologia é usada para remover essa heterogeneidade,

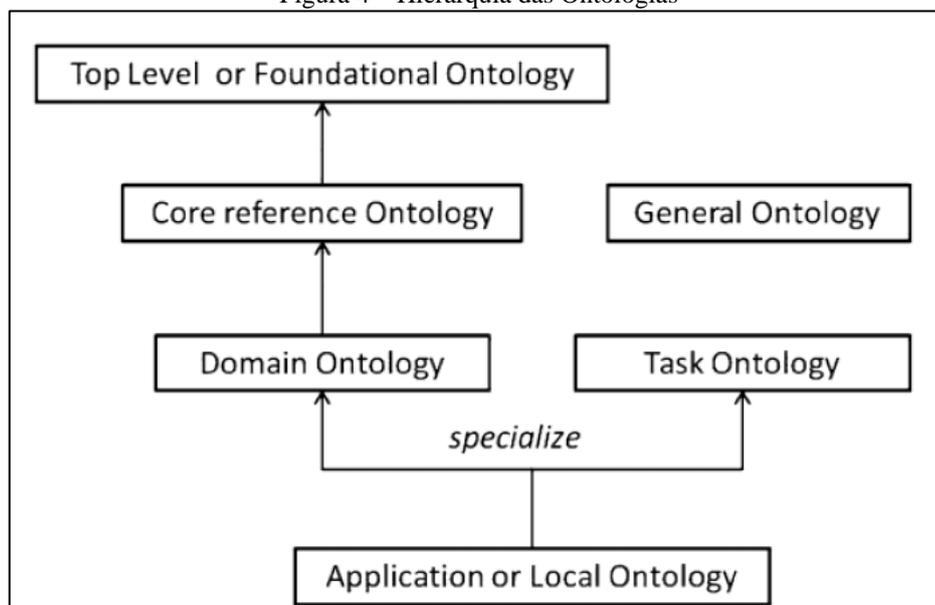
oferecendo um método para compartilhar um entendimento comum em uma estrutura de informações entre os envolvidos no projeto.

Uma das principais preocupações no desenvolvimento de sistemas embarcados está relacionada à comunicação entre as equipes de *hardware* e *software*, que se torna o principal desafio a ser superado neste contexto (Yacout, 2014). A falta de informações e mal-entendidos entre estas equipes pode afetar o produto final no cronograma e no orçamento, bem como a qualidade geral desse sistema. O uso da ontologia facilita a comunicação entre os engenheiros, pois apresenta um vocabulário comum, claro e compreensível entre as partes interessadas no projeto. Isso torna a especificação de requisitos mais precisa, pois reduz as ambiguidades.

### 3.2.1 Tipologia das Ontologias

Considerando que ontologias podem apresentar diversos tipos de abstrações e granularidades, existe um nível de detalhamento relacionado ao escopo de utilização definido para uma ontologia. Alguns autores sugerem formas de organizar ontologias de acordo com o nível de generalização, ou seja, o grau de proximidade da descrição do mundo. A Figura 4 descreve um modelo organizado em níveis, proposto por (ROUSSEY et al., 2011).

Figura 4 – Hierarquia das Ontologias



Fonte: (ROUSSEY et al., 2011).

- **Ontologias Gerais (*General*)** - compreende ontologias com classes mais genéricas, como espaço, tempo, processos, entre outros, independente do domínio;

- **Ontologias Centrais (*core*)** – é uma ontologia geral de um domínio específico, ou seja, especificam um conjunto mínimo de conceitos para um campo específico (Biologia, Medicina, Sistemas Embarcados ...), que ainda podem ser subdivididos em outros subdomínios (Biologia Molecular, Biologia Marinha, Fisioterapia, Cardiologia, Sistemas Embarcados Aviônicos, Sistemas Embarcados Automotivos, entre outros).
- **Ontologias de Domínio (*domain*)** – definem os conceitos (ou classes) de um domínio particular, geralmente, a partir de alguma ontologia de topo;
- **Ontologias de Tarefa** – com classes e relações destinadas a realizar uma tarefa ou atividade.;
- **Ontologias de Aplicação** – com entidades dependentes do domínio e de tarefa que exercem papel exclusivo na execução de uma aplicação.

### 3.2.2 Componentes de uma Ontologia

Para modelar os conhecimentos de um domínio arbitrário, evitando omissões, redundâncias e inconsistências, durante a atividade de conceitualização são utilizadas algumas representações intermediárias, construídas em torno dos principais componentes ontológicos descritos abaixo (CORCHO et al., 2005):

- **Classes:** as classes são conjuntos que contém os indivíduos. Podem ser descritas de forma que sejam apresentados os requisitos para participação na classe. Elas são representações concretas de conceitos do domínio.
- **Axiomas:** estabelecem sentenças que sempre são verdadeiras. São usados para representar um conhecimento que não pode ser expresso pelos outros componentes.
- **Relações:** são as propriedades que relacionam duas ou mais instâncias. Também são conhecidas como propriedades ou atributos.
- **Instâncias:** representam objetos no domínio de interesse, são também conhecidos como indivíduos.

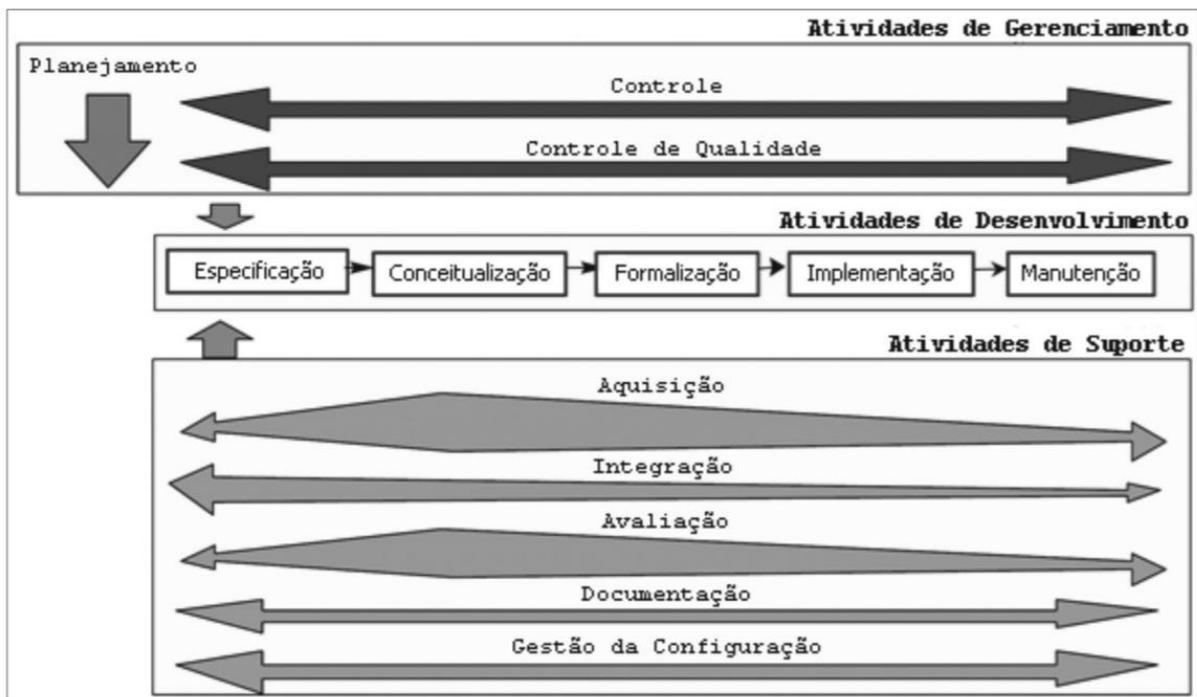
### 3.2.3 Engenharia Ontológica: Methontology

A construção de uma ontologia envolve uma sistemática de passos, e várias metodologias foram e estão sendo desenvolvidas para a engenharia ontológica. Portanto, nesta tese, foi escolhida a Methontology para construção da OntoREES. pois existe uma maturidade

maior dessa metodologia. A Engenharia Ontológica abrange um conjunto de métodos, processos, ferramentas e atividades que, juntos constroem uma metodologia a qual fundamenta o ciclo de vida das ontologias, desde sua especificação até sua implementação e manutenção (CORCHO et al., 2005). A Methontology, metodologia utilizada para construção das ontologias, foi desenvolvida pelo laboratório de Inteligência Artificial da Universidade Politécnica de Madrid, e foi aprimorada ao longo dos anos.

A Methontology define um framework de atividades que segue o modelo de prototipagem de Pressman (2016). Ela é uma metodologia bem estruturada para construir ontologias a partir do zero. Também, inclui em seus processos técnicas para entregas de atividades produzidas e fomenta o reuso de ontologias existentes. A Figura 5 apresenta a visão geral da Methontology.

Figura 5 – Visão Geral da Methontology



Fonte: (RAUTENBERG et al., 2008)

As atividades da Methontology estão divididas em três grupos: **atividades de gerenciamento**, **desenvolvimento** e **suporte**. As atividades de gerenciamento são responsáveis pelo planejamento de tarefas que precisam ser realizadas durante a construção da ontologia, os recursos necessários para cada, controle do andamento destas e analisa os resultados de cada uma. As atividades de desenvolvimento correspondem aos estágios que

dão corpo ao ciclo de vida da metodologia, sendo: Especificação, Conceitualização, Formalização, Implementação e Manutenção. Já as atividades de suporte são atividades realizadas durante todo o ciclo de vida da ontologia. O “nível” de exigência de cada tarefa se traduz na largura da seta que a acompanha, conforme apresentado na Figura 5. Discutimos com mais detalhes cada etapa da Methontology no 6.

### 3.2.4 Estratégias para Construção de Ontologias

Ainda sobre a Engenharia de Ontologias, existe uma estratégia proposta por Corcho et al., (2005), para auxiliar o processo de criação de ontologias, satisfazendo as necessidades de representação e uso correto da informação. Essas estratégias são definidas em:

- **bottom-up** – a maioria dos conceitos e estruturas conceituais do domínio, bem como a terminologia usada para expressá-los, estão contidos em documentos. A partir de conceitos mais específicos, constroem os termos mais genéricos. Esta estratégia sofre com problemas de reusabilidade.
- **top-down** – a construção da ontologia começa modelando os conceitos de nível superior, que são posteriormente refinados. Essa abordagem é normalmente realizada manualmente por especialistas de domínio e leva a uma ontologia com potenciais problemas de estabilidade.
- **mildle-out** – é uma versão híbrida (utiliza as estratégias bottom-up e top-down), onde é indicado os principais termos correlatos ao domínio (ou seja, os termos intermediários). Diante disso, é decidido a necessidade de especialização e generalização das categorias. Nesta tese foi adotada esse tipo de estratégia, por conseguir amenizar os problemas das outras.

### 3.3 LÓGICA DESCRITIVA (DL)

A Lógica Descritiva (do inglês, *Description Logic*, DL) é uma família de linguagens de representação de conhecimento usadas para representar o conhecimento de distintos domínio de aplicação de uma forma estruturada e formalmente bem definida (HORROCKS; SATTLER; TOBIES, 1999). Ontologias de alta qualidade são cruciais para a Web Semântica, e sua construção, integração e evolução dependem muito da disponibilidade de uma semântica bem definida e boas ferramentas de raciocínio. As diferentes aplicações em potencial, entretanto, requisitam diferentes níveis de expressividade e de complexidade de raciocínio, o

que justifica o uso de diferentes fragmentos da DL. Nas seções seguintes, será explorada a sintaxe e semântica da sublinguagem  $\mathcal{SROIQ}$  (KRÖTZSCH; SIMANČÍK; HORROCKS, 2014).

### 3.3.1 Sintaxe da Lógica Descritiva $\mathcal{SROIQ}$

A sintaxe de  $\mathcal{SROIQ}$  é definida em termo de três conjuntos finitos de símbolos disjuntos entre si: o conjunto das assinaturas dos conceitos ( $N_C$ ), o conjunto das assinaturas das relações ( $N_R$ ) e o conjunto das assinaturas dos indivíduos ( $N_I$ ) (KRÖTZSCH; SIMANČÍK; HORROCKS, 2014). O Quadro 3 apresenta a estrutura das expressões em  $\mathcal{SROIQ}$ .

Quadro 3 - Expressões em  $\mathcal{SROIQ}$

Expressões
$C ::= N_C \mid C \sqcap C \mid C \sqcup C \mid \neg C \mid \top \mid \perp \mid \exists R.C \mid \forall R.C \mid \geq n R.C \mid \leq n R.C \mid \exists$ $R.self \mid N_I$
$R ::= U \mid N_R \mid N_{-R}$

Fonte: (KRÖTZSCH; SIMANČÍK; HORROCKS, 2014)

A primeira regra na gramática define os construtores de conceitos, e a segunda define os construtores das relações. Definimos  $\mathcal{L}$  como o conjunto de todos os conceitos  $\mathcal{SROIQ}$ . Os construtores de conceitos podem ser classificados em construtores booleanos, restrições de relações, e enumerações nominais. Também, uma base de conhecimento DL ( $\mathcal{KB}$ ) é estruturada em dois componentes,  $\mathcal{KB} := \langle \mathcal{A}, \mathcal{T} \rangle$ , onde temos (RODRIGUES, 2019):

- **Axiomas assercionais ( $\mathcal{A}$ ):**
  - ✓ Instanciação de Conceito,  $C(N_I)$
  - ✓ Instanciação de Relação,  $R(N_I \times N_I)$
  - ✓ Igualdade e desigualdade de Indivíduos,  $N_I = N_I$ ,  $N_I \neq N_I$ , respectivamente; já que DL não suporta *Unique Name Assumption* (UNA), ou seja, instâncias podem referenciar o mesmo objeto no domínio (se não for dito nada ao contrário)
- **Axiomas terminológicos ( $\mathcal{T}$ ):**
  - ✓ Inclusão de conceitos,  $C \sqsubseteq C$
  - ✓ Equivalência de conceitos,  $C \equiv C$

Fragmentos sofisticados como  $\mathcal{SROJQ}$  possuem ainda um componente relacional ( $\mathcal{R}$ ), chamado de RBox, que contempla (RODRIGUES, 2019):

- ✓ Inclusão de propriedade,  $R \sqsubseteq R$ ;
- ✓ Equivalência de propriedade,  $R \equiv R$ ;
- ✓ Composição de propriedade,  $R \circ R \sqsubseteq R$ ;
- ✓ Propriedades Disjuntas, *Disjoint* ( $R, R$ );
- ✓ Propriedade Inversa,  $R \equiv R^{-}$ , além de outros axiomas, como transitividade, simetria (e assimetria), e reflexividade.

O Quadro 4 apresenta os construtores booleanos de operações para conjunção, disjunção e negação de conceitos lógicos.

**Quadro 4 – Construtos Booleanos de Conceitos  $\mathcal{SROJQ}$ .**

Nome	Sintaxe	Descrição
União	$\text{Pai} \sqcup \text{Mãe}$	Declaração que representa o conjunto de indivíduos formados por instâncias dos conceitos Pai <b>ou</b> Mãe.
Interseção	$\text{Mulher} \sqcap \text{Parente}$	Declaração que representa um conjunto de indivíduos que são, ao mesmo tempo, uma Mulher <b>e</b> Parente. O conceito Parente equivale a pai ou mãe.
Complemento	$\text{Mulher} \sqcap \neg \text{Casado}$	Construção que representa o conjunto de indivíduos do conceito Mulher que não são Casados.
Top Concept	$\top \sqsubseteq \text{Homem} \sqcup \text{Mulher}$	Utilizado quando se quer enfatizar todas as instâncias. Por exemplo, <b>todas</b> as instâncias no domínio citado são membros de Homem <b>ou</b> Mulher.
Bottom Concept	$\text{Homem} \sqcap \text{Mulher} \sqsubseteq \perp$	Estabelece que dois conceitos são disjuntos, isto é, não apresentam instâncias em comum. Exemplo: Homem é <b>disjunto</b> de Mulher.

**Fonte: A Autora (2021).**

A DL  $\mathcal{SROJQ}$  apresenta as restrições relacionadas as relações, o Quadro 5 apresenta essas restrições. São utilizados os quantificadores  $\exists$  (existencial) e  $\forall$  (universal).

**Quadro 5 – Construtores para restrições de relações  $\mathcal{SROJQ}$**

Nome	Sintaxe	Descrição
Restrição Existencial	$\exists \text{mãeDe.Mulher}$	Descreve aqueles indivíduos que são Mãe de <b>pele menos um</b> indivíduo Mulher, ou seja, aqueles que têm uma filha.
Restrição Universal	$\forall \text{filhoDe.Mãe}$	Define o conjunto com <b>todas as instâncias</b> que caso

		possuam um Filho, este deve ter uma Mãe.
Restrição de Domínio	$\exists \text{éFilho. } \top \sqsubseteq \text{Parente}$	Restringe o domínio da relação $\text{éFilho}$ para indivíduos do conceito $\text{Parente}$ .
Restrição Numérica	$\geq 2 \text{ filhoDe.Parente}$	Restringe o número de indivíduos participantes da relação. Portanto, descreve o conjunto de indivíduos que são filhos de pelo menos dois $\text{Parentes}$ (levando em consideração que $\text{Parentes} = \text{pai e mãe}$ ).

Fonte: A Autora (2021).

Para o quantificador existencial,  $\exists \mathbf{R.C}$ , é representado pelo conjunto dos indivíduos  $x$  tais que **existe um indivíduo**  $y$ , tal que  $R$  relaciona  $x$  com  $y$  e  $y$  está em  $C$ . Já para o quantificador universal, temos que  $\forall \mathbf{R.C}$  representa o conjunto dos indivíduos  $x$  tais que para **qualquer indivíduo**  $y$ , se  $R$  relaciona  $x$  com  $y$ , então  $y$  está em  $C$ .

### 3.3.2 Semântica da Lógica Descritiva $\mathcal{SROIQ}$

Segundo Reiter (1978), a semântica da lógica descritiva assume a suposição do mundo aberto (Open World Assumption (OWA)), segundo a qual a informação ausente não é tratada como informação negativa, mas como desconhecida. Essa semântica se refere a como as combinações válidas de símbolos, definidas pela sintaxe, devem ser lidas e interpretadas. A semântica de  $\mathcal{SROIQ}$  assemelha-se à semântica da lógica clássica de primeira ordem (lógica de predicados), pois muitas DLs podem ser vistas como fragmentos dessa lógica (OLIVEIRA, 2019).

Segundo Baader et al. (2017), a semântica da lógica descritiva é definida através do conceito de interpretação, assim sendo: uma Interpretação ( $\mathcal{I}$ ) é uma tupla  $\langle \Delta \mathcal{I}, \cdot^{\mathcal{I}} \rangle$ , onde  $\Delta \mathcal{I}$  representa o conjunto não vazio conhecido como o domínio de  $\mathcal{I}$ , enquanto que  $\cdot^{\mathcal{I}}$  é uma função que mapeia conceitos para subconjuntos de  $\Delta \mathcal{I}$ , relações para subconjuntos de  $\Delta \mathcal{I} \times \Delta \mathcal{I}$  e instâncias para elementos de  $\Delta \mathcal{I}$ .

Quanto mais axiomas uma ontologia possui, menos interpretações existem que satisfazem a todos os axiomas. Quanto menos interpretações satisfazem todos os axiomas da ontologia, mais axiomas podem ser satisfeitos por essas interpretações e, portanto, mais consequências lógicas se seguem da ontologia (KRÖTZSCH; SIMANČÍK; HORROCKS, 2014). A Figura 1 apresenta a sintaxe e semântica da  $\mathcal{SROIQ}$ .

Figura 6 – Sintaxe X Semântica da *SRQL*

Construtor	Sintaxe	Semântica
<b>Indivíduos</b>		
Nome do Indivíduo	$a$	$a^I$
<b>Papéis</b>		
Relação	$R$	$R^I$
Relação Inversa	$R^-$	$\{\langle x,y \rangle \mid \langle y,x \rangle \in R^I\}$
Relação Universal	$U$	$\Delta^I \times \Delta^I$
<b>Conceitos</b>		
Conceito	$A$	$A^I$
Intersecção	$C \sqcap D$	$C^I \cap D^I$
União	$C \sqcup D$	$C^I \cup D^I$
Complemento	$\neg C$	$\Delta^I \setminus C^I$
<i>Top Concept</i>	$\top$	$\Delta^I$
<i>Bottom Concept</i>	$\perp$	$\{\}$
Restrição Existencial	$\exists R.C$	$\{x \in \Delta^I \mid \exists y, (x,y) \in R^I \text{ and } y \in C^I\}$
Restrição Universal	$\forall R.C$	$\{x \in \Delta^I \mid \forall y, (x,y) \in R^I \Rightarrow y \in C^I\}$
Restrição Numérica	$\geq nR.C$	$\{x \in \Delta^I \mid \exists \text{ ao menos } n y, (x,y) \in R^I \text{ and } y \in C^I\}$
	$\leq nR.C$	$\{x \in \Delta^I \mid \exists \text{ no máximo } n y, (x,y) \in R^I \text{ and } y \in C^I\}$
Reflexividade Local	$\exists R.Self$	$\{x \mid \langle x,x \rangle \in R^I\}$
Nominais	$\{a\}$	$\{a\}^I$

Fonte: (KRÖTZSCH; SIMANČÍK; HORROCKS, 2014).

### 3.4 UFO: A ONTOLOGIA DE FUNDAMENTAÇÃO UNIFICADA

Em Guizzardi (2005) foi proposta uma ontologia de fundamentação – denominada *Unified Foundational Ontology* (UFO). Essa ontologia foi criada com o propósito específico de servir como uma teoria base para modelagem conceitual. Trata-se de um desenvolvimento com abordagem interdisciplinar de Ontologia Formal, Lógica Filosófica, Linguística e Psicologia Cognitiva. A UFO é derivada de outras ontologias de fundamentação: GFO/GOL (HERRE, 2010) e OntoClean/DOLCE (GUARINO; WELTY, 2000), porém é atualmente desenvolvida de forma independente dessas ontologias. Nesse contexto, outro trabalho que está diretamente relacionado com a UFO é a OntoUML. Ela é uma linguagem ontologicamente bem fundada para Modelagem Conceitual baseada em modelos ontológicos (GUIZZARDI et al., 2011). OntoUML é construída como uma extensão UML baseada na UFO.

A UFO fornece uma teoria de distinções e restrições ontológicas para conduzir a construção de modelos ontologicamente consistentes. Ela aborda uma tipologia de universais como os tipos sortais rígidos (*kind*, *subkind*), sortais anti-rígidos (*phase*, *role*), e os tipos dispersivos não sortais (ou seja, aqueles que definem conceitos mais genéricos, tais como

*mixin, category, rolemixin*), entre outros. O Quadro 6 apresenta os perfis dos tipos Universais da UFO.

Quadro 6 – Perfil dos Tipos Universais da UFO

Estereótipo	Descrição	Restrição
«kind»	Tipo Sortal Rígido	Não pode especializar «subkind», «phase», «role», «roleMixin».
«subkind»	Tipo Sortal Rígido	Não pode especializar «phase», «role», «roleMixin», e deve existir exatamente um supertipo «kind», direta ou indiretamente.
«phase»	Tipo Sortal instanciado apenas em certos contextos, e dependente de uma propriedade intrínseca	Definido como parte de uma partição. Deve existir exatamente um supertipo «kind», direta ou indiretamente.
«role»	Tipo Sortal instanciado apenas em certos contextos, e dependente de uma relação externa	Cardinalidade do lado oposto ao tipo «role» deve ser $\geq 1$ . Deve existir exatamente um supertipo «kind», direta ou indiretamente.
«mixin»	Tipo genérico representando propriedades abstratas de tipos disjuntos	Não pode especializar «kind», «subkind», «phase», «role», «roleMixin».
«category»	Tipo «mixin» rígido, que se especializa em diferentes «kinds»	Não pode especializar «kind», «subkind», «phase», «role», «roleMixin».
«rolemixin»	Tipo «mixin» que se especializa em diferentes «roles», com supertipos «kind» distintos	Não pode especializar «kind», «subkind», «phase», «role». Deve existir uma relação com cardinalidade $\geq 1$ no lado oposto da associação.

Fonte: (GUIZZARDI, 2005)

A UFO é organizada em três camadas:

- UFO-A (*Ontology of Endurants*): uma ontologia de endurantes, ou de objetos sem partes temporais, que persistem no tempo e mantêm sua identidade;
- UFO-B (*Ontology of Perdurants*): uma ontologia que incrementa a UFO-A com a introdução da noção de perdurantes (Perdurant); e
- UFO-C (*Ontology of Social and Intentional Entities*): com base na UFO-A e na UFO-B, a UFO-C é uma ontologia de entidades sociais e intencionais, incluindo aspectos linguísticos.

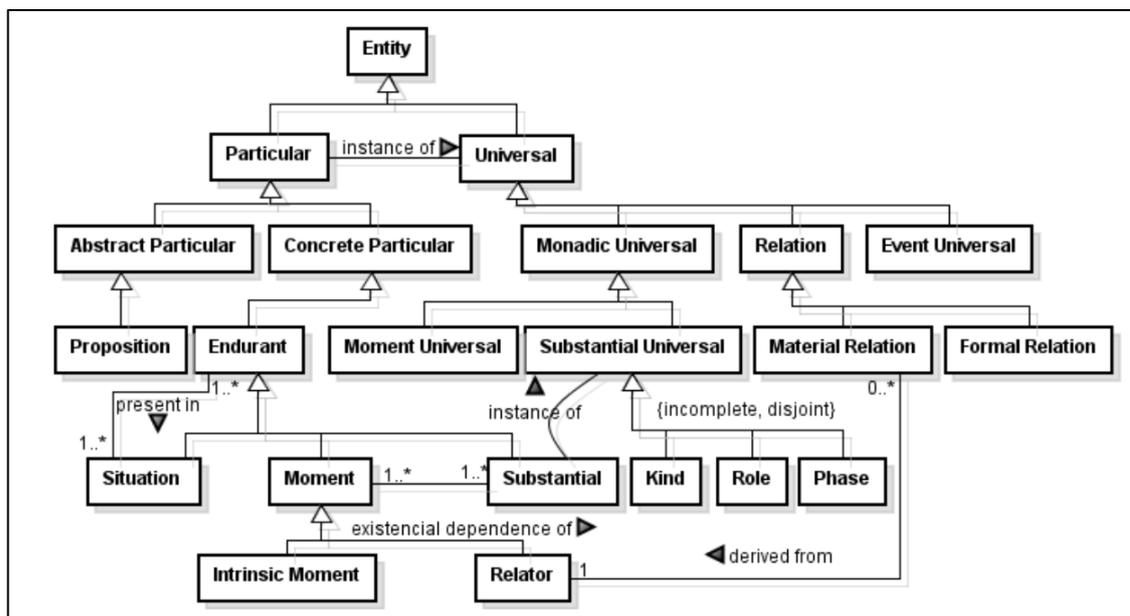
A UFO-A é o núcleo da ontologia UFO, porque sistematiza conceitos como tipos e estruturas taxonômicas, relações parte-todo, propriedade intrínsecas e espaços de valores, papéis, propriedades relacionais, entre outros. Nesta tese, utilizamos a UFO-A e UFO-C. Nas seções seguintes abordaremos mais detalhes sobre cada uma (UFO-A, UFO-B e UFO-C). Entretanto, não está no escopo desta tese descrever detalhadamente as particularidades

utilizadas na ontologia de fundamentação, pois a obra original (GUIZZARDI, 2005) e vários trabalhos publicados (GUIZZARDI, 2012; GUIZZARDI; FALBO; GUIZZARDI, 2008) durante todos esses anos já o faz de forma apropriada para seus objetivos.

### 3.4.1 UFO-A: Uma Ontologia para Endurantes

A Figura 7 apresenta uma visão parcial da UFO-A. Uma distinção fundamental nesta ontologia é entre as categorias de Particular (Instâncias) e Universal (Tipo). Particulares são entidades que existem na realidade e possuem uma identidade única. Os universais são padrões de características, que podem ser percebidos em uma série de particularidades diferentes. Ela é uma ontologia de Endurantes, que são entidades perenes que existem, mesmo com o passar do tempo. Eles podem ser existencialmente independentes (*Substantial*) ou existir somente quando associados a outro particular (*Moment*).

Figura 7 – Fragmentos da UFO-A



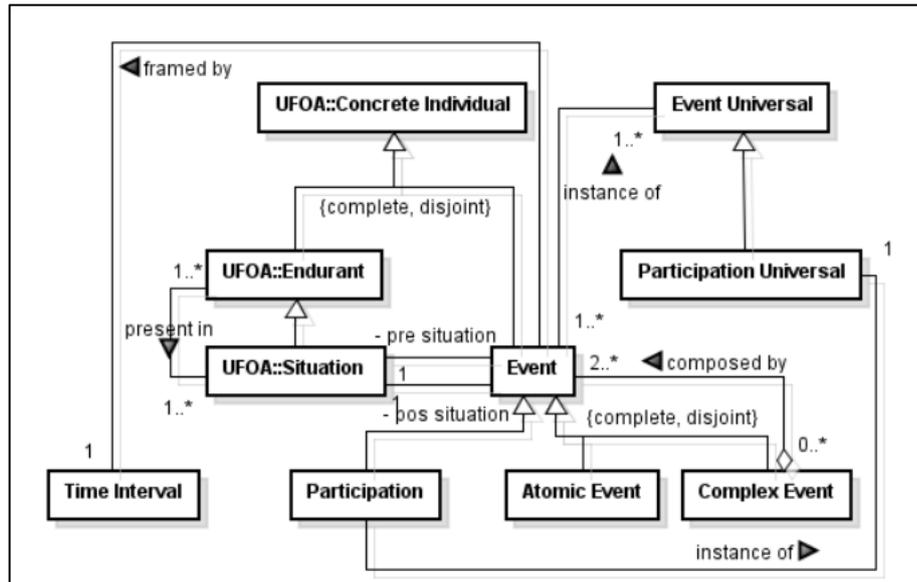
Fonte: (GUIZZARDI; FALBO; GUIZZARDI, 2008).

### 3.4.2 UFO-B: Uma Ontologia para Perdurantes

A UFO-B está relacionada a eventos (Perdurantes). As entidades contidas na UFO-B são delimitadas por um intervalo de tempo.

Alguns pontos de vista distintos descrevem a ontologia de Perdurantes, como destacado em Guizzardi (2012): a estrutura mereológica dos eventos, eventos como mapeamentos de uma situação anterior para uma situação posterior, a participação de objetos em eventos, a ordenação temporal de eventos, e eventos como manifestações das disposições de objetos.

Figura 8 – Fragmentos da UFO-B.

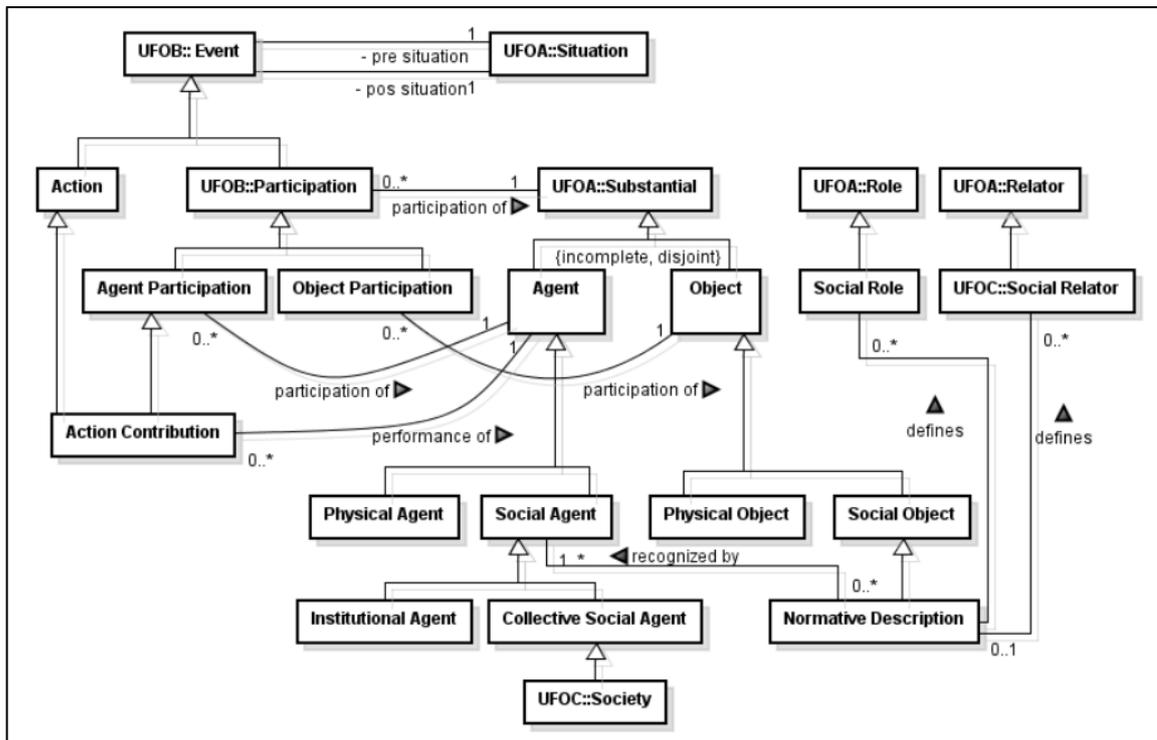


Fonte: (GUIZZARDI; FALBO; GUIZZARDI, 2008).

### 3.4.3 UFO-C: Uma Ontologia para Entidades Sociais

Com base na UFO-A e na UFO-B, a UFO-C é uma ontologia de entidades sociais e intencionais, incluindo aspectos linguísticos. A UFO-C aborda entidades como Agentes e Objetos. Os agentes podem ser físicos (por exemplo, uma pessoa) ou sociais (por exemplo, uma organização). Os objetos também podem ser categorizados em objetos físicos e sociais. Os objetos físicos incluem um livro, uma árvore, um carro, e os objetos sociais incluem dinheiro, linguagem e descrições normativas. A Figura 9 apresenta um fragmento da UFO-C.

Figura 9 – Fragmentos da UFO-C.



Fonte: (GUIZZARDI; FALBO; GUIZZARDI, 2008).

### 3.5 PADRÕES E ANTI-PADRÕES

Analisando de maneira geral, um padrão é algo que serve como base de referência para determinar características e/ou qualidade a algo. Várias áreas de pesquisas científicas têm proposto padrões para ajudar na resolução de determinados problemas. Diante desse contexto, pesquisadores da área ontológica começaram a definir padrões para modelagem conceitual. Falbo (2013) define que um padrão de ontologia descreve um problema de modelagem recorrente e particular, que surge em contextos de desenvolvimento de ontologia específicos e apresenta uma solução comprovada para o problema.

A UFO define padrões de modelagem, tais como: *Role Pattern*, *Phase Partition Pattern*, *RoleMixin Pattern*, e *Relator* e *Material Relations Pattern*. Guizzardi e Sales (2014) dispõe de um catálogo de anti-padrões, tais como: *Relation Specialization*, *Imprecise Abstraction*, *Relator With Overlapping Roles*, e *Twin Relator Instances*. A ontologia proposta nesta tese (OntoREES), foi avaliada com relação aos padrões definidos pela UFO. Portanto, no 7 é apresentada as estratégias de modelagem utilizadas.

### 3.6 SÍNTESE DO CAPÍTULO

De maneira geral, esse capítulo apresentou informações relativas ao referencial teórico, especificamente sobre Web Semântica e Ontologias. Foi apresentado conceitos de ontologias e quais as linguagens OWL existem. Também destacamos a engenharia de ontologias, e a lógica descritiva. Por fim, foi apresentada a UFO, que foi utilizada para criação da Ontologia de Requisitos para Sistemas Embarcados apresentada nesta pesquisa

O próximo capítulo apresentará informações referentes ao processo de revisão sistemática da literatura conduzido nesta pesquisa.

## 4 REVISÃO SISTEMÁTICA DA LITERATURA

Este capítulo apresenta os procedimentos realizados durante o desenvolvimento da revisão sistemática da literatura definido para esse trabalho. Esta revisão é uma extensão de um mapeamento sistemático que foi publicado por Souza et al.(2016), e visou traçar um panorama geral sobre ontologias em sistemas embarcados e os problemas em aberto nessa área. Segundo Kitchenham (2010), um mapeamento sistemático provê uma visão geral de uma área de pesquisa, permitindo que as evidências sejam plotadas em um alto nível de granularidade. Diante disso, as evidências apresentadas no mapeamento despertaram a necessidade de uma revisão sistemática da literatura para ampliarmos o entendimento sobre a temática de pesquisa. Com base no protocolo definido neste estudo, realizamos uma atualização, incluindo outras questões de pesquisa. As informações detalhadas sobre o mapeamento sistemático da literatura encontram-se no Apêndice A.

Através dos resultados alcançados e apresentados no presente capítulo, a seguinte questão de pesquisa pode ser respondida: **QP01**- Quais estudos estão disponíveis na literatura, que tratam sobre ontologias de requisitos em projetos de sistemas embarcados?

### 4.1 INTRODUÇÃO

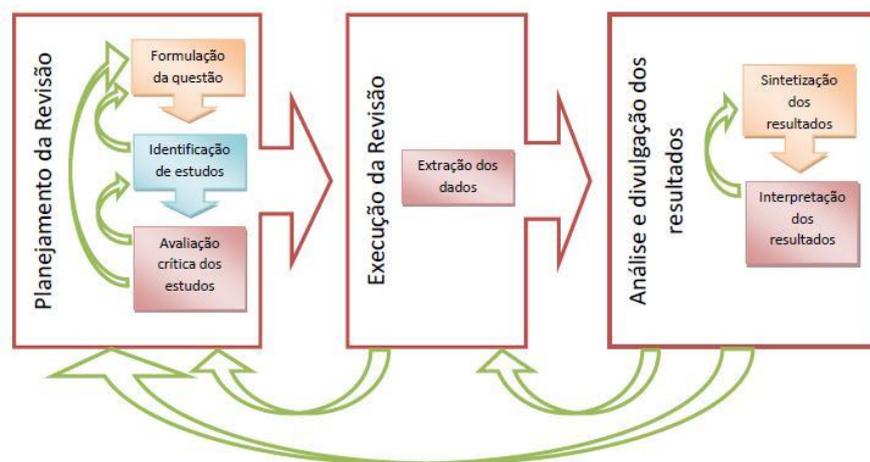
A revisão de literatura, também chamada revisão bibliográfica, é a fundamentação teórica adotada para tratar o tema e o problema de pesquisa (DYBÅ; KITCHENHAM; JØRGENSEN, 2004).

Uma revisão sistemática da literatura (RSL) é utilizada para se referir a um método específico de pesquisa, desenvolvida com o objetivo de reunir e avaliar as evidências disponíveis referentes a um tema específico (MIAN et al., 2007). Atua como um meio para identificar, avaliar e interpretar toda pesquisa relevante e disponível sobre uma questão de pesquisa específica, tópico ou fenômeno de interesse, fazendo uso de uma metodologia de revisão que seja confiável, rigorosa e que permita auditagem (MAFRA; TRAVASSOS, 2006). Diferente da revisão de literatura tradicional (conduzida de forma não sistemática), ela estabelece um processo formal para conduzir a investigação, evitando a introdução de rumo da revisão de literatura informal, dando maior credibilidade à pesquisa em andamento (SAMPAIO; MANCINI, 2007). A partir da pesquisa em um tópico em particular, este tipo de revisão pode induzir a identificação, seleção e produção de evidências, considerando os conhecimentos e as iniciativas existentes no campo de interesse (KITCHENHAM, 2004). A

metodologia pode ser caracterizada por usar métodos sistemáticos e explícitos, que são escolhidos com o objetivo de minimizar erros, possibilitando assim pesquisas mais confiáveis e que possam ser usadas, inclusive, para a tomada de decisões (MAFRA; TRAVASSOS, 2006). Desta forma, a revisão sistemática é um estudo retrospectivo e secundário, que tem por objetivo reunir estudos semelhantes, publicados ou não, avaliando-os criticamente em sua metodologia e reunindo-os numa análise estatística, a metanálise, quando isto é possível (SAMPAIO; MANCINI, 2007).

A revisão sistemática envolve uma série de atividades importantes, atreladas a um conjunto de fases dentro do processo de condução, estabelecidos dentro de um protocolo que conduzirá, de forma sistematizada, todo o processo de condução da revisão. Mian et al. (2007) e Kitchenham (2004) estabelecem três fases do processo de condução: planejamento da revisão, execução da revisão, e análise e divulgação dos resultados. As fases de condução da revisão e as atividades utilizadas neste trabalho foram adaptadas dos trabalhos citados anteriormente e estão detalhadas na Figura 10 e no Quadro 7, incluindo ainda um comparativo entre uma revisão sistemática e uma revisão de literatura comum. As fases do processo de revisão sistemática não precisam ser predominantemente sequenciais: elas podem ser iniciadas na fase de planejamento e refinadas posteriormente, conforme apresentado na Figura 10.

**Figura 10 – Processo de condução da revisão sistemática.**



**Fonte: (KITCHENHAM, 2004; MIAN et al., 2007).**

**Quadro 7 – Fases, atividades e comparativo do processo de revisão sistemática e revisão da literatura comum**

<b>Fases do processo de condução</b>	<b>Atividades</b>	<b>Revisão sistemática</b>	<b>Revisão de literatura comum</b>
<b>Planejamento da revisão</b>	Formulação da questão	Questão focada no objetivo da pesquisa (ideia-tema).	Questão focada no objetivo da pesquisa, embora

			permita uma questão mais abrangente, fora da ideia-tema.
	Identificação de estudos	Uso de estratégia de pesquisa transparente, que leva a resultados embasados cientificamente.	Sem uma estratégia definida, tornando os resultados menos relevantes.
	Avaliação crítica dos estudos	Critérios específicos de inclusão e exclusão são utilizados para avaliação	De forma geral, os estudos identificados não são avaliados de acordo com critérios.
<b>Execução da revisão</b>	Extração dos dados	Utilização de formulários padronizados para catalogação e avaliação posterior desses dados.	Geralmente os dados são extraídos de forma singular, sem a utilização de um modelo padrão.
<b>Análise e divulgação dos resultados</b>	Sintetização dos resultados	Os dados são analisados de forma descritiva, levando-se em conta a qualidade dos estudos, amostras extraídas e outros fatores que possam intervir nos resultados.	Os dados são avaliados de forma isolada, sem levar em conta a qualidade dos estudos nos quais foram extraídos os dados.
	Interpretação dos resultados	Os resultados interpretados estão menos sujeitos a desvios do sentido original dos estudos e opiniões de pessoas invasivas.	A interpretação dos resultados tende ao acúmulo de desvios do sentido original dos estudos obtidos, além de acumular opiniões pessoais.

Fonte: Adaptado de (PAI et al., 2004).

## 4.2 FASE DE PLANEJAMENTO

Durante a fase de planejamento foram estabelecidos os objetivos da pesquisa e foi criado um protocolo de revisão, adaptado do modelo de Mian et al.(2007) e Kitchenham (2004), contendo itens como: identificação e seleção das bases de dados, métodos de busca e palavras-chave, estratégia de busca e critérios de inclusão e exclusão dos estudos. Esta seção apresenta alguns tópicos do protocolo que guiou o estudo. As informações detalhadas presentes no protocolo de pesquisa podem ser visualizadas no Apêndice B.

### 4.2.1 Questões de pesquisa

Com o objetivo de enriquecer e fortalecer os conhecimentos teóricos dos pesquisadores em relação ao domínio do problema – a engenharia de requisitos em sistemas embarcados – procurou-se responder a algumas questões de pesquisa. Foram elas:

- Q1 – Quais as etapas no processo de desenvolvimento dos sistemas embarcados foram apoiadas pelo uso de ontologias?
- Q2 – Quais os tipos de contribuição do estudo?
- Q3 – Quais os domínios de Sistemas Embarcados que o estudo suporta?
- Q4 – Quais as linguagens de representação de ontologias foram utilizadas?
- Q5 – Quais os requisitos (funcionais e não-funcionais) identificados?
- Q6 – Quais os desafios/problemas identificados na literatura relacionadas a ontologias em sistemas embarcados?

#### 4.2.2 Estratégia de busca

A construção dos termos de busca foi realizada seguindo uma estratégia composta pelos seguintes passos:

- A partir das variáveis independentes e dependentes, as principais palavras-chave são identificadas;
- Sinônimos são identificados baseados em artigos conhecidos e relevantes na área de pesquisa;
- A *string* de busca é gerada a partir da combinação dessas palavras-chave e sinônimos. São usados os operadores OR (ou) entre os operadores identificados e AND (e) entre as palavras-chave.

Os termos e sinônimos identificados são apresentados abaixo:

- **Ontologia:** Inglês (*ontology*, *ontologies*).
- **Sistemas embarcados:** Inglês (*embedded systems*, *Safety Critical Systems*, *Real Time Systems*, *Embedded Product*).

O Quadro 8 apresenta a *string* de busca gerada.

**Quadro 8 – String de busca.**

<i>String</i>
(("ontology" OR "ontologies") AND ("embedded system" OR "embedded software" OR "real time system" OR "safety critical systems" OR "embedded product"))

**Fonte: A Autora (2021).**

### 4.2.3 Fontes de busca

A busca por trabalhos foi realizada de forma eletrônica, através de mecanismos de busca de *sites web* especializados e de renome científico-acadêmico. Três critérios para a seleção das fontes foram: disponibilidade de consultar os artigos na *web*, presença de mecanismo de busca usando palavras-chave; e importância e relevância das fontes. O

Quadro 9 apresenta as fontes utilizadas.

**Quadro 9 – Bases de dados utilizadas.**

<b>Fonte de busca</b>	<b>URL para Acesso</b>
<i>IEEE Xplorer</i>	<a href="http://www.ieeexplore.ieee.org/Xplore">http://www.ieeexplore.ieee.org/Xplore</a>
<i>Science Direct</i>	<a href="http://www.sciencedirect.com">http://www.sciencedirect.com</a>
<i>Scopus</i>	<a href="http://www.scopus.com">http://www.scopus.com</a>
<i>ACM Digital Library</i>	<a href="http://portal.acm.org">http://portal.acm.org</a>
<i>Web of Science</i>	<a href="https://login.webofknowledge.com/">https://login.webofknowledge.com/</a>

**Fonte: A Autora (2021).**

### 4.2.4 Critérios de inclusão e exclusão dos estudos

A inclusão de um estudo é determinada pela relevância em relação as questões de pesquisa, determinada pela análise do título, palavras-chave, resumo, introdução e conclusão.

Os seguintes critérios de inclusão e exclusão foram definidos:

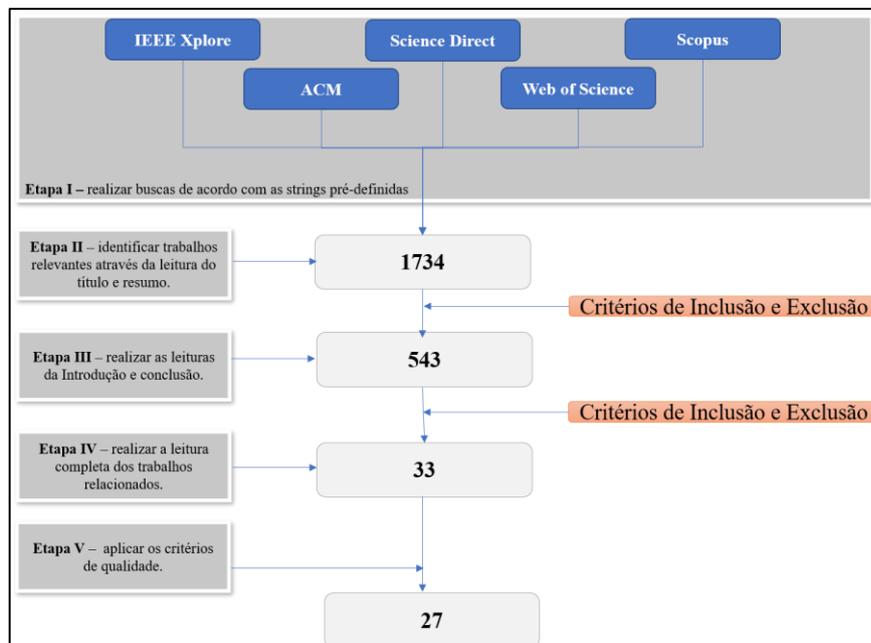
- Estudos primários;
- Estudos publicados entre 2000 a 2019;
- As fontes devem estar na *web*;
- As fontes devem disponibilizar os trabalhos na íntegra;
- Estudos repetidos (se o estudo estiver disponível em diferentes fontes de busca, a primeira pesquisa será considerada);
- Para trabalhos que representam os mesmos resultados de pesquisa, será aceito o trabalho que apresentar os dados de forma mais completa.

### 4.3 FASE DE EXECUÇÃO

Nessa etapa, foi realizada a condução e análise dos estudos primários no período compreendido entre 15/08/2019 e 21/01/2020. Foram aplicados os critérios de inclusão e exclusão definidos no protocolo de pesquisa e os resultados registrados. As buscas foram realizadas e os resultados passaram por cinco etapas de seleção, conforme estabelecido no protocolo da pesquisa e apresentado na

Figura 11.

**Figura 11 – Etapas para seleção dos trabalhos.**



**Fonte: A Autora (2021).**

Para apoiar a extração e registro dos dados e posterior análise, todos os trabalhos encontrados na Etapa I foram exportados para a ferramenta StArt<sup>5</sup>. Em seguida, foi aplicada a Etapa II do protocolo, onde foram lidos os títulos dos trabalhos e resumos e analisados quais estavam disponíveis por completo, sendo selecionados 543 dos 1734 estudos resultantes da busca inicial. A Etapa III consistiu em avaliar os resultados por meio da leitura da introdução e conclusão, sendo selecionados 33 dos 543 estudos. Na Etapa IV, os trabalhos aprovados na

<sup>5</sup> <http://lapes.dc.ufscar.br/software/start-tool>

etapa anterior passaram por uma leitura completa. Por fim, na Etapa V, em todos os trabalhos selecionados foram aplicados critérios de qualidade, resultando em 27 estudos primários. Os trabalhos considerados aprovados foram selecionados para compor a tese.

#### 4.4 AVALIAÇÃO DE QUALIDADE

Realizar uma avaliação de qualidade dos estudos selecionados é uma forma de evitar vieses e maximizar a validade interna e externa (KEELE, 2007). Para avaliação dos estudos selecionados, utilizou-se os critérios de qualidade apresentado no Quadro 5.

A avaliação da qualidade dos estudos será realizada por meio de uma técnica de pontuação para avaliar a credibilidade, integridade e relevância dos estudos selecionados. Os critérios de avaliação foram adaptados da literatura e proposto pelo autor. As possibilidades de respostas são: 0 (não atende = NA), 1 (atende = A) e 0.5 (atende parcialmente = AP).

**Quadro 10 – Critérios de Qualidade.**

#	Critério	Possibilidades de Respostas
C1	Existe uma justificativa porque o estudo foi realizado?	$A = 1, NA = 0, AP = 0.5$
C2	O artigo é baseado em pesquisa (ou apenas um relatório de lições aprendidas com base na opinião de especialistas)?	$A = 1, NA = 0$
C3	Existe uma afirmação clara sobre os objetivos da pesquisa?	$A = 1, NA = 0, AP = 0.5$
C4	A abordagem proposta está claramente descrita?	$A = 1, NA = 0, AP = 0.5$
C5	As abordagens suportam mais de um domínio (automotivo, aviônico, médico, outros.)	$A = 1, NA = 0$
C6	O estudo é suportado por alguma ferramenta?	$A = 1, NA = 0$
C7	Existe uma discussão sobre os resultados do estudo?	$A = 1, NA = 0, AP = 0.5$
C8	A limitação desse estudo é explicitamente discutida?	$A = 1, NA = 0, AP = 0.5$
C9	Existe uma descrição clara sobre os problemas em aberto relacionadas ao estudo proposto?	$A = 1, NA = 0, AP = 0.5$

**Fonte: A Autora (2021).**

#### 4.5 FASE DE ANÁLISE E DIVULGAÇÃO DOS RESULTADOS

Nesta fase, os estudos primários que atendiam ao propósito da revisão sistemática foram analisados criticamente e sintetizados no formulário de aprovação dos trabalhos, através da preparação de resumos, conforme Kitchenham (2004). O Quadro 11 apresenta os estudos selecionados após a fase de avaliação de qualidade.

**Quadro 11 – Trabalhos selecionados.**

<b>ID</b>	<b>Ano</b>	<b>Trabalho</b>	<b>Referência</b>	<b>Fonte</b>
<b>EP-05</b>	2002	A component-based approach for embedded software development	(YEN et al., 2002)	IEEE
<b>EP-04</b>	2007	Device Modeling for a Flexible Embedded Systems Development Process	(THAMBOULIDIS; DOUKAS; KOUMOUTSOS, 2007)	IEEE
<b>EP-01</b>	2010	Interacting Entities Modelling Methodology for Robust Systems Design	(MEZHUYEV; SPUTH; VERHULST, 2010)	IEEE
<b>EP-24</b>	2010	Semantic management of nonfunctional requirements in an e-Health system	(KOAY; KATARIA; JURIC, 2010)	Scopus
<b>EP-03</b>	2011	A conceptual framework for semantic case-based safety analysis	(DARAMOLA et al., 2011)	IEEE
<b>EP-06</b>	2012	Integrating metrics in an ontological framework supporting SW-FMEA	(BICCHIERAI et al., 2012)	IEEE
<b>EP-13</b>	2012	Using Ontologies to Reason About the Usability of Interactive Medical Devices in Multiple Situations of Use	(BOWEN; HINZE, 2012)	ACM
<b>EP-15</b>	2013	Towards a core ontology for robotics and automation	(PRESTES et al., 2013)	Science Direct
<b>EP-21</b>	2013	OntCheck: An ontology-driven static correctness checking tool for component-based models	(LIN; ZHANG; GU, 2013)	Scopus
<b>EP-22</b>	2013	An automated transformation from OntoUML to OWL and SWRL	(BARCELOS et al., 2013)	Scopus
<b>EP-07</b>	2014	Transferring research into the real world: How to improve RE with AI in the automotive industry	KÖRNER, Sven J et al., 2014	IEEE
<b>EP-11</b>	2014	Integrating security mechanisms into embedded systems by domain-specific modelling	(VASILEVSKAYA et al., 2014)	Web of Science
<b>EP-19</b>	2014	Ontology-based runtime reconfiguration of distributed embedded real-time systems	(HOFTBERGER; OBERMAISSER, 2014)	Scopus

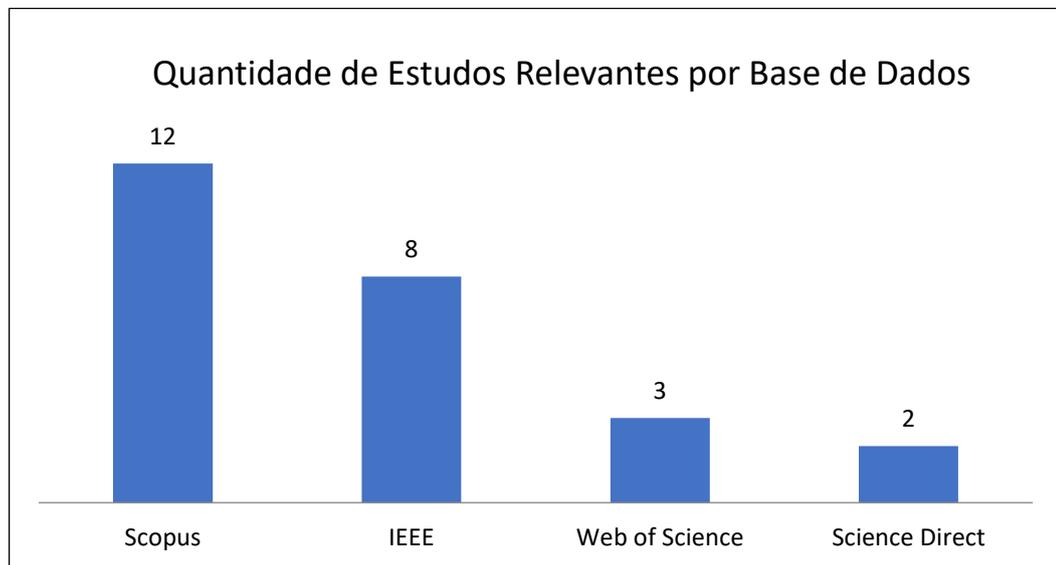
<b>EP-20</b>	2014	A semantic support for testing activities of safety-critical embedded systems	(VENTICINQUE; MAZZOCCA; VENTICINQUE, 2014)	Scopus
<b>EP-23</b>	2014	Towards a general framework for modeling, simulating and building sensor/actuator systems and robots for the Web of Things	(DIACONESCU; WAGNER, 2014)	Scopus
<b>EP-10</b>	2015	A Kind of Safety Requirements Description Method of the Embedded Software Based on Ontology	(LIU; AOYAMA, 2015)	Web of Science
<b>EP-12</b>	2015	Support for Cross-domain Composition of Embedded Systems Using MARTE Models	(VASILEVSKAYA; NADJIM-TEHRANI, 2015)	ACM
<b>EP-25</b>	2015	Design of ontology-based generative components using enriched feature diagrams and metaprogramming	(ŠTUIKYS; DAMAŠEVIČIUS, 2015)	Scopus
<b>EP-09</b>	2016	Development and Evaluation of a Software Requirements Ontology	(TAN et al., 2016)	Web of Science
<b>EP-14</b>	2016	Requirements and languages for the semantic representation of manufacturing systems	(NEGRI et al., 2016)	Science Direct
<b>EP-17</b>	2016	Fostering concurrent engineering of cyber-physical systems: A proposal for an ontological context framework	(DAUN et al., 2016)	Scopus
<b>EP-18</b>	2016	Semantic design space refinement for model-based systems engineering	(SCHMIT et al., 2016)	Scopus
<b>EP-02</b>	2017	Ontology-based service discovery framework for dynamic environments	(ZESHAN et al., 2017)	IEEE
<b>EP-16</b>	2017	OWL ontology to ecore metamodel transformation for designing a domain specific language to develop aviation scenarios	(JAFER; CHHAYA; DURAK, 2017)	Scopus
<b>EP-08</b>	2019	An Ontological Model for Map Data in Automotive Systems	(SURYAWANSHI et al., 2019)	IEEE
<b>EP-26</b>	2019	ADISTES ontology for active diagnosis of sensors and actuators in distributed embedded systems	(SAEED et al., 2019)	Scopus
<b>EP-27</b>	2019	An Ontology-Based Method for HW/SW Architecture Reconstruction	(MUSAVI; HASHEMI, 2019)	Scopus

**Fonte: A Autora (2021).**

As informações a seguir mostram dados quantitativos referentes aos trabalhos selecionados na etapa de seleção final da RSL. Percebe-se que na fase final foram selecionados 27 estudos, distribuídos da seguinte forma: a base Scopus, com 45%, foi a que apresentou a maioria dos resultados; a IEEE apresentou 30%; a Web of Science contribuiu com 11%; e a ACM e Science Direct representam 7% cada. Os Gráficos 1 e 2 apresentam os

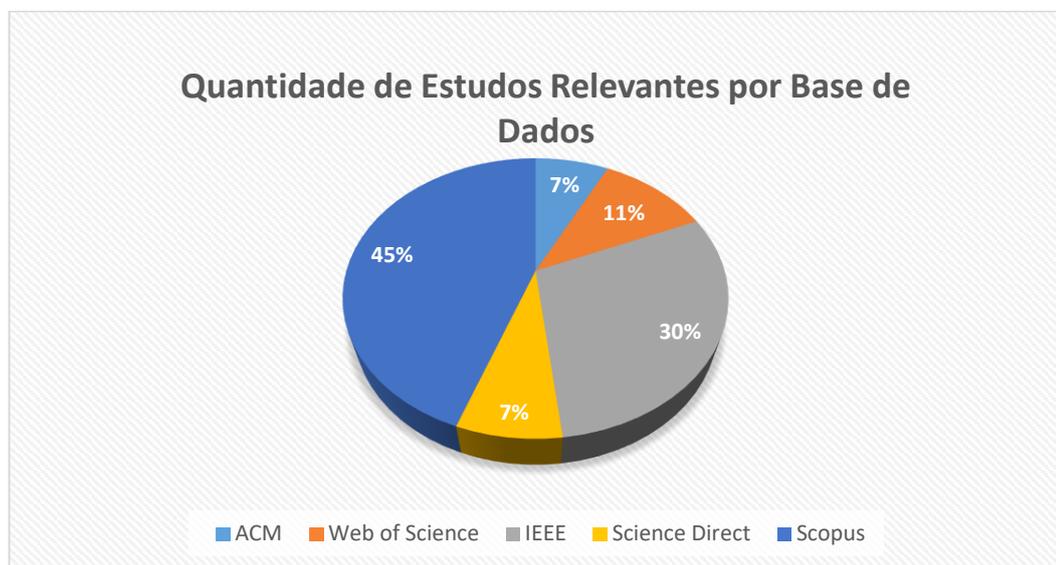
resultados consolidados da revisão sistemática, referentes ao número e porcentagem de trabalhos retornados por base científica.

**Gráfico 1 – Quantidade de trabalhos relevantes por base dados.**



**Fonte: A Autora (2021).**

**Gráfico 2 – Porcentagem de trabalhos relevantes por base de dados.**



**Fonte: A Autora (2021).**

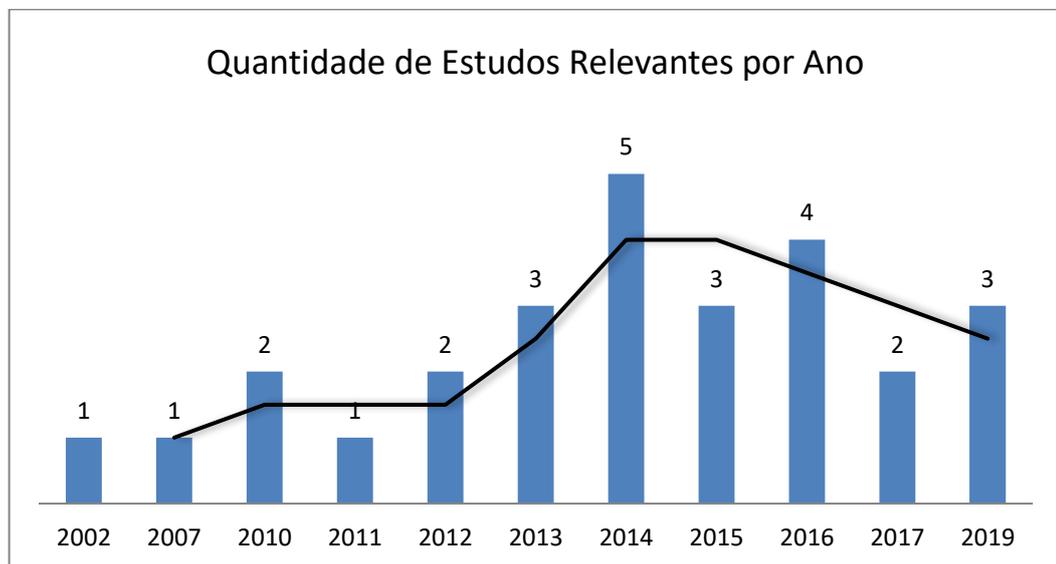
## 4.6 RESULTADOS E ANÁLISE

Nesta seção, apresentamos os resultados obtidos por meio da extração de dados dos 27 artigos que preencheram os critérios de inclusão. Apresentamos uma visão geral das características gerais de os estudos, bem como os resultados das questões de pesquisa.

### 4.6.1 Ano de Publicação

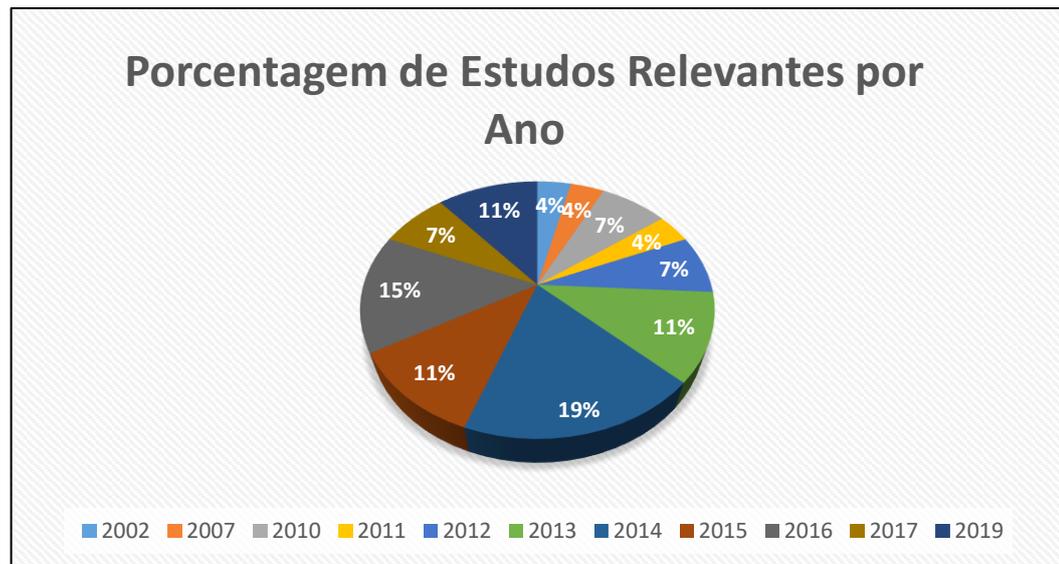
Os estudos analisados nessa RSL foram publicados entre 2000 a 2019, apresentados nos Gráficos 3 e 4, sendo o ano de 2014 o que obteve mais resultados, com 19%, seguido pelo ano de 2016, representando 15%, 2013, 2015 e 2019 com 11% cada, 2010, 2012 e 2017 com 7% cada e 2002, 2007, e 2011 representaram 4% cada. Mesmo considerando o aumento do número de artigos nos últimos anos, é possível observar que a quantidade de publicações na área começou a crescer a partir de 2010, podendo isso representar um aumento no interesse pelo tema de pesquisa. Nos últimos cinco anos teve um aumento significativo na quantidade estudos sobre o tema, destacando o ano de 2014 e 2016.

**Gráfico 3 – Quantidade de estudos relevantes por ano.**



**Fonte: A Autora (2021).**

Gráfico 4 – Porcentagem de estudos relevantes por ano

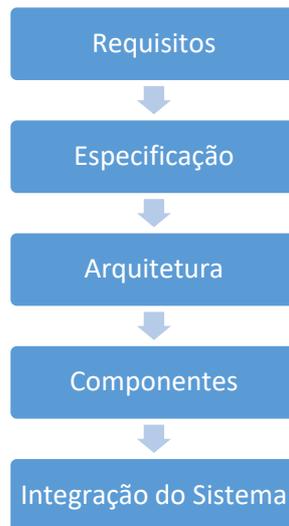


**Fonte: A Autora (2021).**

#### 4.6.2 Q1 – Quais as etapas no processo de desenvolvimento dos sistemas embarcados foram apoiadas pelo uso de ontologias?

O objetivo dessa questão é identificar, no desenvolvimento de sistemas embarcados, em quais as etapas as ontologias têm sido utilizadas (essas incluem todo o processo de desenvolvimento desses sistemas). O processo de Wolf (2008) foi utilizado para fazer essa classificação. Esse processo propõe cinco etapas em um nível alto de abstração (requisitos, especificação, arquitetura, componentes e integração do sistema). Quando utiliza um processo geral de desenvolvimento, os membros da equipe podem entender mais facilmente o que devem fazer, o que devem receber de outros membros da equipe em determinados momentos, e o que eles devem entregar quando eles completam suas tarefas. A Figura 12 apresenta as cinco etapas do processo de Wolf (2008) e o Quadro 12 apresenta as etapas abordadas pelos estudos.

**Figura 12 – Etapas do Processo de Wolf.**



**Fonte: Adaptado (Wolf, 2008)**

Segundo Wolf (2008), as etapas são:

- **Requisitos** – antes de projetarmos um sistema, precisamos saber o que estamos projetando. Essa etapa é responsável pelo levantamento dos requisitos. Os requisitos podem ser divididos em requisitos funcionais e não-funcionais. Ontologias podem ser utilizadas nessa etapa;
- **Especificação** – a especificação é mais precisa. Serve como um contrato entre o cliente e os engenheiros. Essa etapa deve ser cuidadosamente escrita, pois reflete com precisão os requisitos do cliente;
- **Arquitetura** – a especificação não diz como o sistema faz as coisas, apenas o que o sistema faz. Descrever como o sistema implementa as funções é o objetivo dessa etapa. Portanto, esta deve ser concebida para satisfazer os requisitos funcionais e não-funcionais;
- **Componentes** – o objetivo dessa etapa é estruturar os sistemas em componentes de *hardware* e *software*. Os componentes são definidos em conformidade com as etapas de especificação e arquitetura; e
- **Integração do sistema** – responsável por integrar e testar todas as partes do sistema. Erros são normalmente encontrados durante essa etapa, e um bom planejamento pode ajudar a encontrar os erros mais rapidamente.

Quadro 12 – Etapas do processo de desenvolvimento abordados nos estudos.

Estudos Primários	Etapas do Processo de Desenvolvimento de Sistemas				
	Requisitos	Especificação	Arquitetura	Componentes	Integração
EP-01	X	X	X		
EP-02	X				
EP-03	X				
EP-04			X	X	
EP-05	X			X	
EP-06	X				
EP-07			X		
EP-08		X			
EP-09			X		
EP-10	X	X			
EP-11	X	X			
EP-12	X	X			
EP-13			X		
EP-14	X				
EP-15	X				
EP-16	X				
EP-17	X	X			
EP-18	X	X			
EP-19			X		
EP-20	X	X			
EP-21			X		
EP-22	X				
EP-23		X			
EP-24	X	X			
EP-25		X	X		
EP-26					X
EP-27			X		

Fonte: A Autora (2021).

Analisando o Quadro 12, percebemos que os estudos utilizam ontologias em cinco etapas do desenvolvimento de sistemas (requisitos, especificação, arquitetura, componentes e integração). A etapa de requisitos é abordada por 60% dos estudos. Esse resultado era esperado, pois ontologias é uma técnica utilizada para levantamento de requisitos. O uso de ontologias na etapa de requisitos facilita o levantamento, porque possibilita uma descrição exata do conhecimento envolvido na área de sistemas embarcados, evitando que os requisitos sejam interpretados de maneiras diferentes por diversos *stakeholders*, o que é muito comum em linguagem natural, onde as palavras têm semântica diferente do contexto do sistema.

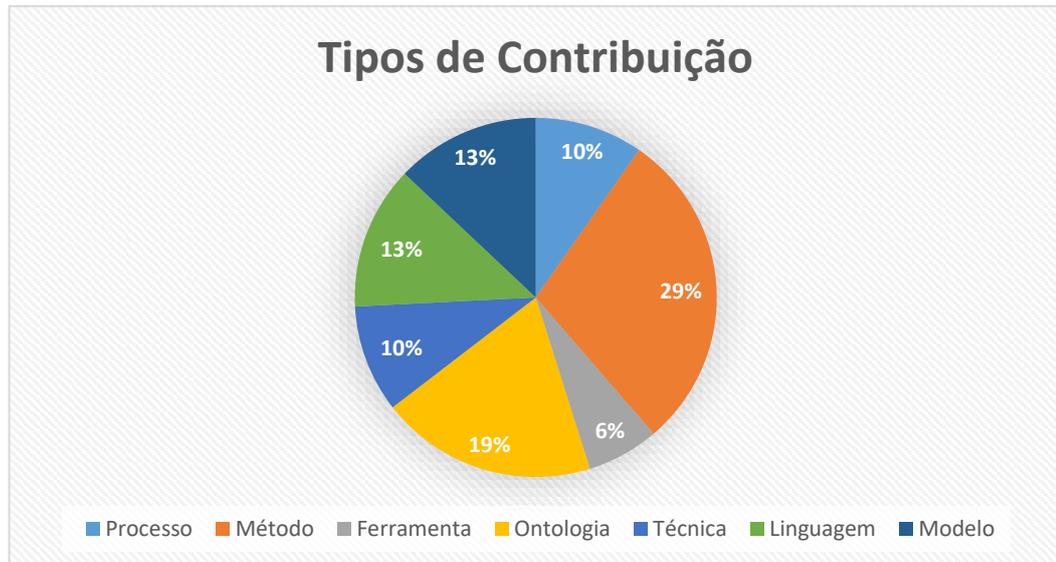
A etapa de especificação também é abordada por muitos estudos (41%). De fato, até certo ponto, esse resultado era esperado, já que as ontologias podem ser usadas para especificar formalmente os requisitos no desenvolvimento de sistemas embarcados, com o mínimo de interpretações dúbias.

A etapa da arquitetura apresentou 33% dos estudos. Componentes e integração não é tão representativa, apenas 8% e 4% respectivamente. Isso pode indicar que um pequeno número de artigos está preocupado com o uso de ontologias nessas etapas ou outras técnicas têm sido mais precisa para definição dessas etapas. Esse resultado é importante, pois poderia indicar um nicho a ser descoberto sobre a temática de pesquisa.

#### 4.6.3 Q2 – Quais os tipos de contribuição do estudo?

Nessa questão o objetivo foi identificar e descrever quais os tipos de contribuição propostas nos estudos (método, modelo, processo, técnica, ferramenta, linguagem, ontologia). A contribuição do estudo poderia ser uma ontologia, ou alguma abordagem que utilizava ontologia para desenvolvimento dos sistemas embarcados. O Gráfico 5 apresenta a quantidade de contribuição dos estudos (um estudo pode ter mais de uma contribuição).

Gráfico 5 – Tipos de Contribuição do Estudo.



**Fonte: A Autora (2021).**

Alguns estudos apresentaram mais de uma contribuição. Analisando o gráfico, percebemos que Método foi a principal contribuição dos estudos, com 29% (9 estudos).

Ontologia teve 19% (6 estudos). Analisando as ontologias propostas, verificou-se que os principais termos dos sistemas embarcados não são representados e nenhuma faz uma classificação de requisitos funcionais e não-funcionais. Diante disso, no 6 será apresentada uma ontologia específica de domínio levando em consideração requisitos funcionais e não-funcionais. Foram propostas três questões de pesquisa para analisar as ontologias identificadas nos estudos.

Linguagem e modelo obtiveram 13% cada (4 estudos), seguido de técnica e processo, 10% cada (3 estudos). Ferramenta obteve apenas 6% (representando apenas um estudo).

#### 4.6.4 Q3 – Quais os domínios de Sistemas Embarcados que o estudo suporta?

**O objetivo desta pergunta foi identificar os diferentes domínios que os estudos apoiam. Isto pode ajudar a reconhecer quais os domínios precisam de maior atenção, no que diz respeito à ontologia para SE. A pesquisa não se concentrou em um tipo específico de domínio. O**

Quadro 13 apresenta os tipos de domínios explorados pelos estudos. É importante notar que um estudo poderia ter abordado mais de um domínio do sistema embarcado.

**Quadro 13 – Domínios dos Sistemas Embarcados apoiados pelos estudos.**

<b>Domínio</b>	<b>Estudo</b>	<b>Quantidade</b>	<b>%</b>
Médico	EP-13 e EP-24	2	7,5%
Sistemas Críticos	EP-08 e EP-20	2	7,5%
Aviônico	EP-09, EP-16 e EP-18	3	11%
Sistemas de Tempo Real	EP-06, EP-15 e EP-19	3	11%
Nenhum domínio específico	EP-01, EP-02, EP-03, EP-04, EP-05, EP-07, EP-10, EP-11, EP-12, EP-14, EP-17, EP-21, EP-22, EP-23, EP-25, EP-26 e EP-27	17	63%

**Fonte: A Autora (2021).**

O resultado desta questão de pesquisa é importante para identificar os diferentes domínios que são abordados pelos estudos e para quais domínios as ontologias estão sendo propostas. A partir dos resultados do Quadro 9, sistemas aviônicos e sistemas de tempo real foram mais representados (11% cada), seguido de médico e sistemas críticos (7,5% cada). A maioria dos estudos não apresentam um domínio específico (63%, 17 estudos primários). Os sistemas embarcados podem ser classificados como automotivos, médicos, eletrônicos de consumo, aviônicos, sistemas críticos, sistemas de tempo real, entre outros. A opção “nenhum domínio específico” foi utilizada para estudos que abordam o uso de ontologias para qualquer tipo de sistemas embarcados. Esses estudos tratam conceitos gerais desses tipos de sistemas, deixando de classificar particularidades específicas de cada domínio. Esse resultado era esperado, pois um fator que dificulta uma maior padronização é o fato de existirem diversos domínios como sistemas automotivos, médicos, eletrônicos de consumo e outros, cada um com características diferentes.

#### **4.6.5 Q4 – Quais as linguagens de representação de ontologias foram utilizadas?**

O objetivo dessa questão foi identificar quais as linguagens de representação de ontologias foram utilizadas nos estudos. A classificação dessas linguagens foi definida de acordo com as recomendações semânticas da W3C (World Wide Web Consortium) - por exemplo, OWL, XML, SPARQL, SWRL, entre outras. No entanto, durante a extração dos dados, outras linguagens foram identificadas e inseridas para responder a essa questão de

pesquisa (UML e SysML). O Quadro 11 apresenta a distribuição das linguagens identificadas nos estudos.

**Quadro 14 – Linguagens das Ontologias.**

Domínio	Estudo	Quantidade	%
OWL	EP-01, EP-04, EP-05, EP-06, EP-07, EP-08, EP-09, EP-12, EP-13, EP-14, EP-16, EP-17, EP-18, EP-19, EP-20, EP-21, EP-22, EP-24, EP-25, EP-26 e EP-27	21	77%
UML/SysML	EP-02, EP-03, EP-06, EP-07, EP-11, EP-12, EP-14, EP-15, EP-16, EP-22, EP-23	11	40%
SWRL	EP-02, EP-06, EP-22 e EP-25	4	15%
SPARQL	EP-06 e EP-20	2	7,5%
Não específica	EP-10	1	3%

**Fonte: A Autora (2021).**

A resposta a essa pergunta é importante porque pode servir como um guia para pesquisadores que tenham interesse em usar alguma linguagem ontológica para o desenvolvimento de sistemas embarcados.

A OWL é utilizada pela maioria dos estudos (77%, ou 21 estudos), seguido por UML (40%, ou 11 estudos), SWRL (15%, ou 4 estudos) e SPARQL (7,5%, ou 2 estudos). Apenas um estudo não especificou a linguagem utilizada. Um artigo poderia ter utilizado mais de uma linguagem para representar a ontologia.

Os 27 estudos examinados usam uma ou mais linguagens recomendadas pela W3C. Vale ressaltar que a maioria dos estudos que abordam a UML/SysML também utilizam alguma linguagem recomendada pelo W3C (EP-02, EP-06, EP-07, EP-12, EP-14, EP-16, EP-22, EP-23). Assim, notavelmente, os artigos estão preocupados com o uso dos padrões recomendados pela W3C para especificar e propor ontologias. No entanto, notamos que, embora OWL seja a linguagem mais expressiva e aceita, a OWL não foi utilizada por todos os estudos incluídos nesta revisão. Assim, pode-se dizer que muitos estudos que propõem o uso de ontologias no campo de sistemas embarcados não estão apenas utilizando tecnologias padrão para especificar e manusear ontologias.

#### 4.6.6 Q5 – Quais os requisitos (funcionais e não-funcionais) identificados?

O objetivo dessa questão é identificar qual a contribuição do estudo com relação aos tipos de requisitos (funcionais e não-funcionais) abordados. É importante investigar se as abordagens propostas foram usadas para gerenciar requisitos funcionais ou não-funcionais. Os requisitos identificados foram utilizados para definição da ontologia proposta nesse trabalho (Capítulo 4).

Segundo Belgamo (2000), requisitos são necessidades coletadas de clientes e usuários para sabermos o que o sistema a ser idealizado fará. Eles podem ser de dois tipos: funcionais e não-funcionais, conforme apresentado na seção 2.1.

Em projetos de sistemas embarcados, não é interessante evidenciar apenas um tipo de requisito, pois um requisito funcional pode impactar um requisito não-funcional, ou o contrário (GRAAF; LORMANS; TOETENEL, 2003; NETO, 2010; OSSADA et al., 2012). Quando tratamos de sistemas críticos, como aviônicos e médicos, um erro na especificação de um requisito pode provocar perdas de vidas. Portanto, a elaboração de ontologias de requisitos para sistemas embarcados, deve evidenciar os dois tipos de requisitos (funcionais e não-funcionais).

Lidar com os requisitos não-funcionais não é uma tarefa fácil, mas seu tratamento é vital para a construção de sistemas embarcados. Esses requisitos não são adequadamente cobertos pela maioria das técnicas de engenharia de requisitos, principalmente quando se trata de requisitos de sistemas embarcados. Esses requisitos são muitos diversos e, usualmente, devem ser expressos em formas específicas de domínio. Os requisitos não-funcionais mais explorados nos estudos foram: qualidade (desempenho, confiabilidade, segurança) e *hardware* (requisitos operacionais e ambientais). Esses requisitos foram utilizados para definição das Ontologias propostas no 6.

#### 4.6.7 Q6 – Quais os benefícios, identificados na literatura, relacionados ao uso de ontologia no desenvolvimento de sistemas embarcados?

O objetivo dessa questão é apresentar os benefícios existentes relacionados ao uso de ontologia no desenvolvimento de sistemas embarcados. É importante porque identifica um conjunto de direções futuras de pesquisa. Os resultados desta questão de pesquisa são importantes para considerar se há evidências para afirmar que o uso de ontologias pode beneficiar positivamente o processo de desenvolvimento de um sistema embarcado. O Quadro

15 apresenta os principais benefícios da utilização de ontologias no desenvolvimento de sistemas embarcados.

**Quadro 15 – Benefícios do uso de ontologia em SE.**

<b>Benefícios</b>	<b>Estudos</b>	<b>Quantidade</b>	<b>%</b>
Facilitar a etapa de definição de requisitos (diminuição das ambiguidades e inconsistência entre os requisitos)	EP-05, EP-06, EP-07, EP-10, EP-11, EP-13, EP-14, EP-18, EP-20, EP-21	10	37%
Não especificou	EP-02, EP-14, EP-16, EP-17, EP-22, EP-23 e EP-26	7	26%
Facilitar a comunicação entre os <i>Stakeholders</i>	EP-01, EP-08, EP-09, EP-10, EP-15, EP-25	6	22%
Representação do conhecimento do domínio (requisitos)	EP-01, EP-06, EP-12, EP-14, EP-21	5	18%
Representação dos requisitos não funcionais	EP-03, EP-05, EP-06, EP-13, EP-24	5	18%
Integração entre os requisitos e arquitetura ( <i>hardware</i> e <i>software</i> )	EP-04, EP-05, EP-19 e EP-27	4	15%

**Fonte: A Autora (2021).**

Os resultados mostram que a maioria dos estudos estão utilizando ontologias para auxiliar o processo de desenvolvimento dos sistemas embarcados. É importante destacar que mais de 60% afirmam melhorias relacionadas aos requisitos do sistema. Os principais benefícios encontrados nos artigos foram:

- **Representação do conhecimento do domínio (18%)** – durante o desenvolvimento de um sistema embarcado, há muitos motivos para incluir conceitos do domínio do problema durante todo projeto. Além de tornar o sistema mais compreensível para aqueles com entendimento de domínio, torna-se possível verificar se o sistema está em conformidade com os critérios de correção expressos no domínio de interesse. As ontologias são adequadas para representar o conhecimento do domínio, facilitando o

levantamento de requisitos e, portanto, produzir especificações de requisitos com melhor qualidade.

- **Facilitar a comunicação entre os Stakeholders (22%)** – o uso da ontologia pode facilitar a comunicação entre os engenheiros, pois apresenta um vocabulário comum entre todos os envolvidos no projeto. Esse é um ponto vital no desenvolvimento desses sistemas, pois os *stakeholders* (engenheiros de hardware, software, entre outros) precisam de técnicas para facilitar a comunicação entre eles.
- **Facilitar a etapa de definição dos requisitos (37%)** – a etapa de definição de requisitos é considerada a mais crítica no processo de desenvolvimento desses sistemas, ela serve como entrada para as outras etapas. Um erro em um requisito pode comprometer todo o sistema. Portanto, o uso de ontologias é importante para padronizar e entender os requisitos, ajudando a reduzir a ambiguidade. Ontologias também ajudariam na verificação automática de erros.
- **Representação dos requisitos não-funcionais (18%)** – esses requisitos são os principais no domínio dos sistemas embarcados e precisam de uma especificação mais precisa. Contudo, são poucas as técnicas que dão prioridade a esse tipo de requisito. O uso de ontologia facilita a padronização desses requisitos e podem ser reutilizados em outros projetos.
- **Integração entre os requisitos e arquitetura (15%)** – a fase de requisitos diz apenas o que o sistema faz, e a arquitetura diz como o sistema faz. É a fase de arquitetura que descreve como o sistema implementa as funcionalidades. Portanto, o uso de ontologias facilita a integração da fase de requisitos e arquitetura, uma vez que os requisitos estão devidamente formalizados.

#### 4.7 SÍNTESE DO CAPÍTULO

Neste capítulo abordamos os dados referentes ao processo de condução da revisão sistemática da literatura. Foi possível visualizar o processo empregado na RSL, desde o planejamento até a apresentação dos resultados obtidos. Por meio dessa revisão sistemática da literatura, identificaram-se trabalhos relevantes para a temática da pesquisa.

As *strings* de busca aplicadas retornaram 1734 trabalhos, totalizando as sete bases de dados. Após todo o processo de seleção dos trabalhos, apenas 27 foram selecionados.

Por meio dessa revisão, foi constatado que existem estudos sobre ontologias para sistemas embarcados. Foi possível perceber que alguns artigos, abordavam algum tipo de iniciativa de ontologia para facilitar o desenvolvimento de sistemas embarcados. Além disso, ficou evidente a necessidade de ontologias de mais alto nível que foque na eliciação de requisitos, podendo ser convenientemente utilizada em diferentes domínios de SE.

## 5 SURVEY

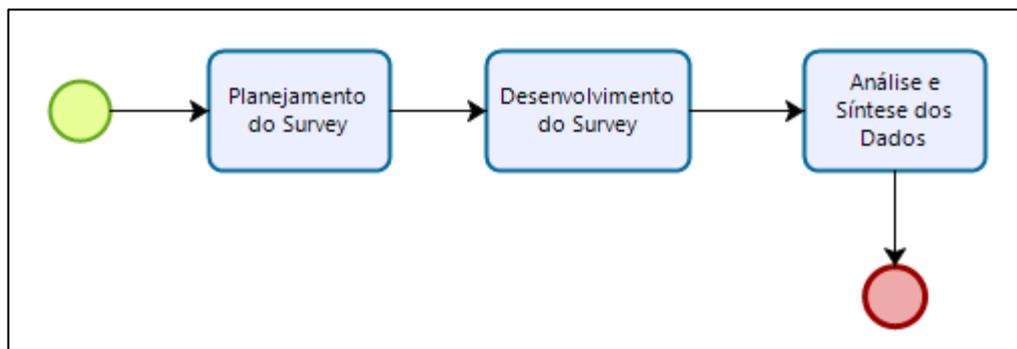
Neste capítulo, explicitam-se os aspectos metodológicos e o passo a passo de toda a execução do *Survey*, protocolo da pesquisa, entrevistas, questionário, análise e síntese dos dados e questões relativas à ameaça à validade. Através dos resultados alcançados e apresentados no presente capítulo, a seguinte questão de pesquisa pode ser respondida:

**QP02-** Como acontece a engenharia de requisitos em desenvolvimento de sistemas embarcados, e quais abordagens estão sendo utilizadas por especialistas do domínio?

### 5.1 ASPECTOS METODOLÓGICOS

Um dos métodos empíricos escolhido para realização dessa tese foi um *Survey*. Segundo Provdanov (2013), um *Survey* ocorre quando envolve a investigação direta com pessoas cujo comportamento desejamos conhecer através de algum tipo de questionário. Ele não é indicado para uma análise de problemas mais complexos, pois é considerado um método mais superficial. Portanto, foi escolhido esse método para investigar (de maneira geral) o processo de engenharia de requisitos empregado durante o desenvolvimento de um sistema embarcado. Entende-se que a finalidade de um *survey* é produzir estatísticas, ou seja, descrições quantitativas ou numéricas de alguns aspectos da população de estudo. As etapas são apresentadas na Figura 13 (PFLEEGER; KITCHENHAM, 2001):

Figura 13 –Etapas do Survey.



Fonte: Adaptado (PFLEEGER; KITCHENHAM, 2001).

Na **Etapa I – Planejamento do Survey** foi realizado o planejamento para definir o objetivo, identificar a população-alvo, as questões de pesquisas e os métodos de análise de dados. Portanto, o objetivo foi analisar como é realizado o processo de Engenharia de

Requisito no desenvolvimento de Sistemas Embarcados, identificando quais as principais abordagens utilizadas durante todas as fases da ER (elicitação, análise e negociação, especificação, validação e gerenciamento) e os principais requisitos desses sistemas.

O público-alvo deste estudo, isto é, o grupo de indivíduos a qual a pesquisa se aplica, foi de profissionais que desenvolvem sistemas embarcados. Esses profissionais foram escolhidos por facilidade de acesso, indicação, ou por alguma outra razão que se considera representativa para pesquisa. Diante desse contexto, esse método de amostragem é considerado não probabilístico porque a seleção dos elementos da população para compor a amostra depende do julgamento do pesquisador (PFLEEGER; KITCHENHAM, 2001).

Na **Etapa II – Desenvolvimento do Survey** realizou-se a coleta dos dados. O instrumento de coleta de dados utilizado foi um questionário (ver Apêndice D), elaborado através da ferramenta online do Google Forms. O questionário utilizado nesta pesquisa pode ser classificado como questionário semiestruturado, uma vez que constaram questões fechadas de múltipla-escolha, além de questões abertas.

Antes da aplicação do *survey* na amostra final da população, foi realizado um estudo piloto, com o intuito de melhor avaliar o questionário. Neste estudo piloto, a coleta de dados foi enviada a um profissional que trabalha com desenvolvimento de sistemas embarcados há mais de 5 anos.

O questionário foi enviado por e-mail (ver Apêndice C) a profissionais e pesquisadores de sistemas embarcados. No primeiro momento, o questionário foi enviado com data para ser respondido entre os dias 13/04/2018 a 17/07/2018, somente 3 profissionais responderam. Houve uma segunda chamada e este foi reenviado para os mesmos profissionais, outros por indicação e grupos de pesquisa sobre a temática de pesquisa. É importante ressaltar, que a segunda chamada ficou aberta até 05/10/2019. Foi assegurado o compromisso ético em preservar as informações coletadas através do questionário aplicado. Foram considerados como critérios de exclusão para este estudo, os sujeitos que não se enquadraram no perfil da pesquisa e aqueles que não responderam completamente o questionário. Considerando esse contexto, foram enviados cerca de 50 convites, tendo um retorno de 14, porém apenas 10 foram considerados válidos. Os 4 descartado tinham informações incompletas ou fora do escopo da temática.

Na **Etapa III – Análise e Síntese dos Dados** buscaram-se identificar como acontece o processo de engenharia de requisitos no desenvolvimento de sistemas embarcados, destacando os principais métodos utilizados e os principais requisitos que são considerados. Portanto, diante do que foi exposto sobre o *Survey*, pode-se perceber que, apesar de ser um método de

pesquisa muito difundido, não é um método simples. Existem considerações a serem feitas para uma positiva execução. As seções seguintes apresentam os resultados do *Survey*.

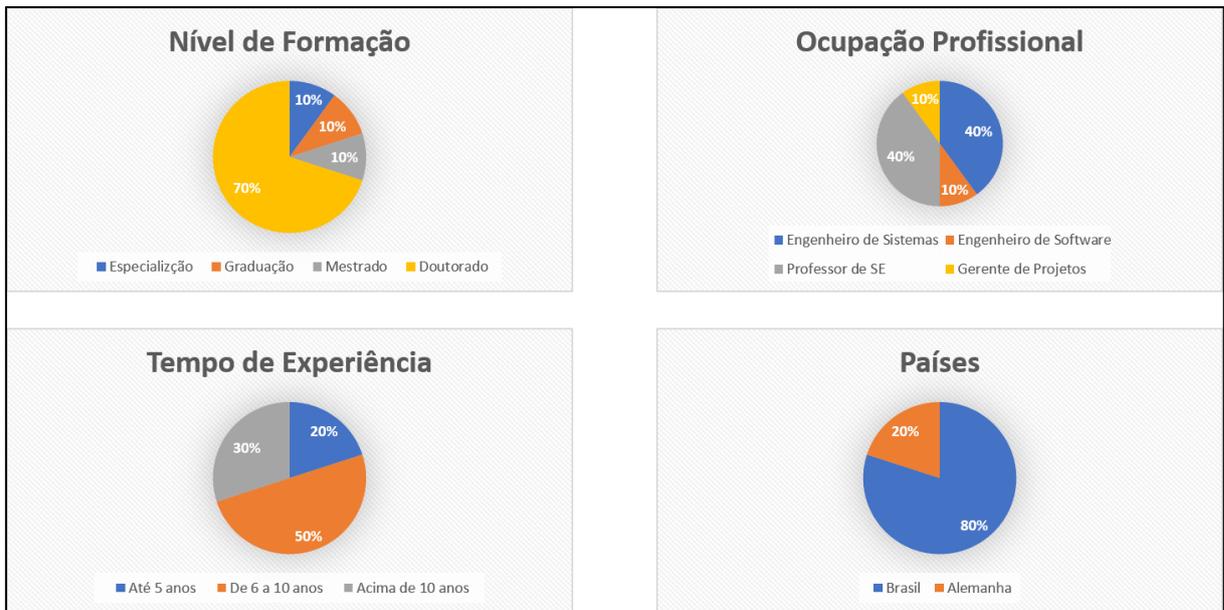
## 5.2 ANÁLISE E SÍNTESE DOS DADOS

Esta seção apresenta a análise e síntese dos dados que foram extraídos do questionário aplicado. Esse questionário foi dividido em três partes, que serão apresentadas nas subseções seguintes.

### 5.2.1 Primeira I – Características dos Respondentes

A primeira parte do questionário se propôs a identificar as características e identificação dos respondentes, conforme apresentado na Figura 14 . De início, foi possível verificar que os respondentes têm um bom nível de formação, pois 70% apresentaram o nível de doutorado. Graduação, especialização e mestrado representaram apenas 10% cada. Com relação a ocupação profissional, os respondentes têm uma forte relação com sistemas embarcados. Engenheiro de sistemas e professor de sistemas embarcados representaram 40% cada, já engenheiro de software e gerente de projetos representaram 10% cada, respectivamente. Analisando o tempo de experiência dos respondentes, a maioria disse que possui entre 6 e 10 anos de experiência (representando 50%), acima de 10 anos representou 30% e até 5 anos 20%. Isso mostra que todos os respondentes possuem um tempo consideravelmente bom de experiência e conseguem responder as questões do questionário. Por fim, com relação aos Países de atuação, a maioria estão no Brasil 80%; os demais 20% trabalham na Alemanha.

**Figura 14 – Características e Identificação dos Respondentes.**



Fonte: A Autora (2021).

### 5.2.2 Parte II – A Engenharia de Requisitos

A segunda parte do questionário se propôs a investigar sobre questões específicas da Engenharia de Requisito, no desenvolvimento desses tipos de sistemas. De início, questionamos quais os domínios de SE, se os requisitos são levantados no início do projeto, quem era o responsável e se os clientes participavam dessa definição. Conforme apresentado na Figura 15, o domínio de Automação foi o mais representado (45%, 8 respostas), seguido pelo domínio Automotivo (22%, 4 respostas), Médico, Aviônico e Energia (11%, ou 2 respostas cada, respectivamente). É importante destacar que os respondentes podiam marcar mais de um domínio.

Foi questionado se os requisitos são levantados no início do projeto. Todos responderam que **Sim** (100%). Essa resposta foi bem representativa para o contexto da pesquisa, pois mostrou que todos estão preocupados com a definição de requisitos logo no início dos projetos. A definição dos requisitos é realizada por engenheiro de sistemas ou software (40% das respostas), ou por todos os membros da equipe (30% das respostas), ou ainda por engenheiros de requisitos (30% das respostas). E, por fim, questionamos se o cliente participa dessa etapa. O resultado foi positivo, pois 100% das respostas foram **Sim**. Isso confirma a importância de incluir o cliente em todas as etapas da engenharia de requisitos, segundo que alguns autores (SOMMERVILLE; KOTONYA, 1998) (PRESSMAN; MAXIM,

2016) ratificam que incluir o cliente nesse processo diminui os possíveis erros e a má especificação dos requisitos. De fato, todos os requisitos refletem exatamente a necessidade dos clientes e qualquer mudança ou inclusão desses precisam de uma aprovação deste.

**Figura 15 – Definição dos Requisitos.**

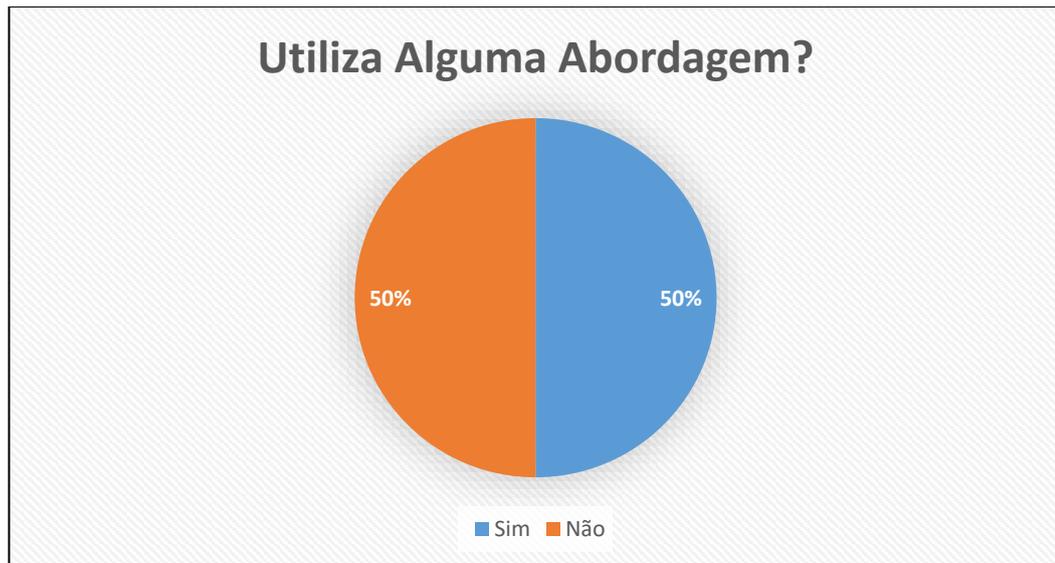


**Fonte: A Autora (2021).**

Ainda na segunda parte do questionário, investigou-se dados sobre o processo de engenharia de requisitos, levando em consideração todas as etapas definidas (elicitação, análise/negociação, especificação, validação e gerenciamento). Na primeira questão, foi perguntado se os respondentes utilizam alguma abordagem para engenharia dos requisitos. É importante destacar, que a palavra abordagem, no contexto deste *Survey*, pode ser uma técnica, método, modelo, processo, ferramenta, linguagem, entre outros. Conforme mostrado na Figura 16, o resultado foi de que 50% utilizam e 50% não utilizam. Aos que responderam **Não**, foi perguntado qual era o motivo da não utilização. A maioria dos respondentes (80%) disseram que é a falta de conhecimento de boas abordagens que consigam contemplar requisitos de *hardware* e *software*. Ademais, apenas 1 respondente (20%) afirmou que é a dificuldade que os engenheiros encontram na utilização. Realizando um cruzamento de dados entre os resultados apresentados na ocupação profissional com essa questão da abordagem, percebemos que 60% ocupam cargos como engenheiros de sistemas (profissionais mais específicos da área de *hardware*). Portanto, uma explicação pode ser porque profissionais

específicos dessa área de *hardware* não tiveram disciplinas específicas, durante o processo de formação, de engenharia de requisitos. O foco maior é em disciplinas de desenvolvimento, com isso eles focam mais nessa etapa e negligenciam as etapas da engenharia de requisitos. No que concerne aos que responderam **Sim**, a maior parte ocupa cargos voltados para requisitos (60%).

**Figura 16 – Utilização de Abordagem.**



**Fonte: A Autora (2021).**

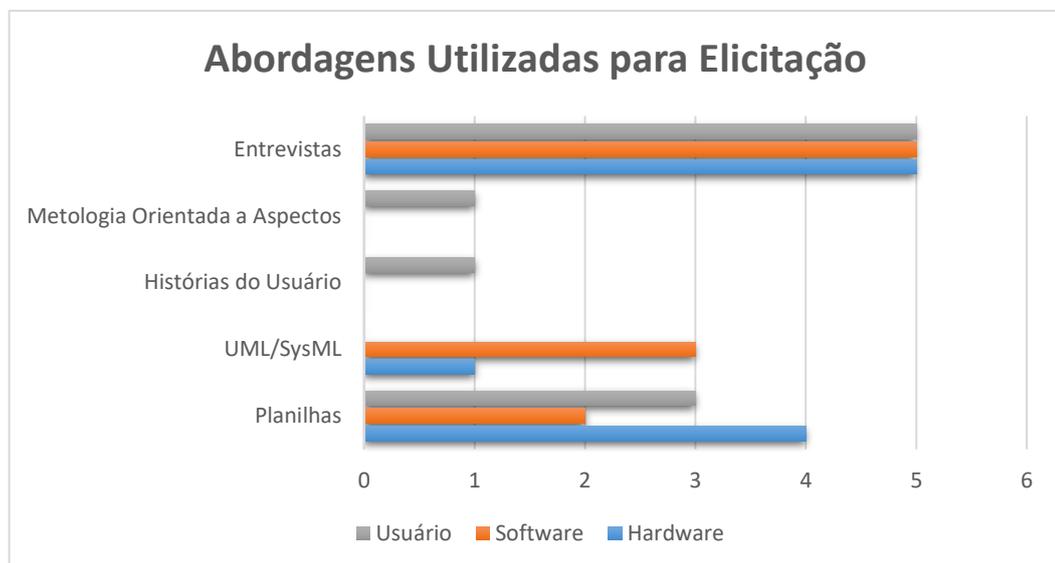
Diante do resultado apresentado na Figura 16, apenas os que responderam sim (5 respostas) tiveram acesso as questões específicas sobre as etapas da ER.

### **A. Elicitação de Requisitos**

Segundo Sommerville (2011), a elicitação é a primeira etapa, seu principal objetivo é obter informações relevantes para o desenvolvimento do sistema. Algumas questões de pesquisa foram definidas para entender como acontece a elicitação de requisitos no desenvolvimento dos SE. O objetivo foi identificar quais as principais abordagens utilizadas para elicitação dos requisitos de *hardware*, *software* e usuário. Os respondentes podiam apresentar mais de uma abordagem. De acordo com a Figura 17, percebemos que as entrevistas foram citadas por todos os respondentes (100%) para todos os tipos de requisitos, e foram associadas com outras abordagens. As entrevistas são uma abordagem sistemática para obter informações de uma pessoa ou grupo de pessoas. A explicação porque ela é a técnica mais utilizada para elicitação de requisitos, pode surgir do fato de que ela consegue explorar ou esclarecer os requisitos com mais detalhes.

Com relação às outras abordagens utilizadas para *hardware*, a maior parte utiliza planilhas (4 respostas) e apenas um utiliza modelagem UML/SysML. Na perspectiva de software, a maioria utiliza modelagem UML/SysML (3 respostas) e dois utilizam planilhas. Para elicitação dos requisitos de usuário são utilizadas histórias do usuário (1 resposta), metodologia orientada a aspectos (1 resposta) e planilhas (3 respostas). Percebemos que os requisitos de *hardware* são os que mais utilizam planilhas. Esse resultado pode ser explicado pelo fato da maioria das abordagens de engenharia de requisitos serem voltadas para requisitos de *software*, deixando muitas particularidades do *hardware* de lado. Dessa forma, elaborar e/ou utilizar abordagens que contemple todos os tipos de requisitos é mais satisfatório para o desenvolvimento desses sistemas.

**Figura 17 – Abordagens para Elicitação.**



**Fonte: A Autora (2021).**

## **B. Análise e Negociação**

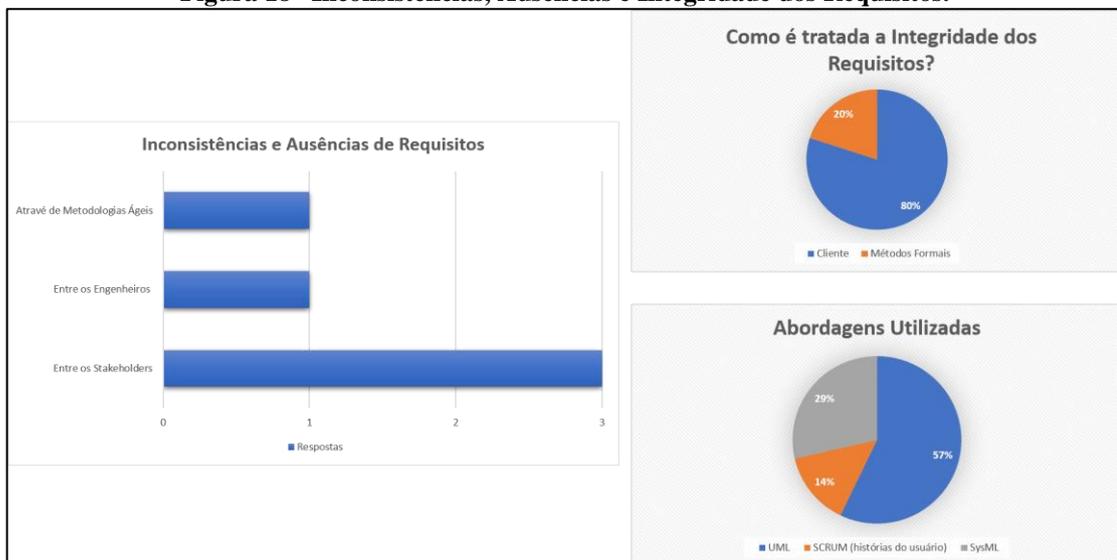
Nesta etapa é realizada a análise das necessidades dos usuários e clientes para atingir uma definição precisa dos requisitos (SOMMERVILLE; KOTONYA, 1998). Essa etapa é considerada crítica, pois verifica se existem requisitos incompletos, ambíguos e potenciais conflitos entre eles. Foram definidas 4 questões com o objetivo de identificar como acontece a análise e negociação de requisitos, se utiliza alguma abordagem e como é realizada a priorização. A primeira pergunta se propôs a responder: Como são analisadas as ausências de requisitos e inconsistências entre eles?. A Figura 18 apresenta as respostas dessa questão. Analisando-as, percebe-se que não existe uma formalização para tratar ausências de requisitos e as inconsistências entre eles; apenas um respondente (representando 20%) utiliza algum tipo

de metodologia (ágeis através de história do usuário). A maior parte (60% dos entrevistados) realiza essa análise através dos *stakeholders*. Isso pode representar um risco para o projeto, pois os *stakeholders* não sabem o que realmente quer, expressam requisitos em seus próprios termos e diferentes *stakeholders* podem ter requisitos conflitantes.

Em outra questão, foi possível identificar como eles tratam a integridade dos requisitos. Como resultado, 80% das respostas utilizam a aprovação do cliente para definir se os requisitos estão íntegros ou não, pois estes devem refletir exatamente a necessidade do cliente. Apenas um trabalho (representando 20%) utiliza métodos formais para provar que realmente os requisitos estão íntegros, conforme apresentado na Figura 18.

Com relação as abordagens utilizadas, foram citadas UML (57%), SysML (29%) e histórias do usuário (14%). Os respondentes podiam citar mais de uma abordagem. A UML é uma linguagem de modelagem para software, e algumas particularidades do *Hardware* não são possíveis de serem modeladas. Diante disso, surgiu a SysML, que é uma linguagem de modelagem voltada para sistemas (contempla os requisitos de *Hardware* e *Software*). É importante destacar, que as abordagens adotadas nessa etapa (análise e negociação) são bem representadas por pesquisadores e engenheiros desses tipos de sistemas.

**Figura 18– Inconsistências, Ausências e Integridade dos Requisitos.**

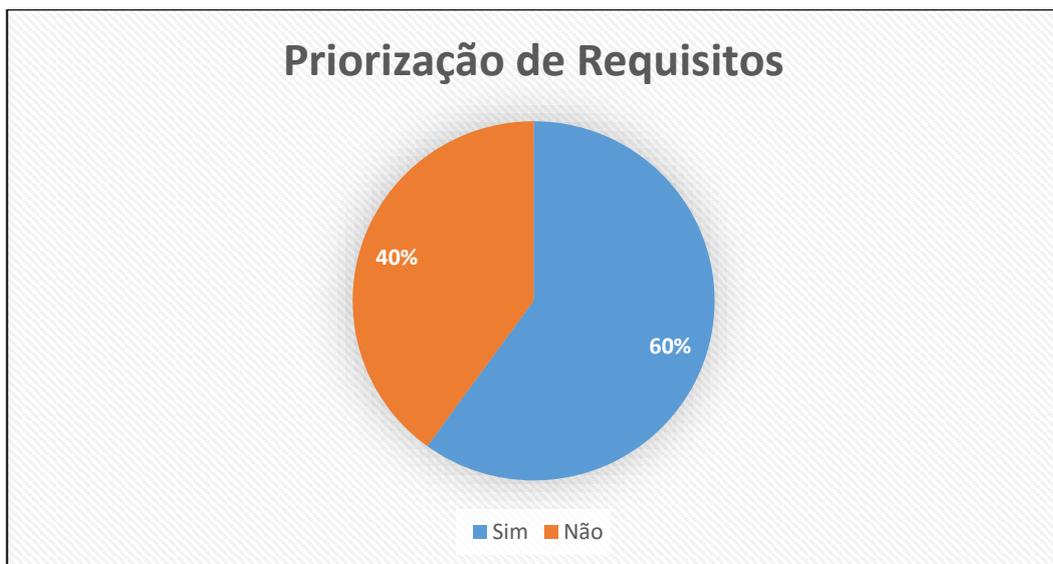


Fonte: A Autora (2021).

Por fim, foi perguntado se era utilizado algum procedimento para realizar a priorização dos requisitos. Segundo Berander (2005), a priorização de requisitos é essencial para que qualquer projeto seja bem executado. A priorização garante que o projeto se concentre nos requisitos mais importante primeiro, e que todos possam entender e concordar.

Nesse *survey*, 60% dos respondentes estão utilizando priorização de requisitos e 40% não usam. Porém, quando questionado sobre as técnicas utilizadas, nenhum citou alguma técnica específica para priorização. Todos afirmaram que a equipe ou um engenheiro específico decide quais são os requisitos mais críticos e priorizam. O interessante seria que alguma técnica específica para priorização seja utilizada.

**Figura 19– Priorização de Requisitos.**



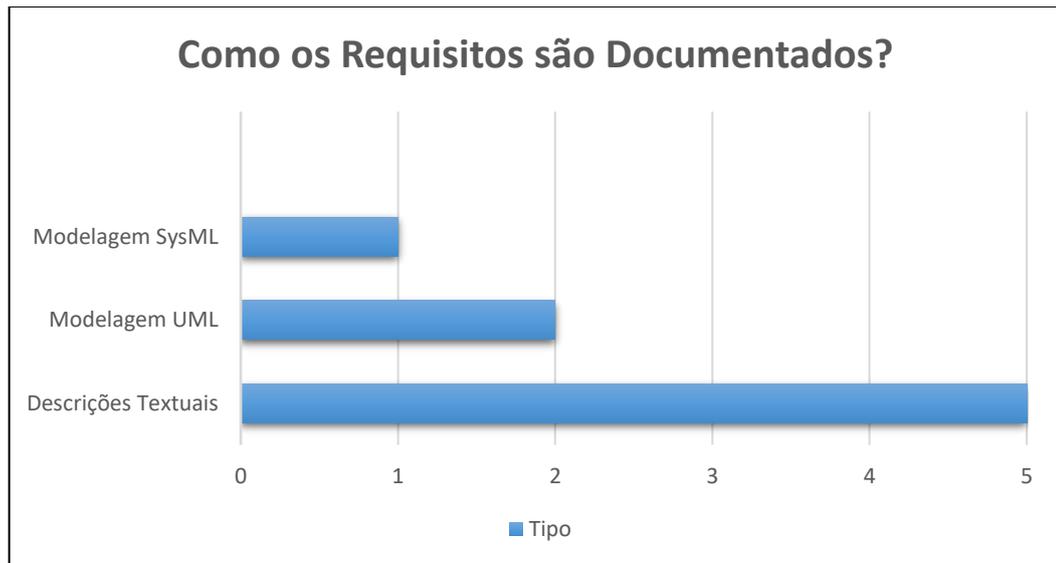
**Fonte: A Autora (2021).**

### **C. Especificação**

Na etapa de especificação de requisitos, foram definidas 4 questões. Essa etapa corresponde a descrição detalhada do sistema que será desenvolvido, incluindo todos os seus requisitos (SOMMERVILLE; KOTONYA, 1998). O documento de especificação de requisitos deve ser consistente, não ambíguo e de fácil entendimento. Principalmente em projetos de sistemas embarcados, visto que a equipe envolve pessoas de *hardware* e *software*. A primeira questão teve como objetivo identificar como os requisitos são documentados. A Figura 20 mostra que todos os respondentes (100%) utilizam descrições textuais para documentar esses requisitos. Embora a linguagem natural seja ambígua, é imprescindível uma descrição textual dos requisitos. Elas completam os modelos, pois esses apresentam os requisitos apenas de uma maneira visual. Contudo, mesmo que a linguagem natural seja imprescindível, é necessária atenção na descrição dos requisitos, porque existem problemas como falta de clareza, confusão e inconsistência entre requisitos. Dentre os 5 (cinco) respondentes, três utilizam alguma técnica de modelagem para complementar a descrição

textual. As linguagens de modelagem citadas foram UML (2 respostas) e SysML (1 resposta). Na engenharia de requisitos é necessário identificar e implementar diferentes requisitos que podem estar espalhados pelos artefatos de ER. Portanto, é importante conhecer como esses requisitos serão representados na modelagem.

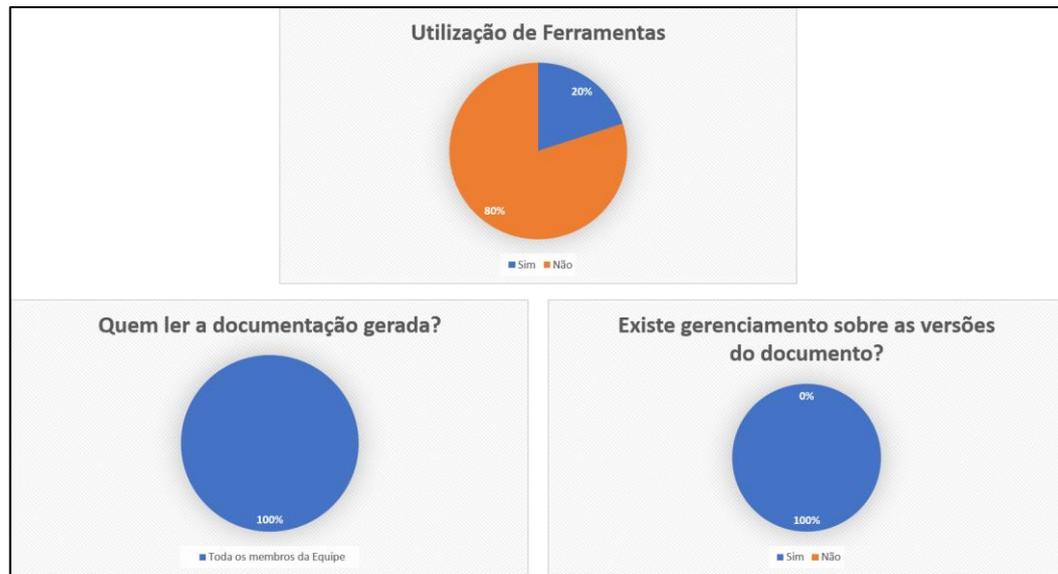
**Figura 20– Documentação dos Requisitos.**



**Fonte: A Autora (2021).**

A Figura 21 apresenta as respostas sobre utilização de ferramentas para auxiliar a especificação, quem lê a documentação gerada e se existe algum tipo de gerenciamento sobre as versões do documento. No que concerne a utilização de ferramentas, apenas 20% (1 respondente) afirmou que utiliza uma ferramenta conhecida como Tamino. O maior número de respondentes (80%) não utiliza nenhuma ferramenta. Em seguida, sobre a leitura da documentação gerada, 100% afirmaram que todos os membros da equipe realizam a leitura desse documento. Portanto, como a equipe tem pessoas de *hardware* e *software*, é necessário que a documentação seja clara e sem ambiguidades para ambas as partes. Por fim, foi perguntado se existe um gerenciamento sobre as versões que são geradas desse documento e todos responderam que **Sim** (100%).

**Figura 21– Ferramentas e Gerenciamento de Versões.**



**Fonte: A Autora (2021).**

#### **D. Validação**

A etapa de validação de requisitos tem como objetivo garantir que os requisitos especificados atendam às necessidades do cliente. Ela está preocupada em encontrar problemas com os requisitos (SOMMERVILLE; KOTONYA, 1998). Os custos com os erros na identificação dos requisitos são elevados, além de comprometer todo sistema. Diante disso, é mais vantajoso corrigir um erro de requisitos logo no início do projeto do que na instalação do sistema. Isto é necessário principalmente em sistemas embarcados, já que um erro em um requisito pode custar a vida de uma pessoa. As questões dessa etapa se propuseram a responder se era realizada a validação de requisito e como era realizada.

Observa-se, na Figura 22, que todos os respondentes (100%) afirmaram que realizam a validação dos requisitos. Ela é realizada pelos *stakeholders* (2 respondentes), por método formais (1 respondente), clientes (1 respondente), e *Product Owner* (1 respondente). Percebe-se, que apenas um respondente apresentou um tipo de abordagem (método formais), o restante informou que uma ou mais pessoas realiza essa etapa. Acredita-se que esse resultado seja porque existe pouco conhecimento, por parte dos responsáveis por esses sistemas, com relação as técnicas que podem ser utilizadas para essa etapa. Os sistemas embarcados são considerados críticos, e para satisfazer as necessidades do cliente e/ou usuários é importante que esses requisitos sejam validados, senão podem deixar de lado erros de ambiguidades, incompletudes e contradições. Esses erros podem levar a custos de retrabalho extensivos e depois que o sistema está implementado pode ocorrer erros mais críticos.

**Figura 22– Validação de Requisitos.**



**Fonte: A Autora (2021).**

## **E. Gerenciamento**

As mudanças em requisitos durante o ciclo de vida do sistema fazem parte do processo. Necessidades não encontradas inicialmente, alterações nos requisitos e correções de erros são alguns motivos que destacam a importância do gerenciamento dos requisitos. Nesse contexto, essa etapa refere-se ao conjunto de atividades que possibilitam a equipe do projeto a identificar, controlar e rastrear esses requisitos, bem como as suas alterações, ou seja, é o processo que monitora as mudanças nos requisitos do projeto (SOMMERVILLE; KOTONYA, 1998). É nessa etapa que acontece o processo de rastreabilidade dos requisitos – ou seja, o processo para prover relacionamentos entre os requisitos.

**Portanto, foi questionado se é realizado o gerenciamento de requisitos e se utilizam alguma ferramenta.**

**Conforme apresentado na**

Figura 23, a maior parte faz o gerenciamento (80% - 4 respondentes) e apenas 20% (1 respondente) não faz. Com relação a utilização de ferramentas, 75% não utiliza ferramentas para auxiliar esse processo de gerenciamento e apenas 25% (1 – respondente) citou que utilizava a ferramenta Tamino. Observou-se que essa ferramenta Tamino é utilizada em algumas etapas da ER e nela realiza-se também a rastreabilidade dos requisitos. Fazendo um cruzamento entre os domínios de SE e a utilização de ferramenta para etapa de gerenciamento, foi possível observar que a Tamino está sendo utilizada para domínios automotivos e aviônicos. Esses tipos de sistemas são mais críticos e precisam lidar com requisitos mais complexos, então o uso de uma ferramenta pode facilitar o gerenciamento dos relacionamentos entre os requisitos, analisar o impacto de quaisquer alterações e gerenciar aprovações.

**Figura 23– Gerenciamento dos Requisitos.**



Fonte: A Autora (2021).

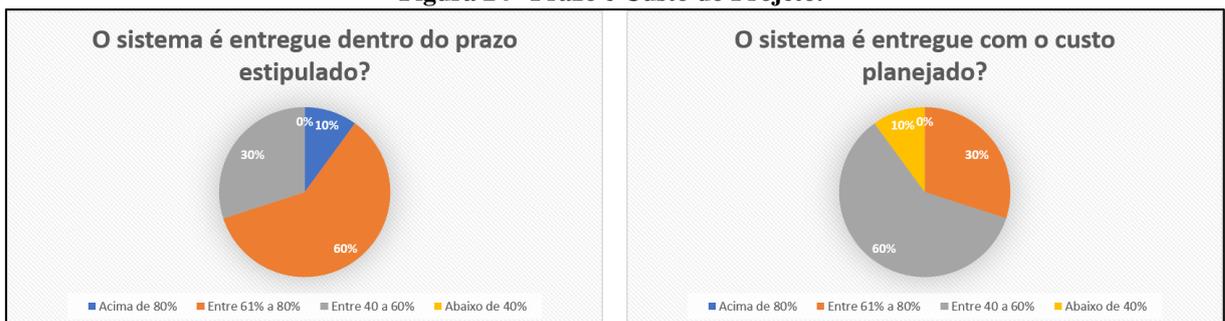
### 5.2.3 Parte III – Requisitos de Hardware e Software

A terceira parte do questionário teve como objetivo levantar requisitos específicos de *hardware* e *software* para o desenvolvimento de sistemas embarcados, se o sistema é entregue dentro do prazo estipulado e se o sistema é entregue com o custo planejado inicialmente.

Com relação aos requisitos levantados, eles são mencionados no 6. Esses requisitos foram utilizados para compor a ontologia proposta nesta tese. Observando a Figura 24, percebemos que 60% (6 respostas) dos projetos são entregues entre 61% a 80% do prazo estipulado, 30% (3 respostas) entre 40% e 60% do prazo estipulado e 10% (1 resposta) acima de 80% do prazo estipulado. Analisando esses resultados, é notável que muitos dos projetos não estão sendo entregues dentro do prazo estipulado, ou seja, de 10 projetos realizados, apenas 1 fica na margem Acima de 80%.

Com relação ao custo do projeto, 60% (6 respostas) entregam **entre 40% a 60%**, 30% **entre 61% e 80%** (3 respostas) e 10% (1 resposta) abaixo de **40%**. Percebe-se que, existe um número alto de projetos que não são entregues com os custos que foram planejados no início. Entre 10 projetos realizados, apenas 3 ficam na margem de **entre 61% a 80%**.

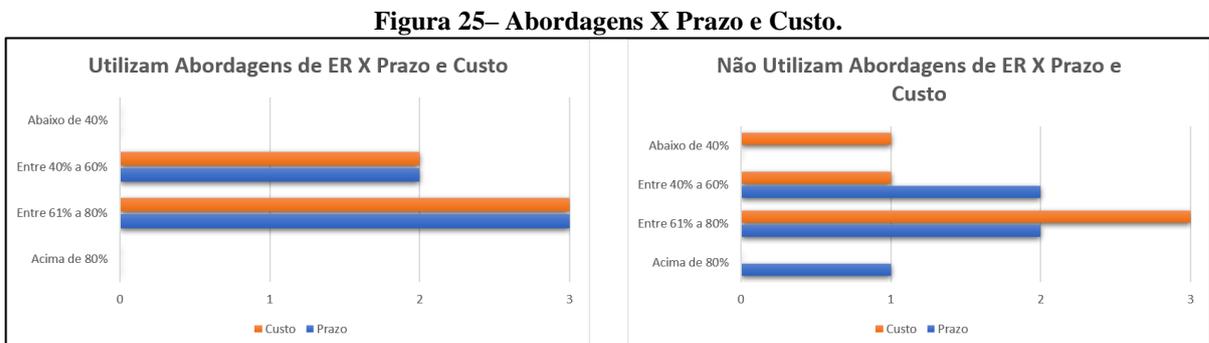
**Figura 24– Prazo e Custo do Projeto.**



Fonte: A Autora (2021).

Diante dos resultados apresentados anteriormente, foi realizado alguns cruzamentos de dados para entender quais os possíveis motivos dos projetos não estarem sendo entregues no prazo e pelos custos bem maiores do que o planejado. A Figura 25 apresenta os cruzamentos realizados. Foi dividido entre os que utilizam algum tipo de abordagem e os que não utilizam (o resultado da abordagem foi apresentado na Figura 16). Analisando os resultados, percebemos que a maior parte que utiliza algum tipo de abordagem, para auxiliar o processo de ER, estão em uma margem de entrega do projeto no prazo estipulado considerada razoavelmente boa (entre 61% a 80%). Com relação aos custos, apresentaram os mesmos resultados.

Analisando os que não utilizam qualquer tipo de abordagem, percebe-se que a maioria (3 respondentes) estão na margem **entre 61% a 80%** que não cumpre os gastos acordados no início do projeto. Esse número é significativo, principalmente quando falamos em sistemas embarcados, visto que muitos desses sistemas tem um custo elevado. Apenas um respondente citou a margem **Abaixo de 40%**. Talvez a dificuldade em estourar os custos dos projetos seja o fato de não utilizar alguma abordagem para auxiliar o processo de ER, ele acontece de qualquer forma, pois estão mais preocupados com o desenvolvimento. No que concerne o prazo de entrega, apenas 1 (um) respondente citou que estão **acima de 80%**, 2 respondentes **entre 61% a 80%** e o restante **entre 61% e 80%**.



**Fonte: A Autora (2021).**

### 5.3 AMEAÇAS À VALIDADE

A questão fundamental a respeito dos resultados dos métodos de pesquisa é quão válidos são eles (TRAVASSOS; GUROV; AMARAL, 2002). Assim, os resultados devem ser válidos para a população da qual o conjunto de participantes foi recebido. É importante também para generalizar os resultados para uma população mais ampla. Diante disso, os resultados possuem a validade adequada se são válidos para a população a qual tendem ser generalizados.

Travassos (2002) define 4 (quatro) tipos de validade para os resultados de métodos de pesquisa: validade de conclusão, validade interna, validade de construção e validade externa.

A validade de construção considera os relacionamentos entre a teoria e a observação, ou seja, se o tratamento reflete bem a causa e se o resultado reflete bem o efeito (TRAVASSOS; GUROV; AMARAL, 2002). No contexto desse *Survey*, temos a definição do questionário como uma validade de construção. Existem considerações que precisam ser realizadas em relação às possíveis ameaças à validade do questionário, pois é necessário garantir que ele meça exatamente o que se propõe a medir. Uma das ameaças a validade está relacionada ao viés do pesquisador. É preciso garantir a coerência das perguntas evitando ambiguidade e imprecisão delas, bem como avaliar se o tipo das respostas é adequado às perguntas elaboradas. Diante disso, uma estratégia foi utilizada para minimizar possíveis vieses. Primeiro, conforme descrito na seção 5, houve um teste piloto do questionário, ele foi submetido a um especialista do domínio de sistemas embarcados para uma primeira validação. Depois dessa validação, foram realizados alguns ajustes e acréscimos de questões.

A validade interna define se o relacionamento observado entre o tratamento e o resultado é causal, ou seja, não é resultado da influência de outro fator (não controlado ou medido) (TRAVASSOS; GUROV; AMARAL, 2002). Com relação a validade interna, neste *Survey*, está relacionada aos problemas que podem surgir com os participantes do estudo, em relação ao questionário que foi aplicado ou respostas irrelevantes que possam levar a erros no momento da síntese dos dados. Para minimizar esses problemas foi elaborada uma carta convite (APÊNDICE C), explicando o contexto da pesquisa, ajudando assim, que só participassem pessoas relacionadas à temática da pesquisa. Outra medida adotada foi que o questionário possuía informações sobre o contexto da pesquisa, conceitos abordados, e informações que os participantes pudessem precisar para responder determinadas perguntas.

Segundo Travasso (2002), a validade de conclusão considera as possíveis conclusões errôneas retiradas dos resultados. Diante disso, foram adotadas estratégias de análise e síntese de dados para mitigar essas ameaças.

Por fim, a validade externa define condições que limitam a habilidade de generalizar os resultados de um experimento para o contexto da indústria (TRAVASSOS; GUROV; AMARAL, 2002). Quanto à validade externa, a principal ameaça está relacionada ao tamanho da amostra utilizada neste trabalho. O método de amostragem utilizado foi o não probabilístico. Esse método não define o tamanho da amostra, pois a representatividade da população é praticamente impossível de medir (PFLEEGER; KITCHENHAM, 2001). A superioridade da amostragem probabilística é incontestável, mas existem momentos em que o uso da

amostragem não probabilística deve ser considerado. O *Survey* descrito nesta tese é um desses casos, pois é praticamente impossível medir a população do estudo (profissionais de sistemas embarcados).

#### 5.4 SÍNTESE DO CAPÍTULO

Neste capítulo abordamos os dados referentes ao processo de condução do *Survey*. O objetivo deste foi analisar como acontece a Engenharia de Requisitos no desenvolvimento de Sistemas Embarcados. Foi possível visualizar todo processo empregado para execução dele, desde o planejamento, desenvolvimento e análise e síntese dos dados. Diante do exposto, neste *Survey*, é possível identificar os benefícios da condução do processo de Engenharia de Requisitos no desenvolvimento de SE.

## 6 ONTOLOGIA DE REQUISITOS PARA SISTEMA EMBARCADO

Este capítulo descreve uma ontologia *core* para sistemas embarcados (OntoREES) e uma ontologia de domínio para sistemas aviônicos. Ambas as ontologias foram construídas a partir da ontologia de topo UFO, o que possibilita avaliar a modelagem conceitual por meio de primitivas e restrições de modelagem estereotipadas. Essas ontologias foram construídas a partir dos resultados obtidos pela RSL (**Erro! Fonte de referência não encontrada.**), pelo *Survey* com especialistas em Sistemas Embarcados (5) e através de entrevistas com especialistas do domínio de aviônicos e automotivos. No decorrer deste capítulo, buscamos responder a seguinte questão de pesquisa:

**“Como é possível formalizar o conhecimento dos requisitos de sistemas embarcados, através de uma notação Ontológica que possa ser usada e reutilizada para simular a definição de requisitos, por intermédio da lógica descritiva?”**

### 6.1 CONSTRUÇÃO DA ONTOLOGIA

O processo de desenvolvimento de ontologias segue uma metodologia similar utilizada para o desenvolvimento de sistemas, ou seja, etapas sistemáticas que tornam o processo mais dinâmico e um resultado final mais satisfatório. Porém, um dos grandes problemas com o desenvolvimento de ontologias é que a padronização de um processo para construção de uma ontologia ainda é incipiente. Muitas ontologias foram desenvolvidas por diferentes comunidades, utilizando abordagens, métodos e técnicas diferenciadas, uma vez que ainda não há consenso sobre uma metodologia para o processo de construção dessas ontologias (CORCHO; FERNÁNDEZ-LÓPEZ; GÓMEZ-PÉREZ, 2006). Diante disso, neste trabalho, escolhemos a Methontology para o desenvolvimento da OntoREES, por ser utilizada e recomendada pela *Foundation for Intelligent Physical Agents* (FIPA)<sup>6</sup>, e por já ter sido indicada como metodologia madura com relação a outras utilizadas. Além disso, ela foi a mais citada nos artigos da RSL.

Com base nos resultados da RSL e do *Survey*, foi desenvolvida uma ontologia central de requisitos para sistemas embarcados, a OntoREES (Ontologia de Requisitos para Sistemas Embarcados). Com os resultados dessas revisões sistemáticas, percebemos que nenhum

---

<sup>6</sup> <http://www.fipa.org/>

estudo existente abrange os principais conceitos do domínio de um SE em uma ontologia. A falta desse modelo conceitual dificulta a representação do conhecimento do domínio. A ontologia representa uma descrição mais aproximada do conhecimento e, diferentemente da linguagem natural, ela pode ser escrita em linguagem formal (PEASE, 2011).

A escolha da ontologia para representação do conhecimento do domínio é que ela fornece uniformidade aos conceitos e termos quanto à sintaxe e semântica, facilita a comunicação entre os envolvidos no projeto e, conseqüentemente, melhora a compreensão dos requisitos. Como já mencionado, um fator importante no desenvolvimento de um sistema embarcado é a comunicação entre os envolvidos no projeto (engenheiros de *hardware* e *software*), pois precisam de técnicas que facilitem a comunicação (SOUSA, 2016). Se não há um consenso entre os conceitos do domínio, conseqüentemente gera requisitos ambíguos e inconsistentes, além de outras anomalias. Assim, a ontologia ajudará a fase de definição de requisitos no domínio de Sistemas Embarcados. Por essa razão, a denominamos OntoREES, uma ontologia para definição de requisitos em SE.

As etapas sugeridas pela Methontology (FERNÁNDEZ-LÓPEZ, 2004) e que foram adotadas para a construção da OntoREES são: especificação, aquisição do conhecimento, conceitualização, formalização, integração, implementação, avaliação, documentação e manutenção.

### 6.1.1 Especificação

Essa fase tem como objetivo a elaboração de um documento, utilizando a linguagem natural, contendo informações como o principal objetivo da ontologia e seus demais propósitos.

Em particular, no âmbito do desenvolvimento de sistemas embarcados, o objetivo basilar dessa ontologia é ajudar os engenheiros de *hardware* e *software* a descobrirem requisitos durante a atividade de elicitação, bem como especificar os requisitos elicitados.

Algumas Questões de competência (QC) para o domínio de sistemas embarcados foram criadas, para ajudar na especificação de requisitos e delimitação do escopo da Ontologia. A seguir, listamos as QC.

QC1 – O que é um sistema embarcado?

QC2 – Quais os tipos de recurso podam estar presentes em um sistema embarcado?

QC3 – Quais os tipos de requisitos de um sistema embarcado?

- QC4 – Quais os tipos de requisitos não funcionais de um sistema embarcado?
- QC5 – Quais os tipos de *stakeholders* responsáveis por um requisito?
- QC6 – O que identifica um requisito funcional?
- QC7 – O que identifica um requisito não funcional?
- QC8 – O que identifica um requisito não funcional de qualidade?
- QC9 – O que identifica um requisito não funcional físico?
- QC10 – O que identifica um requisito não funcional de medição?
- QC11 – Quais requisitos são conflitantes entre si?
- QC12 – Quais tipos de conflitos podem existir na elicitação de requisitos?
- QC12 – Quais requisitos são dependentes de outros requisitos?
- QC13 – Um requisito pode ser um funcional e não funcional no mesmo projeto?
- QC14 – O que é uma norma no contexto de desenvolvimento de SE?

### 6.1.2 Aquisição do Conhecimento

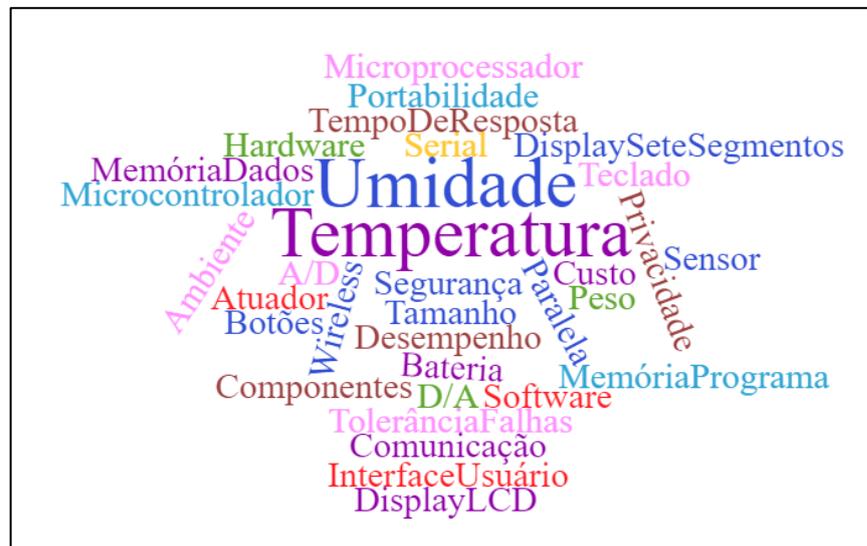
A etapa de aquisição busca as possíveis fontes de conhecimentos, tais como: entrevistas com especialistas do domínio, consulta a livros, artigos, ontologias já existentes, taxonomia, entre outros. Apesar de ocorrer com mais intensidade na fase inicial da engenharia ontológica, deve estar presente em todas as outras fases da construção da ontologia. Para compor a aquisição do conhecimento da OntoREES, utilizamos os artigos encontrados na RSL (apresenta no **Erro! Fonte de referência não encontrada.**), o *Survey* (apresentado no REF\_Ref48813750 \r \h 5), artigos de uma RSL sobre a engenharia de requisitos no desenvolvimento de sistemas embarcados (PEREIRA et al., 2017) e um MSL sobre ontologias no desenvolvimento de SE (SOUSA et al., 2016).

Nesta tese, foi adotada a estratégia *middle out* para a aquisição do conhecimento da OntoREES. Conforme discutido na Seção 3.2.4, existem três tipos de estratégias (*top down*, *bottom up* e *middle out*). A abordagem *middle out* foi utilizada neste trabalho, pois possibilita um equilíbrio em termo de nível de detalhes. O objetivo dessa abordagem é identificar conceitos centrais do domínio do conhecimento (requisitos de sistemas embarcados); em seguida, esses conceitos são generalizados e especializados para produzir a ontologia. Segundo Corcho et al. (2006), essa abordagem tende a encorajar o surgimento de campos temáticos e aumentar a modularidade e estabilidade dos resultados. É importante destacar que, durante a etapa de aquisição de conhecimento, dois especialistas do domínio de sistemas embarcados ajudaram na definição dos conceitos propostos na OntoREES.

### 6.1.3 Conceitualização

É considerada como a principal fase para construção da ontologia. Trata da estruturação do domínio do conhecimento em um modelo conceitual que descreve o problema e sua solução, em termos do vocabulário de domínio. A primeira atividade realizada foi a construção de um glossário de termos. Esse glossário reúne o conhecimento de domínio útil e potencialmente utilizável e seus significados. A Figura 26 apresenta a nuvem de palavras com os principais conceitos da OntoREES. Essa nuvem foi gerada pela ferramenta StArt, a partir dos principais conceitos que apareceram na RSL.

Figura 26 – Nuvem de Palavras.



Fonte: A Autora (2021).

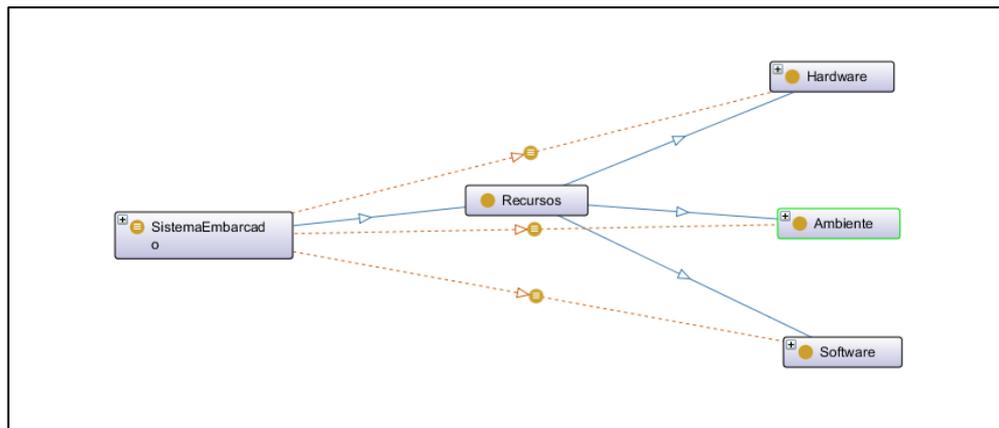
### 6.1.4 Formalização

A OntoREES é uma ontologia central que compreende uma definição formal dos requisitos para sistemas embarcados. A axiomatização é ilustrada através da notação OntoUML2, e formalizada por axiomas na lógica descritiva. A OntoUML2 é uma extensão da UML para modelagem de ontologias a partir da UFO. Na OntoUML2, as regras sintáticas evoluíram com base no feedback de usuários e em análises de como a linguagem era usada na prática (GUIZZARDI et al., 2021) (FONSECA et al., 2019). Diante disso, os conceitos são considerados como classes UML, as relações assumem o papel de herança e associações, e as instâncias são descritas por meio de diagramas de objetos. Para simplificar a apresentação da ontologia, a OntoREES será dividida em dois módulos. No primeiro módulo são apresentadas as três classificações principais adotadas neste trabalho e os agentes envolvidos, enquanto

que, no segundo módulo, são apresentados os requisitos. A ontologia foi classificada em três grandes grupos: **Hardware**, **Software** e **Ambiente**. A classe **SistemaEmbarcado** é uma classe definida (*Defined Class*), pois existem condições necessárias e suficiente que determina que qualquer indivíduo que seja **Hardware**, **Software** e **Ambiente** pertence a **SistemaEmbarcado**.

Figura 27 apresenta a ontologia com as três classificações.

Figura 27 – Ontologia com as três classificações.



Fonte: A Autora (2021).

Inicialmente, construiu-se a classe **SistemaEmbarcado**. Essa classe está relacionada com as classes **Ambiente**, **Software** e **Hardware**, através da propriedade “temRecurso”, ou seja, um “**SistemaEmbarcado tem um Recurso Ambiente**”, um “**SistemaEmbarcado tem um Recurso Software**” e um “**SistemaEmbarcado tem um Recurso Hardware**”.

#### 6.1.4.1 Classificações dos Recursos e Agentes

A Figura 28 apresenta o módulo referente aos Recursos e Agentes que podem estar presentes no projeto de definição dos requisitos de um sistema embarcado. A partir das entidades da UFO-C *Agent*, foram definidos os conceitos genéricos para o domínio de sistemas embarcados. A UFO-C é dividida em duas categorias *Social Agent* (**Organização**) e *Physical Agent* (**Pessoa**). Diante disso, com relação aos agentes presentes na definição de requisitos, quatro papéis foram definidos. Dois papéis são especializações de **Stakeholder e Pessoa**, ou seja, aqueles responsáveis pela definição dos **Requisito** de Hardware (**StakeholderHardware**) e os responsáveis pela definição dos **Requisito** de Software

(*StakeholderSoftware*). Por fim, dois papéis são especializações de *Stakeholder* e *Organizacao*, o *StakeholderTime* abordando os grupos constituídos com um propósito em uma organização ou projeto (exemplos: equipe de teste, equipe de engenharia de requisitos, equipe de codificação), e *StakeholderOrganizacao* expressando organizações como clientes, parceiros e fornecedores. Os agentes *Stakeholder*, *StakeholderTime* e *StakeholderOrganizacao* foram reutilizados e adaptados da ontologia proposta por Ruy et al. (2016). A relação *define* faz uma associação do *Stakeholder* ao *Requisito*, ou seja, um *Stakeholder* é responsável por definir um requisito. Também foi criada uma relação inversa (*foiDefinidoPor*), que tem como objetivo mostrar que um requisito é definido por um determinado *Stakeholder*.

No contexto dos recursos, a Figura 28 especifica os recursos que fazem parte de um sistema embarcado. *SistemaEmbarcado* foi classificado com o estereótipo <<*kind*>>, pois representa um complexo funcional, ou seja, um todo que possui partes que contribuem de diferentes maneiras para sua funcionalidade. Portanto, com relação aos recursos que fazem parte de um *SistemaEmbarcado*, foi agrupado em *Ambiente*, *Hardware*, *Software* e *Norma* – esses conceitos foram construídos a partir do *Kind* da UFO-A.

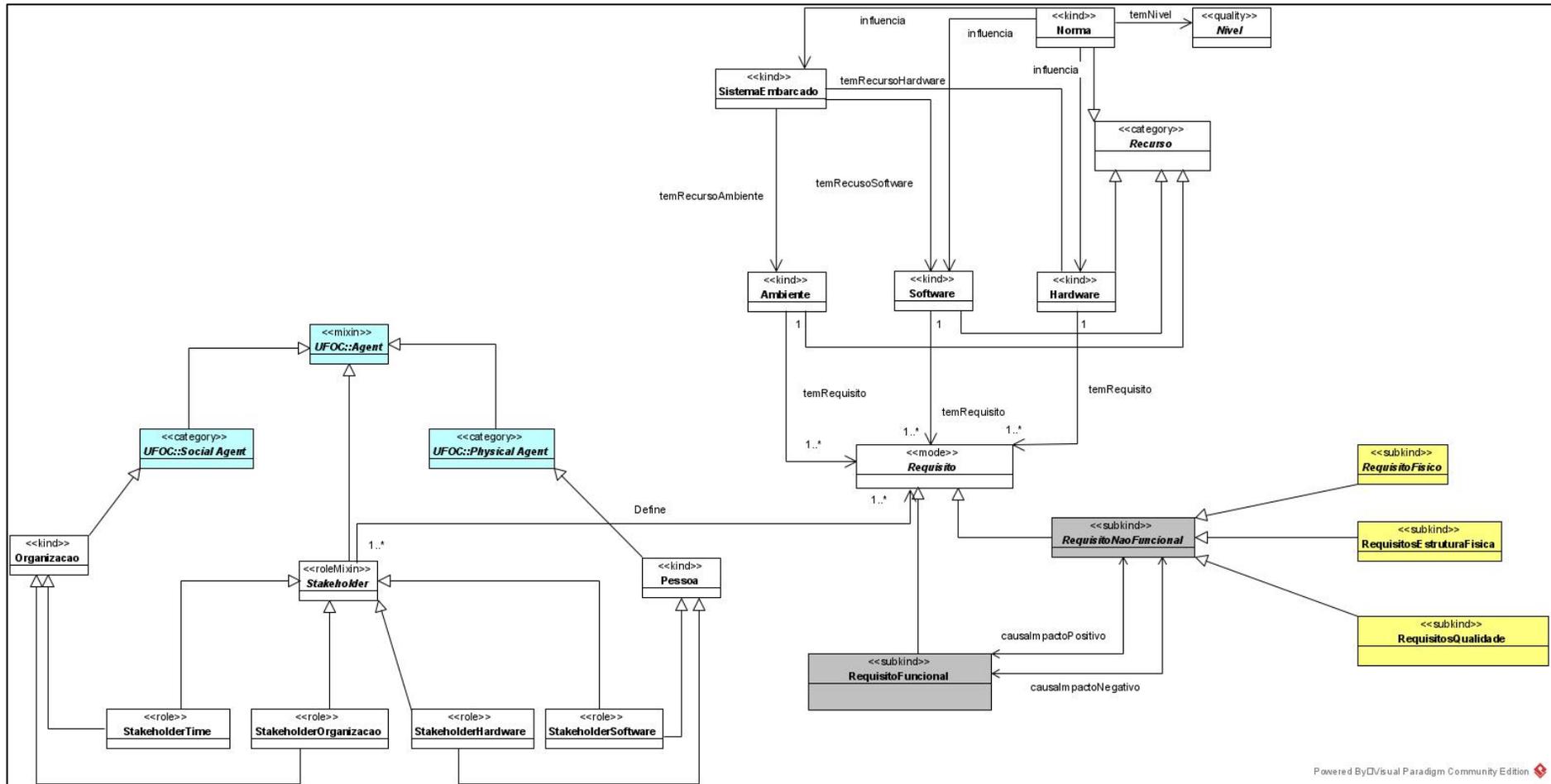
Foram definidas três associações entre *SistemaEmbarcado* e os recursos. Portanto, a relação *temRecursoAmbiente* faz uma associação entre *SistemaEmbarcado* e *Ambiente*, a relação *temRecursoHardware* faz uma associação entre *SistemaEmbarcado* e *Hardware*, e a relação *temRecursoSoftware* faz uma associação entre *SistemaEmbarcado* e *Software*. Portanto, um *SistemaEmbarcado* deve ter, pelo menos, um *Ambiente*, um *Hardware* e um *Software*. Esses recursos têm uma associação *temRequisito* com *Requisito*.

Alguns domínios de Sistema Embarcado são regidos por normas que regulamentam as necessidades exigidas para demonstrar que um sistema é seguro e confiável para se operar neste tipo de ambiente. Existem normas específicas para cada domínio (exemplo de normas no domínio aeronáutico: Software – RTCA DO-178C e Hardware – RTCA DO-254). Portanto, na OntoREES, foi definido o conceito *Norma* (um tipo de *Recurso*). Uma *Norma* pode influenciar o SE, *Hardware* ou *Software*. A associação *influencia* faz o relacionamento entre *Norma* e *SistemaEmbarcado*, *Norma* e *Software*, e entre *Norma* e *Hardware*.

Diante de alguns estudos realizados durante a realização desta tese, identificamos a importância que as normas possuem em domínios de sistemas embarcados, principalmente em sistemas críticos. Assim sendo, esse conceito foi adicionado à ontologia proposta, pois muitas normas influenciam consideravelmente na definição dos requisitos. A partir da OntoREES, ontologias de domínios poderão ser propostas focando em normas específicas. Também, uma

norma pode possuir um nível de criticalidade que implicará uma certa quantidade de esforço requerida para demonstrar adequação aos requisitos de certificação. Diante disso, foi elaborado o conceito *Nível* classificado como um tipo <<*quality*>> da UFO. Classificamos como *quality*, pois são momentos individuais que podem ser mapeados para algum espaço de qualidade, por exemplo, um nível pode mudar para outro nível mantendo sua identidade. Existe um relacionamento *temNível* de *Norma* para *Nível*. Esse nível pode variar em cada norma, por isso decidimos colocar o conceito mais geral (*Nível*) e as ontologias de domínio poderão explorar mais detalhadamente.

Figura 28 – Modelagem Agentes e Objetos da OntoREES.



Fonte: A Autora (2021).

O Quadro 16 apresenta em DL os axiomas  $\mathcal{J}\mathcal{A}\mathcal{R}$  que descrevem os Agentes e Recursos do domínio de sistemas embarcados.

**Quadro 16 – Axiomas de Agentes e Recursos.**

	$\text{Recurso} \equiv \text{Ambiente} \sqcup \text{Hardware} \sqcup \text{Software} \sqcup \text{Norma}$
	$\text{Ambiente} \sqsubseteq \text{Recurso}$
	$\text{Software} \sqsubseteq \text{Recurso}$
	$\text{Hardware} \sqsubseteq \text{Recurso}$
	$\text{Norma} \sqsubseteq \text{Recurso}$
	$\text{SistemaEmbarcado} \sqsubseteq \exists \text{temRecursoAmbiente.Ambiente} \sqcap \exists \text{temRecursoSoftware.Software} \sqcap \exists \text{temRecursoHardware.Hardware}$
	$\text{temRecurso} \equiv \text{éUmRecurso} -$
	$\text{Ambiente} \sqsubseteq \exists \text{temRequisito.Requisito}$
	$\text{Software} \sqsubseteq \exists \text{temRequisito.Requisito}$
	$\text{Hardware} \sqsubseteq \exists \text{temRequisito.Requisito}$
	$\text{Stakeholders} \equiv \text{StakeholderTime} \sqcup \text{StakeholderOrganizacao} \sqcup \text{StakeholderSoftware} \sqcup \text{StakeholderHardware}$
	$\text{StakeholderTime} \sqsubseteq \text{Organizacao} \sqcap \text{Stakeholder}$
$\mathcal{J}\mathcal{A}\mathcal{R}$	$\text{StakeholderOrganizacao} \sqsubseteq \text{Organizacao} \sqcap \text{Stakeholder}$
	$\text{StakeholderSoftware} \sqsubseteq \text{Pessoa} \sqcap \text{Stakeholder}$
	$\text{StakeholderHardware} \sqsubseteq \text{Pessoa} \sqcap \text{Stakeholder}$
	$\text{Stakeholder} \sqsubseteq \exists \text{define.Requisito}$
	$\text{define} \equiv \text{foiDefinidoPor} -$
	$\text{Organizacao} \sqsubseteq \neg \text{Pessoa}$
	$\text{StakeholderOrganizacao} \sqsubseteq \neg \text{StakeholderTime}$
	$\text{Ambiente} \sqsubseteq \neg \text{Software}$
	$\text{Software} \sqsubseteq \neg \text{Hardware}$
	$\text{Hardware} \sqsubseteq \neg \text{Ambiente}$
	$\text{Norma} \sqsubseteq \neg \text{Software}$
	$\text{Norma} \sqsubseteq \neg \text{Hardware}$
	$\text{Norma} \sqsubseteq \neg \text{Ambiente}$
	$\text{Norma} \sqsubseteq \exists \text{influencia.SistemaEmbarcado} \sqcap \exists \text{influencia.Hardware} \sqcap \exists \text{influencia.Software} \sqcap \forall \text{influencia.}(\text{SistemaEmbarcado} \sqcup \text{Hardware} \sqcup$

Software)
Ambiente $\sqsubseteq \exists \text{temRequisitoF.RequisitoFuncional}$
Ambiente $\sqsubseteq \forall \text{temRequisitoNF.RequisitoEstruturaFisica}$
Hardware $\sqsubseteq \exists \text{temRequisitoF.RequisitoFuncional}$
Hardware $\sqsubseteq \exists \text{temRequisitoNF.RequisitoNaoFuncional}$
Hardware $\sqsubseteq \exists \acute{\text{éInfluenciadoPor.Norma}}$
Software $\sqsubseteq \exists \text{temRequisitoF.RequisitoFuncional}$
Software $\sqsubseteq \forall \text{temRequisitoNF.RequisitoQualidade}$
SistemaEmbarcado $\sqsubseteq \exists \acute{\text{éInfluenciadoPor.Norma}}$
influencia $\equiv \acute{\text{éInfluenciadoPor}} -$
Norma $\sqsubseteq \forall \text{temNivel.Nivel}$
Hardware $\sqsubseteq \exists \text{nivelRequerido.Nivel}$
SistemaEmbarcado $\sqsubseteq \exists \text{nivelRequerido.Nivel}$
Software $\sqsubseteq \exists \text{nivelRequerido.Nivel}$

Fonte: A Autora (2021).

#### 6.1.4.2 Classificações dos Requisitos

A Figura 29 apresenta o módulo referente aos Requisitos que podem estar presentes em um projeto de sistema embarcado. A partir das entidades da UFO-A *Mode*, *kind*, *subKind*, e *quality*, foram definidos os conceitos para o domínio de sistemas embarcados. Os requisitos representam as funcionalidades e restrições de um sistema. Na OntoREES classificamos os **Requisito** em **RequisitoFuncional** e **RequisitoNaoFuncional**. Três subdivisões foram criadas a partir do **RequisitoNaoFuncional**, uma para dividir os requisitos relacionados a estrutura física (**RequisitoFisico**), a segunda para dividir os requisitos relacionados as medidas da estrutura física de um componente ou ambiente (**RequisitoEstruturaFisica**), e a última para dividir os requisitos relacionados as restrições (**RequisitoQualidade**). **Requisito** foi definido como  $\langle\langle \text{Mode} \rangle\rangle$ , pois é considerado como uma intenção rígida. Diante disso, segundo Guizzardi (2021), um *Mode* as suas instâncias são aspectos de outros indivíduos e que provê um critério de identidade.

O **RequisitoFisico** foi subdividido em **Componentes**, **Conversores** e **Comunicacao**. Os **Componentes** representam componentes físicos que irão compor um SE. Estes foram subdivididos em **Bateria**, **Sensor**, **Atuador**, **Microcontrolador**, **Microprocessador**,

*MemoriaPrograma*, *MemoriaDados*, *InterfaceUsuario* (*Teclado*, *Botoes*, *DisplaySeteSegmentos* e *DisplayLCD*). É importante enfatizar que um SE é composto por outros componentes, porém, na OntoREES, consideramos os principais componentes encontrados na RSL e no *Survey* com especialistas do domínio.

Os *Conversores* representam as conversões de valores de um sistema para outro, ou seja, *Analógico/Digital* (converte um sinal analógico em digital) e *Digital/Analógico* (converte um sinal digital em analógico). Com relação a *Comunicação* entre os componentes, foram definidas três subdivisões: *Paralela*, *Serial*, e *Wireless*.

Adicionalmente, foram criadas quatro relações de associação (*causaImpactoPositivoEm*, *causaImpactoNegativoEm*, *éImpactadoNegativamentePor* e *éImpactadoPositivamentePor*). Essas associações acontecem entre *RequisitoFuncional* para *RequisitoNaoFuncional* e *RequisitoNaoFuncional* para *RequisitoFuncional*, ou seja, um *RequisitoFuncional* *causaImpactoPositivoEm* e *causaImpactoNegativoEm* *RequisitoNaoFuncional*, assim acontece com *RequisitoNaoFuncional* para *RequisitoFuncional*. Esses relacionamentos são importantes, pois os requisitos funcionais e não funcionais podem se impactar mutuamente. A criação desse relacionamento na OntoREES poderá facilitar o processo de rastreabilidade de requisitos, pois permitem uma análise inteligente de impacto, ou seja, se um *Stakeholder* deseja alterar um requisito, os requisitos rastreáveis permitem que ele verifique e relate o impacto da mudança solicitada. A seguir é apresentado um exemplo de impacto entre requisitos:

Exemplo 6.1: *o consumo de energia de um Transponder em avião deve ser mínimo, pois será alimentado pela bateria da própria aeronave. Esse aparelho instalado em aviões responde a sinais de rádio emitidos por outras fontes com a identificação da aeronave e um código sobre o status do vôo. Se esse componente físico tiver um consumo de energia muito alto pode causar um impacto negativo nos requisitos de Desempenho, ConsumoDeEnergia, Segurança e Custo.*

Um Sistema Embarcado possui ainda requisitos que representam medidas. Portanto, alguns aspectos inerentes a *RequisitoEstruturaFisica* foram definidos, tais como: *Temperatura*, *Peso*, *Altura*, *Largura* e *Umidade*. Esses requisitos foram classificados como <<quality>>, pois, segundo a UFO, é um momento universal intrínseco, ou seja, momentos que são dependentes de um único indivíduo para existir. Em Guizzardi et al. (2018), foi proposta uma melhoria na linguagem OntoUML, como também na teoria da UFO. Nessa proposta de melhoria, o <<quality>> foi considerado como um Sortal Rígido (*Rigid Sortal*) que fornece um Princípio de identidade às suas instâncias. Na OntoREES, um requisito de

*Temperatura, Peso, Altura, Largura e Umidade*, depende do indivíduo *Ambiente* ou *Componentes* para existir.

Exemplo 6.2: *em um sistema de bomba de insulina é necessário que o peso dos componentes físicos não ultrapasse 100 gramas. Portanto, o requisito **Peso** depende de requisitos relacionados a **Componentes** para existir.*

Na OntoREES foram definidas medidas que estão relacionadas a cada *Quality* (*Temperatura, Peso, Altura, Largura e Umidade*). Essas medidas foram classificadas com o estereótipo <<datatype>>, pois este é usado na OntoUML para especificar a estrutura de qualidade. Ele é um classificador semelhante a uma classe, mas suas instâncias são valores e não objetos. Uma estrutura de qualidade <<datatype>> é sempre relacionada a um <<quality>> através de uma relação de associação. Portanto, definimos os *DataType celsius, fahrenheit* para *Temperatura*, *UR%* para *Umidade*, *grama* e *quilo* para *Peso*, *centimetroA, metroA* para *Altura* e *centimetroL, metroL* para *Largura*. Através desses Datype podemos criar restrições para potenciais ontologias de domínio.

Os *Componentes e Ambiente* podem ter algum tipo de medida, assim sendo, foi definida uma associação com *RequisitoEstruturaFisica* através de *temRequisitosEstruturaFisica*.

Os requisitos *Físico* e de *EstruturaFisica* expressam as características físicas sobre um SE. A maioria desses sistemas estão acoplados em um sistema maior (automóvel, avião, eletrodomésticos, equipamentos médicos, entre outros). Diante disso, o seu formato pode ser ditado pela estética e por restrições críticas de dimensões, temperatura e peso, onde geralmente se busca o seu encaixe em pequenos espaços (WOLF, 2012). Definir esses tipos de requisitos se torna crucial em projetos de sistemas embarcados, pois apoiam decisões sobre o projeto de arquitetura.

Assim como os requisitos relacionados a estrutura física, foram definidos outros requisitos como <<quality>>, pois são considerados momento universal intrínseco. Esses requisitos estão relacionados às restrições que afetam a qualidade de um Sistema Embarcado; dessa forma, os seguintes aspectos inerentes a *RequisitoQualidade* foram definidos: *Manutenbilidade, Custo, Privacidade, Usabilidade, ConsumoEnergia, Portabilidade, Segurança, Confiabilidade, Desempenho*. Portanto, esses requisitos dependem de *RequisitoQualidade* para existir. Os requisitos *Desempenho* e *Confiabilidade* têm as seguintes subdivisões, respectivamente, *TempoDeResposta, TempoMedioFalha* e *TaxaOcorrenciaFalha*. Essas subdivisões foram definidas como *datatype*, pois consideramos como uma estrutura de qualidade, cujas instâncias são valores e não objetos. É importante

destacar que nesse trabalho foram escolhidos os principais requisitos não-funcionais de qualidade identificados na RSL e no *Survey*. Evitamos nos aprofundar em definir outros requisitos, pois impactariam na delimitação do escopo dessa pesquisa. É importante salientar, que a ontologia é facilmente extensível para outros elementos.

O Quadro 17 apresenta os axiomas  $\mathcal{R}$  que descrevem os requisitos do domínio de sistemas embarcados.

**Quadro 17 – Axiomas dos Requisitos.**

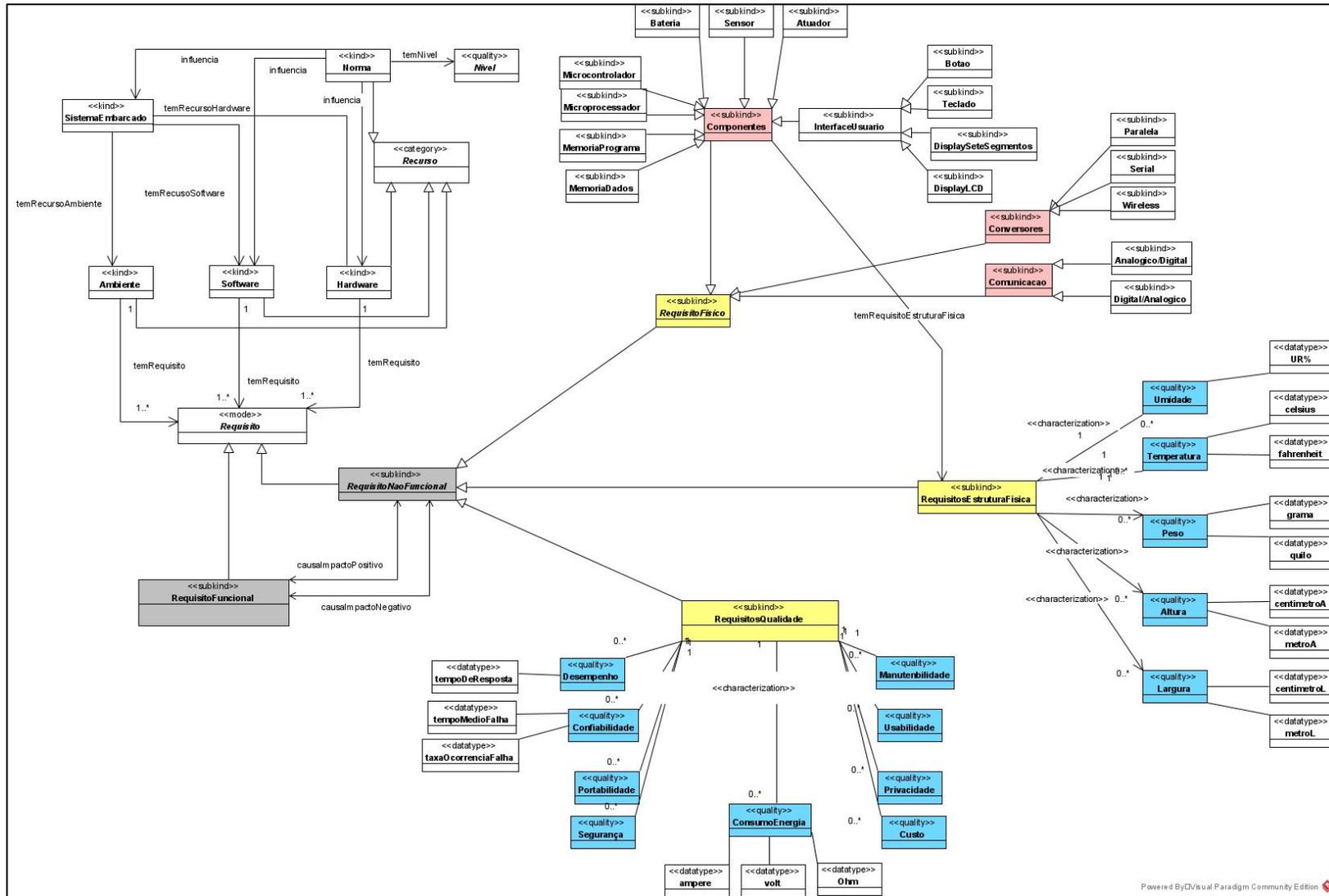
$\mathcal{R}$	Fisico $\sqsubseteq$ RequisitoNaoFuncional
	RequisitoEstruturaFisica $\sqsubseteq$ RequisitoNaoFuncional
	RequisitoQualidade $\sqsubseteq$ RequisitoNaoFuncional
	Componente $\sqsubseteq$ Fisico
	Conversores $\sqsubseteq$ Fisico
	Comunicacao $\sqsubseteq$ Fisico
	Bateria $\sqsubseteq$ Componentes
	Sensor $\sqsubseteq$ Componentes
	Atuador $\sqsubseteq$ Componentes
	Microcontrolador $\sqsubseteq$ Componentes
	Microprocessador $\sqsubseteq$ Componentes
	MemoriaPrograma $\sqsubseteq$ Componentes
	MemoriaDados $\sqsubseteq$ Componentes
	InterfaceUsuario $\sqsubseteq$ Componentes
	Botao $\sqsubseteq$ InterfaceUsuario
	Teclado $\sqsubseteq$ InterfaceUsuario
	DisplaySeteSegmentos $\sqsubseteq$ InterfaceUsuario
	DisplayLCD_ $\sqsubseteq$ InterfaceUsuario
	Analogico/Digital $\sqsubseteq$ Conversores
	Digital/Analogico $\sqsubseteq$ Conversores
	Paralela $\sqsubseteq$ Comunicacao
	Serial $\sqsubseteq$ Comunicacao
	Wireless $\sqsubseteq$ Comunicacao
	Temperatura $\sqsubseteq$ RequisitoEstruturaFisica
Umidade $\sqsubseteq$ RequisitoEstruturaFisica	

Peso $\sqsubseteq$ RequisitoEstruturaFisica
Altura $\sqsubseteq$ RequisitoEstruturaFisica
Largura $\sqsubseteq$ RequisitoEstruturaFisica
Manutenbilidade $\sqsubseteq$ RequisitoQualidade
Custo $\sqsubseteq$ RequisitoQualidade
Privacidade $\sqsubseteq$ RequisitoQualidade
Usabilidade $\sqsubseteq$ RequisitoQualidade
ConsumoEnergia $\sqsubseteq$ RequisitoQualidade
Portabilidade $\sqsubseteq$ RequisitoQualidade
Segurança $\sqsubseteq$ RequisitoQualidade
Confiabilidade $\sqsubseteq$ RequisitoQualidade
Desempenho $\sqsubseteq$ RequisitoQualidade
RequisitoFuncional $\sqsubseteq \neg$ RequisitoNaoFuncional
Fisico $\sqsubseteq \neg$ RequisitoEstruturaFisica
RequisitoEstruturaFisica $\sqsubseteq \neg$ RequisitoQualidade
RequisitoQualidade $\sqsubseteq \neg$ Fisico
Componentes $\sqsubseteq \neg$ Comunicacao
Comunicacao $\sqsubseteq \neg$ Conversores
Conversores $\sqsubseteq \neg$ Componentes
Requisito $\sqsubseteq \exists$ foiDefinidoPor.Stakeholder
Requisito $\equiv$ RequisitoFuncional $\sqcup$ RequisitoNaoFuncional
RequisitoFuncional $\sqsubseteq$ Requisito
RequisitoNaoFuncional $\sqsubseteq$ Requisito
temRequisito $\equiv$ éRequisito –
Componentes $\sqsubseteq \forall$ temRequisitoNF.RequisitoEstruturaFisica
Ambiente $\sqsubseteq \forall$ temRequisitoNF.RequisitoEstruturaFisica
RequisitoNaoFuncional $\sqsubseteq \exists$ temRequisitoNF.Fisico $\sqcap$
$\exists$ temRequisitoNF.RequisitoQualidade $\sqcap$
$\exists$ temRequisitoNF.RequisitoEstruturaFisica $\sqcap \forall$ temRequisitosNF.(Fisico $\sqcup$
RequisitoQualidade $\sqcup$ RequisitoEstruturaFisica)
RequisitoFuncional $\sqsubseteq \forall$ causaImpactoNegativo.RequisitoNaoFuncional
RequisitoFuncional $\sqsubseteq \forall$ causaImpactoPositivo.RequisitoNaoFuncional

	$\text{RequisitoFuncional} \sqsubseteq \forall \text{ImpactadoPositivamentePor.RequisitoNaoFuncional}$
	$\text{RequisitoFuncional} \sqsubseteq \forall \text{ImpactadoNegativamentePor.RequisitoNaoFuncional}$
	$\text{RequisitoNaoFuncional} \sqsubseteq \forall \text{causaImpactoNegativo.RequisitoFuncional}$
	$\text{RequisitoNaoFuncional} \sqsubseteq \forall \text{causaImpactoPositivo.RequisitoFuncional}$
	$\text{RequisitoNaoFuncional} \sqsubseteq \forall \text{ImpactadoPositivamentePor.RequisitoFuncional}$
	$\text{RequisitoNaoFuncional} \sqsubseteq \forall \text{ImpactadoNegativamentePor.RequisitoFuncional}$

**Fonte: A Autora (2021).**

Figura 29 – Modelagem Requisitos da OntoREES.



Fonte: A Autora (2021).

## 6.2 ONTOLOGIA DE DOMÍNIO

Nesta seção iremos explorar a Ontologia de Domínio construída a partir da OntoREES. Essa ontologia é específica para o domínio de Sistemas Embarcados Aviônicos.

### 6.2.1 OntoREAV

A OntoREAV é uma ontologia de domínio desenvolvida para modelar os requisitos de um Sistema Embarcado Aviônico. Essa ontologia foi construída a partir dos conceitos da ontologia core OntoREES. Para definição dessa ontologia foram elaboradas cinco questões de competência (conforme apresentado no Quadro 18) e enviadas a um Engenheiro do domínio Aviônico. Ele foi de extrema importância para determinar e validar a ontologia proposta (Capítulo 7). O domínio aviônico possui algumas normas, *stakeholders* e requisitos específicos. Diante disso, foram criados novos conceitos e relacionamentos a partir da OntoREES.

**Quadro 18 – Questões da OntoREAV.**

<b>Questões para definição da OntoREAV</b>
Quais <i>stakeholders</i> participam do processo de definição dos requisitos de Hardware?
Quais <i>stakeholders</i> participam do processo de definição dos requisitos de Software?
Quais requisitos você listaria como específicos para o contexto aviônico (classificá-los nas categorias da OntoREES)?
Quais requisitos podem impactar positivamente e negativamente entre si?
Quais normas são utilizadas no domínio aviônico? Essas normas influenciam os requisitos? Se Sim, quais?

**Fonte: A Autora (2021).**

A Figura 31 apresenta o módulo referente aos Recursos e Agentes que podem estar presentes no projeto de definição dos requisitos de um sistema embarcado aviônico. Nesse domínio existem alguns *Stakeholders* específicos, sendo eles: Engenheiro de Segurança, Engenheiro Eletrônico, Engenheiro de Certificação, Engenheiro Mecânico e Engenheiro de Usabilidade. Portanto, foram definidos os seguintes conceitos para representar os

*Stakeholders: EngenheiroSeguranca, EngenheiroEletronico, EngenheiroCertificacao, EngenheiroMecanico e EngenheiroUsabilidade.* Cada um possui habilidade específica para definição dos requisitos. O Engenheiro de Certificação, por exemplo, é responsável por garantir a conformidade dos requisitos com as normas. De fato, no contexto de sistema aviônico, o desenvolvimento é orientado por várias normas. Os seguintes conceitos foram, então, definidos para representar as normas: *NormaSoftware*, *NormaHardware* e *NormaSistemaEmbarcado*. Outrossim, para cada conceito foi instanciada a norma correspondente:

- *NormaSoftware* (RTCA-DO-178-C) - apresenta os processos de desenvolvimento de *software*, incluindo o processo de especificação de requisitos.
- *NormaHardware* (RTCA-DO-254) - apresenta os processos de desenvolvimento de *hardware*, incluindo o processo de especificação de requisitos.
- *NormaSistemaEmbarcado* (ARP-4754-A) e (RTCA DO-356) – a primeira apresenta os processos de desenvolvimento do sistema embarcado completo, não apenas o *software* e *hardware* eletrônico. A segunda descreve diretrizes, métodos e ferramentas usadas na realização de um programa de segurança de aeronavegabilidade. Essa segurança está relacionada à proteção de interferências não autorizadas.

As normas RTCA-DO-178-C e RTCA-DO-254 possuem níveis de criticalidade, sendo elas: A (catastrófica), B (perigosa), C (maior), D (menor) e E (sem impacto). Segundo a referida norma: (i) a falha catastrófica é uma condição que causa mortes ou perda do veículo; (ii) a falha perigosa é uma condição que pode causar grandes danos humanos severos ou danificação do veículo; (iii) a falha maior é condição que causa danos menores e pode provocar atraso; (iv) a falha menor é uma condição que não provoca danos, mas deve ser solucionada e, por fim, (v) a falha sem impacto é a situação que não causa efeito nenhum.

É importante ressaltar que cada norma possui objetivos específicos para cada nível. A norma RTCA-DO-178-C, em particular, classifica a criticalidade levando em consideração os objetivos associados (conforme apresentado na Figura 30).

**Figura 30 – Condições de falha de sistemas pela RTCA DO-178C**

Condição de Falha	Nível de Software Requerido	Objetivos Associados
Catastrófica	A	71
Perigosa	B	69
Maior	C	62
Menor	D	24
Sem Impacto	E	0

Fonte: (MARQUES, 2019)

Na OntoREAV é possível classificar qual o nível de criticalidade para o *Hardware*, *Software* e/ou SistemaEmbarcado. Isso torna-se importante, pois o *stakeholders* tem uma atenção maior para determinados requisitos e seguem os objetivos proposto pelas normas (garantindo uma maior segurança ao sistema proposto). Durante a realização da OntoREAV, surgiu a ideia de propor ontologias específicas para normas de sistemas aviônicos, pois cada norma possui suas particularidades e são exigências obrigatórias nesses tipos de sistemas, e depois integrá-las a OntoREAV. Portanto, será uma das propostas de continuidade deste trabalho destacadas no Capítulo 8.

Segundo Marques (2019), na RTCA-DO-178-C, a rastreabilidade entre os requisitos deve acontecer em vários sentidos, ou seja, os requisitos de *software* devem apresentar rastreabilidade com os requisitos de *hardware* e vice-versa. Na OntoREES, foi realizado esse processo de rastreabilidade. Logo, é possível rastrear os requisitos através dos relacionamentos *causaImpactoPositivoEm* e *causaImpactoNegativoEm*. Portanto, a OntoREAV utiliza a norma RTCA-DO-178-C (podendo classificar o nível de criticalidade) e consegue rastrear os requisitos através dos relacionamentos propostos na OntoREES.

É importante destacar que cada norma pode influenciar a definição dos requisitos de um *Software*, *Hardware* ou *SistemaEmbarcado*. Sendo assim, foi utilizada a associação *influencia* da OntoREES para materializar os relacionamentos.

No domínio de sistemas aviônicos, ocorre o carregamento (atualização) do *software*, ou seja, quando uma nova versão do programa corrige problemas identificados anteriormente, incluindo novos recursos ou melhorias (MARQUES; YELISETTY; BARROS, 2020). Essa atualização é crucial em alguns desses sistemas, pois através de uma configuração inadequada, carregamentos incompletos e/ou conflitos de versões podem causar mortes ou perdas irreparáveis. Sendo assim, foi criado na OntoREAV a categoria *CarregamentoSoftware*. Um carregamento pode acontecer de forma básica (*Basico*), Verificação Automática (*VerificacaoAuto*), Verificação Manual (*VerificacaoManual*) e

Troca de Versão (*TrocaVersao*). Foi utilizada a associação *temRequisito* para relacionar *CarregamentoSoftware* com *Requisito*, pois cada um tem requisitos associados.

Os axiomas em *JAR* (apresentado no Quadro 19) detalham a descrição dos agentes, recursos e requisitos nesse domínio. A Figura 31 apresenta a modelagem da OntoREAV, destacando quais os conceitos acrescentados para o domínio específico.

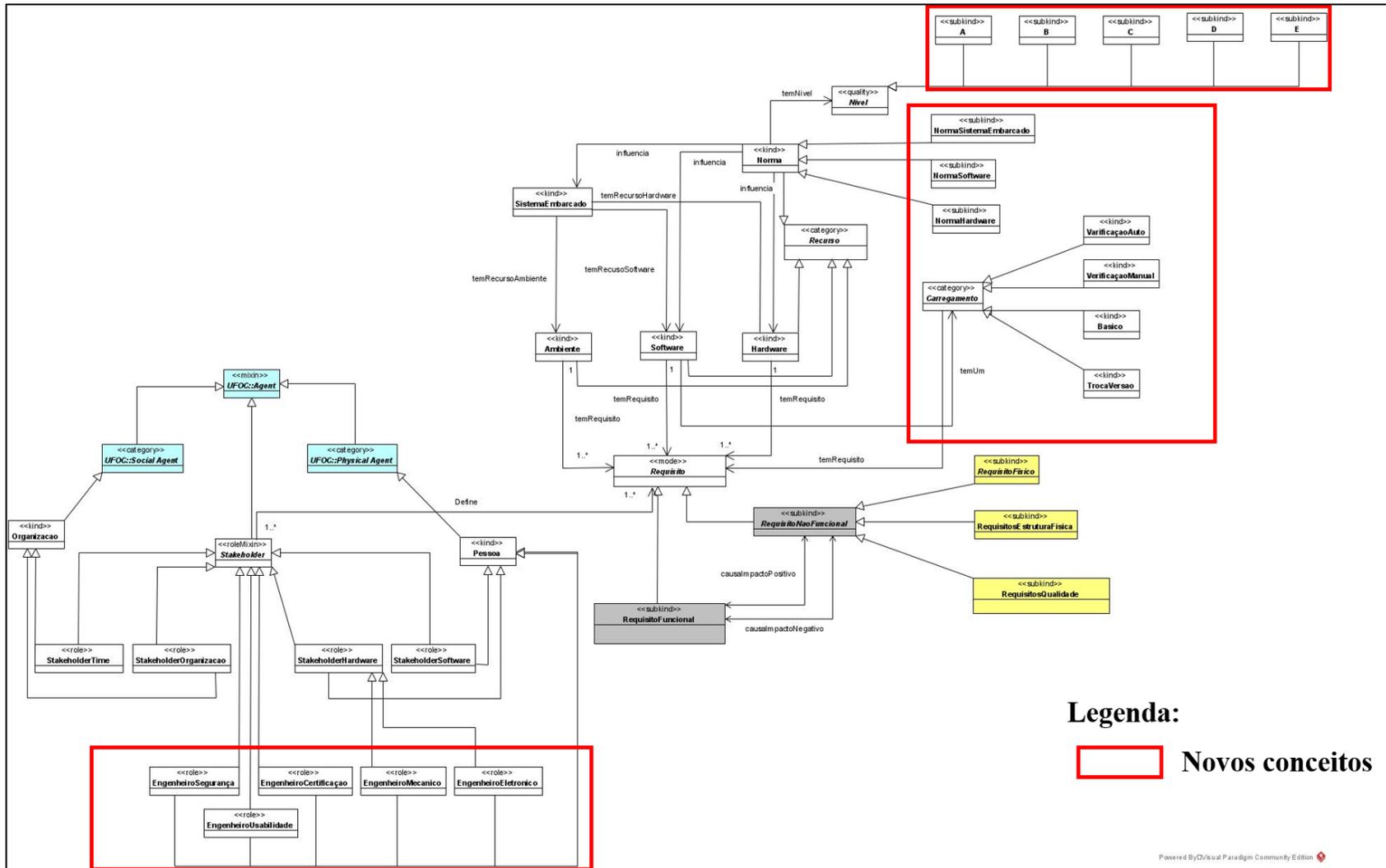
**Quadro 19 – Axiomas dos Agente, Recursos e Requisitos da OntoREAV.**

<b>JAR</b>	$\text{NormaSistemaEmbarcado} \sqsubseteq \text{Norma}$
	$\text{NormaHardware} \sqsubseteq \text{Norma}$
	$\text{NormaSoftware} \sqsubseteq \text{Norma}$
	$\text{NormaSistemaEmbarcado} \equiv \{\text{ARP-4754-A, RTCA DO-356, RTCA DO-326}\}$
	$\text{NormaHardware} \equiv \{\text{RTCA-DO-254}\}$
	$\text{NormaSoftware} \equiv \{\text{RTCA-DO-178-C}\}$
	$\text{NormaSoftware} \sqsubseteq \forall \text{influencia.Software}$
	$\text{NormaHardware} \sqsubseteq \forall \text{influencia.Hardware}$
	$\text{NormaSistemaEmbarcado} \sqsubseteq \forall \text{influencia.SistemaEmbarcado}$
	$\text{EngenheiroSeguran\c{c}a} \sqsubseteq \text{Stakeholder}$
	$\text{EngenheiroCertificacao} \sqsubseteq \text{Stakeholder} \sqcap \text{Pessoa}$
	$\text{EngenheiroEletronico} \sqsubseteq \text{Stakeholder} \sqcap \text{Pessoa}$
	$\text{EngenheiroMecanico} \sqsubseteq \text{Stakeholder} \sqcap \text{Pessoa}$
	$\text{EngenheiroUsabilidade} \sqsubseteq \text{Stakeholder} \sqcap \text{Pessoa}$
	$\text{EngenheiroCertificacao} \sqsubseteq \forall \text{garanteConformidade.Norma}$
	$\text{EngenheiroSeguran\c{c}a} \sqsubseteq \forall \text{define.Seguran\c{c}a}$
	$\text{EngenheiroEletronico} \sqsubseteq \forall \text{define.Fisico}$
	$\text{EngenheiroMecanico} \sqsubseteq \forall \text{define.Ambiente}$
	$\text{EngenheiroUsabilidade} \sqsubseteq \forall \text{define.Usabilidade}$
	$A \sqsubseteq \text{Nivel}$
	$B \sqsubseteq \text{Nivel}$
	$C \sqsubseteq \text{Nivel}$
	$D \sqsubseteq \text{Nivel}$
	$E \sqsubseteq \text{Nivel}$
$A \equiv \{\text{Catastrofica}\}$	

$B \equiv \{\text{Perigosa}\}$
$C \equiv \{\text{Maior}\}$
$D \equiv \{\text{Menor}\}$
$E \equiv \{\text{SemImpacto}\}$
$\text{Software} \sqsubseteq \exists \text{itemRCarregamento}.\text{CarregamentoSoftware}$
$\text{temRCarregamento} \equiv \acute{e}\text{UmRCarregamentoDo} -$
$\text{basico} \sqsubseteq \text{CarregamentoSoftware}$
$\text{VerificacaoManual} \sqsubseteq \text{CarregamentoSoftware}$
$\text{VerificacaoAuto} \sqsubseteq \text{CarregamentoSoftware}$
$\text{TrocaVersao} \sqsubseteq \text{CarregamentoSoftware}$
$\text{CarregamentoSoftware} \sqsubseteq \exists \text{possui}.\text{basico} \sqcap \exists \text{possui}.\text{VerificacaoManual}$ $\exists \text{possui}.\text{VerificacaoAuto} \sqcap \exists \text{possui}.\text{TrocaVersao} \sqcap \forall \text{possui}.\text{(Rbasicos} \sqcup$ $\text{VerificacaoManual} \sqcup \text{VerificacaoAuto} \sqcup \text{TrocaVersao)$
$\text{CarregamentoSoftware} \sqsubseteq \exists \text{itemRequisito}.\text{Requisito}$

**Fonte: A Autora (2021).**

Figura 31 – Modelagem da OntoREAV.



Fonte: A Autora (2021).

### 6.3 TRABALHOS RELACIONADOS

Ontologias de requisitos foram criadas, expandidas e modificadas nos últimos anos. Nesse sentido, uma RSL sobre ontologias na engenharia de requisitos foi realizada (DERMEVAL et al., 2016), mostrando a importância destes modelos conceituais no contexto da Engenharia de Requisitos em sistemas de propósito geral.

A partir dos resultados levantados na RSL, surgiu a necessidade de propor uma ontologia para especificação de requisitos voltada ao domínio de SE, pois não foi encontrado nenhum trabalho nessa temática. Os estudos que apresentam uma correlação mais significativa com esta pesquisa, encontrados na literatura, foram ontologias de requisitos proposta para sistemas de propósito geral e embarcados, mas que não levam em consideração especificidades do domínio de SE. Esses trabalhos são discutidos a seguir:

- **T1** – (SAEED et al., 2019): esse trabalho propõe a ontologia ADISTES (*Active Diagnosis based on Semantic Web Technologies for Distributed Embedded Real-Time Systems*) para um sistema de diagnóstico ativo de sensores e atuadores. A ontologia proposta é uma ontologia geral que pode ser usado por diferentes casos e sistemas para integrar seus sensores e atuadores. O foco principal foi propor uma ontologia que mostra o funcionamento dos sensores e atuadores em SE.
- **T2** – (TAN et al., 2016): nesse estudo foi apresentado uma ontologia que foi desenvolvida para representar os requisitos de *software* embarcado de um aplicativo da indústria aviônica. A ontologia é utilizada para automatizar uma parte do processo de teste, nomeadamente, a criação de geração de casos de teste.
- **T3** – (BORGES RUY et al., 2016): esse estudo visa estabelecer uma conceituação comum no domínio do processo de *software*, incluindo processos, atividades, recursos, pessoas, artefatos e procedimentos. Como uma ontologia central, a ontologia proposta fornece os conceitos gerais para processos de *software*, para serem especializados e reutilizados em ontologias específicas de domínio. A ontologia proposta é específica para domínio de propósito geral e não incluem aspectos específicos de SE.

- **T4** – (ALRUMAIH; MIRZA; ALSALAMAH, 2020): nesse estudo, uma ontologia de domínio é proposta para apresentar uma técnica de classificação de requisitos que pode ser usada para compartilhar e descrever diferentes classificações.

O Quadro 20 explicita as características apresentadas pela maioria dos trabalhos correlacionados (e principais diferenças). Definimos seis características que serão analisadas nos trabalhos relacionados. Essas características foram escolhidas com base no que é proposto nesta tese, o que não diminui a importância dos trabalhos citados.

**Quadro 20 – Trabalhos Relacionados.**

<b>Trabalhos</b> <b>Características</b>	<b>T1</b> (SAEED et al., 2019)	<b>T2</b> (TAN et al., 2016)	<b>T3</b> (BORGES RUY et al., 2016)	<b>T4</b> (ALRUMAI H; MIRZA; ALSALAMA H, 2020)	<b>Tese</b>
Baseada em uma Ontologia de TOPO					
Requisitos Funcionais e Não Funcionais					
Divisão entre <i>Software</i> e <i>Hardware</i>					
Sistemas Embarcados					
Formalização na Lógica Descritiva					
Metodologia para Desenvolvimento					

**Fonte: A Autora (2021).**

Analisando as características citadas no Quadro 20, temos que:

- **Baseada em uma Ontologia de Topo** – apenas um trabalho (T3) leva em consideração uma ontologia de Topo. A ontologia de topo utilizada em T3 foi a UFO, a mesma utilizada nesta tese. Essas ontologias são independentes de qualquer domínio e provê um conjunto de relações e conceitos mais genéricos que podem ser utilizados em diversos domínios.
- **Requisitos Funcionais e Não Funcionais** – com relação aos requisitos abordados nas ontologias, apenas um não classificou a ontologia com relação aos requisitos. O trabalho T1 propôs uma ontologia voltada somente para funcionamento dos sensores e atuadores e não são classificados como requisitos. Os trabalhos T2, T3, T4 e esta tese, classificou os requisitos em funcionais e não funcionais. É importante destacar que a divisão desses requisitos em projetos de SE é importante, pois os requisitos não funcionais são considerados, muitas vezes, mais importantes do que requisitos funcionais. Os trabalhos T2 e T3 expôs os requisitos não funcionais de forma superficial, o foco maior foi em requisitos funcionais.
- **Divisão entre *Software* e *Hardware*** – em projetos de sistemas embarcados é importante realizar a divisão entre os requisitos de *software* e *hardware*, conforme abordado no **Erro! Fonte de referência não encontrada.** Dos dois trabalhos relacionados específicos para SE, apenas um fez essa divisão (T2). Nesta tese fizemos essa divisão e classificamos em requisitos funcionais e não funcionais.
- **Sistema Embarcado** – com relação ao domínio, apenas dois trabalhos foram propostos para sistemas embarcados (T1 e T2).
- **Formalização na Lógica Descritiva** – nenhum dos trabalhos relacionados formalizaram o domínio na lógica descritiva *SR0JQ*. Então, diante dos trabalhos encontrados na literatura, esta tese foi a única proposta nessa linha de lógica descritiva. Como discutido, a formalização permite a capacidade de raciocinar no domínio, concretizada em tarefas de classificações e subsunções.

- **Metodologia para Desenvolvimento** – assim como em desenvolvimento de sistemas que existem etapas para serem seguidas, o desenvolvimento de uma ontologia precisa ser sistemático, ou seja, é necessário etapas claramente definidas para que possa ser gerado um modelo de domínio de maior qualidade. Portanto, analisamos se os trabalhos relacionados utilizaram algum tipo de metodologia e apenas o T1 não utilizou. As metodologias utilizadas foram Methontology, SAbiO e FOCA.

#### 6.4 SÍNTESE DO CAPÍTULO

Neste capítulo abordamos os dados referentes ao processo de condução para elaboração das ontologias OntoREES e OntoREAV. As fases de especificação, aquisição do conhecimento, conceitualização, formalização, documentação e implementação foram descritas nesse capítulo. As fases avaliação (validação e verificação) serão apresentados no próximo capítulo. O processo de axiomatização foi baseado na UFO e formalizada na lógica descritiva. Essa formalização permite a criação de diversas ontologias de domínio (médico, eletrônica de consumo, rodoviário, automotivo, entre outros). Por fim, apresentamos os trabalhos relacionados com esta pesquisa. O próximo capítulo apresenta as validações que foram utilizadas nas ontologias propostas.

## 7 VALIDAÇÃO DAS ONTOLOGIAS

Neste capítulo, descrevemos o planejamento e os resultados da validação das ontologias propostas (OntoREES e OntoREAV). Neste capítulo respondemos a seguinte questão de pesquisa: **“Como disponibilizar os resultados alcançados nesta pesquisa, para que se possa interagir e realizar simulações de definição de requisitos em projetos de SE?”**

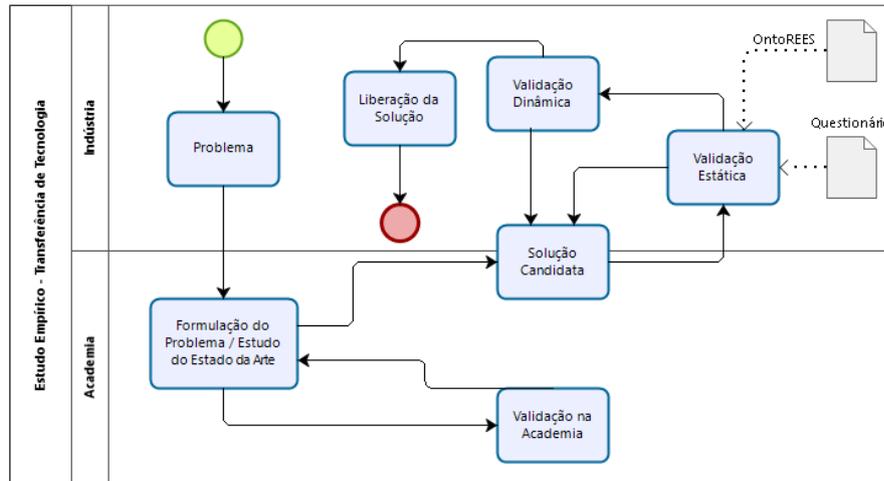
Uma ontologia pode ser construída do zero ou uma ontologia existente pode ser modificada para construir novas ontologias. Em ambos os casos, são necessárias técnicas para avaliar as características e a validade da ontologia. Essas técnicas não são úteis apenas no processo de engenharia de ontologias, mas também para usuários finais que procuram uma ontologia adequada para seu domínio de aplicação. Os usuários podem escolher a melhor ontologia de acordo com seus requisitos de aplicação em várias ontologias.

### 7.1 VALIDAÇÃO EMPÍRICA

**Os estudos empíricos têm um valor autônomo, mas também podem fazer parte de uma troca de conhecimento e esforço de melhoria em conjunto entre a academia e a indústria (WOHLIN et al., 2012). A engenharia de sistemas é preferencialmente conduzida em conjunto pela academia e pela indústria para permitir a transferência de conhecimento em ambas as direções e, ao final, a transferência de novos métodos, tecnologias e ferramentas da academia para a indústria. Portanto, neste trabalho, foi utilizado também o modelo de estudo empírico baseado em transferência de tecnologia proposto por Gorschek et al., (2006). Esse tipo de estudo empírico é dividido em sete etapas (conforme apresentado na**

Figura 32 e descritos a seguir).

Figura 32 – Modelo Empírico de Transferência de Tecnologia.



Fonte: Adaptado de (GORSCHER et al., 2006)

- **Problema** – corresponde à identificação de potenciais áreas de melhoria com base nas necessidades da indústria. Identificamos lacunas na elicitação e especificação de requisitos em sistemas embarcados;
- **Formulação do Problema / Estudo do Estado da Arte** – é formulada uma agenda de pesquisa usando várias avaliações para encontrar tópicos de pesquisa e formular declarações de problemas ao estudar o estado da arte. Diante das lacunas identificadas, foram realizados um Mapeamento Sistemático da Literatura, Revisão Sistemática da Literatura e um *Survey* com especialistas do domínio;
- **Validação na Academia** – deve ser realizada uma validação na academia. Nessa etapa, realizamos validações informais com estudantes de mestrado e doutorado com conhecimento na temática de pesquisa;
- **Solução Candidata** – envolve a formulação de uma solução candidata em cooperação com a indústria. Neste trabalho, a solução proposta foi uma ontologia de requisitos para SE;
- **Validação Estática** – os especialistas da indústria avaliam a solução candidata *offline*. Nesta etapa, realizamos uma apresentação da solução candidata aos especialistas do domínio através de videoconferências, e por e-mails foi entregue dois artefatos para validação (um modelo da OntoREES e um questionário). Com base nessa validação estática, recebemos um *feedback* do especialista do domínio e uma nova solução é proposta com os devidos ajustes;

- **Validação Dinâmica** – assim que a nova solução passar pela validação estática e houver acordo e compromisso para implementar a nova solução, é hora de passar para uma validação dinâmica. Isso é feito preferencialmente como uma avaliação piloto. Diante do cenário de isolamento social (devido ao COVID-19), não foi possível realizar esse tipo de validação. Portanto, apresentamos esse tipo de validação como trabalhos futuros;
- **Liberação da Solução** – aborda a liberação da solução passo a passo, enquanto permanece aberta a mudanças e adições menores.

O objetivo principal dessa validação (baseada em estudos empíricos) é coletar a opinião de especialistas, a fim de analisar se o conhecimento na literatura foi razoavelmente transferido e apresentado na Ontologia. Considerando os aspectos acima, o desenho da validação foi norteado pelas questões de pesquisa apresentadas no Quadro 21.

Para responder essas questões de pesquisa, entrevistamos um especialista do domínio de Sistema Embarcado. O especialista é um Professor Adjunto na Divisão de Ciência de Computação do Instituto Tecnológico de Aeronáutica (ITA). Foi engenheiro de desenvolvimento de produto da EMBRAER (Empresa Brasileira de Aeronáutica S/A), com mais de 15 anos de experiência em desenvolvimento de SE.

**Quadro 21 – Questões, Respostas e Justificativas do Estudo Empírico.**

<b>Questões</b>	<b>Resposta</b>	<b>Justificativa</b>
<b>QP1</b> - A OntoREES tem cobertura suficiente dos principais requisitos utilizados no desenvolvimento de Sistemas Embarcados?	Sim, parcialmente	O domínio de Sistema Embarcado é amplo e explora vários outros requisitos que não foram contemplados na ontologia proposta. Alguns requisitos foram citados para serem acrescentados a OntoREES.
<b>QP2</b> - A OntoREES é adequada para elicitación e especificação de requisitos em Sistemas	Sim, parcialmente	Considerou as mesmas observações da questão QP1.

Embarcados, em termos de utilidade e aplicabilidade?		
<b>QP3</b> – Existe um efeito positivo da aplicação da OntoREES quando utilizada em diferentes domínios de Sistemas Embarcados?	Sim, plenamente	Pela similaridade existente entre os requisitos, as normas e suas características existentes em sistemas críticos, a ontologia é útil para outros domínios como Médico, Automotivo, Ferroviário, entre outros.
<b>QP4</b> - Você recomendaria a OntoREES para outros profissionais e pesquisadores de Sistemas Embarcados?	Sim, plenamente	Como foi justificado na QP3 a ontologia proposta contempla diversos conceitos específicos para o domínio de SE e pode ser aplicado em diversos domínios.
<b>QP5</b> - Você tem alguma mudança, comentário e/ou melhorias que podem ser realizadas na OntoREES com base nas descobertas das questões anteriores ( <b>QP1, QP2, QP3 e QP4</b> )?	Sim	As principais considerações foram: alteração na parte de Interface do Usuário, Norma, Novos Requisitos e Segurança.

Fonte: A Autora (2021).

Analisamos as declarações do especialista e desenvolvemos a segunda versão do OntoREES. As alterações realizadas foram:

- Novos requisitos relacionados a componentes foram incluídos, mas não contemplamos todos que foram citados pelo especialista. Como citado no 6, o domínio de Sistema Embarcado é amplo e contempla vários requisitos específicos para cada domínio. O foco da OntoREES é exatamente contemplar os principais requisitos e, a partir dela,

outros autores e especialistas propor novos conceitos, relacionamentos e novas ontologias de domínios.

- Quanto às Normas, no primeiro momento, só existia a associação *influencia* de *SistemaEmbarcado* para *Norma*. Portanto, foram acrescentadas novas associações de *influencia* para *Software* e *Hardware*, uma vez que uma norma pode influenciar o sistema como um todo ou apenas *Software e Hardware*. Essa modificação foi bem pertinente, porque quando criamos a ontologia de domínio (OntoREAV) percebemos que existem normas específicas para cada recurso ou o sistema como todo;
- Por fim, foram realizadas modificações relacionadas as associações *causaImpactoPositivo* e *causaImpactoNegativo*. Na primeira versão da OntoREES essas associações eram somente entre *RequisitoQualidade* para *RequisitoFuncional* e *Fisico* para *RequisitoQualidade*, mas foi observado que todos os requisitos (Funcionais e Não Funcionais) causam impactos entre si. Em vista disso, essas associações ficaram de *RequisitoFuncional* para *RequisitoNaoFuncional* e vice-versa.

## 7.2 VERIFICAÇÃO – PADRÕES

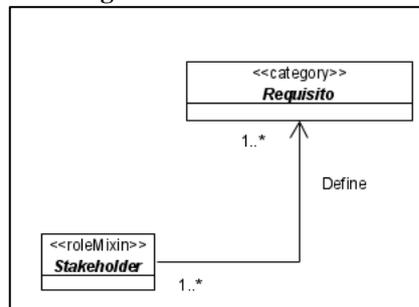
Todas as fases do desenvolvimento das ontologias foram acompanhadas por engenheiros do domínio de sistemas embarcados, com o intuito de que suposições equivocadas fossem identificadas e resolvidas nas fases iniciais do desenvolvimento. Além disso, ao adotar o UFO como a ontologia de nível superior, unimos categorias e estruturas de restrição, o que ajudou a gerar relações taxonômicas coerentes e estabelecer um conceito ontológico bem fundamentado. A modelagem das ontologias propostas está respondendo as regras de fundamentação da Ontologia de Fundamentação UFO. Esta avaliação se deu através do uso de padrões de modelagem.

Na modelagem conceitual, os padrões são fragmentos de certos modelos que encapsulam o conhecimento discutido anteriormente. Segundo Falbo et al. (2013), esses fragmentos são considerados os mais efetivos para nomear, organizar e raciocinar sobre algum tipo de conhecimento. Ao longo da engenharia ontológica, adotamos 03 padrões de modelagem, nomeadamente: Padrão *Role*, Padrão *RoleMixin* e Padrão *Category*.

No que concerne ao Padrão *Role*, os tipos estereotipados com «*role*» devem necessariamente se relacionar a algum outro tipo com o qual o primeiro mantém uma

dependência (portanto, a cardinalidade mínima da extremidade oposta deve ser 1). Um *Stakeholder*, por exemplo, só pode existir se houver um *Requisito* (Figura 33).

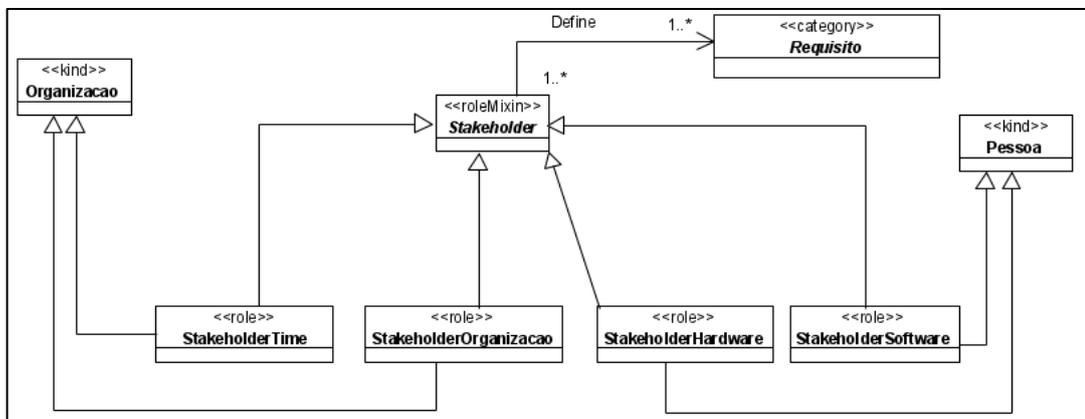
**Figura 33 – Padrão Role.**



Fonte: A Autora (2021).

O padrão RoleMixin é usado para lidar com situações em que o tipo anti-rígido dependente é usado especificamente para instanciar diferentes subtipos de <<kind>>, como *Stakeholder* (apresentado na Figura 34). Nestes casos, além de exigir dependências externas como o tipo <<role>> requisita, o supertipo deve ser um <<rolemixim>>.

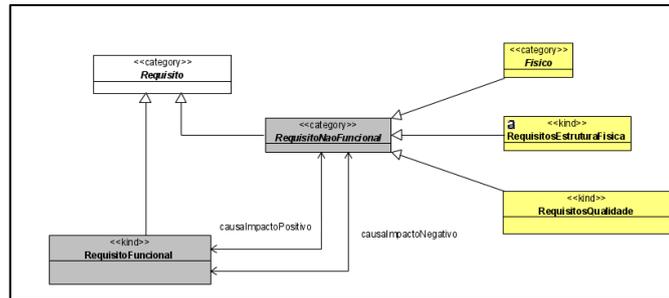
**Figura 34 – Padrão RoleMixin.**



Fonte: A Autora (2021).

Com relação ao padrão Category, os tipos assim estereotipados representam tipos rígidos e são independentes. Eles agrupam propriedades que são comuns a diferentes *Substance Sortals*. Recurso, Nivel, Requisito (Figura 35), RequisitoNaoFuncional (Figura 35) e Fisico, foram definidos como Category.

**Figura 35 – Padrão Category.**



**Fonte: A Autora (2021).**

### 7.3 VALIDAÇÃO – UTILIZANDO INFERÊNCIAS ABOX

Nesta seção, fornecemos exemplos de requisitos utilizados no desenvolvimento de Sistema Embarcados; esses foram extraídos da literatura na temática de pesquisa. Esses exemplos foram instanciados para demonstrar a adequação do modelo para o domínio de sistemas embarcados. A instanciação da ontologia permite consultas e inferências relacionadas ao domínio, mostrando a coerência delas em modelar o domínio ao qual se propõem.

#### 7.3.1 Inferências Realizadas

Conforme apresentado no 6, as ontologias foram construídas utilizando a Ferramenta Protegé (versão 5.5.0). Sendo assim, suas instanciações foram construídas nessa ferramenta, e o *reasoner* utilizado foi o HermiT (versão 1.4.3) e Pellet, também nativos do Protegé. A seguir é apresentado um cenário real com o intuito de ilustrar a simulação realizada pela ferramenta Protegé. Portanto, os resultados apresentados estão em forma de ABox. Para cada exemplo, destacamos a entrada e as inferências obtidas. As ontologias em OWL estão disponíveis em: <https://github.com/aedasousa/OntoREES.git>.

**Cenário 1** – Foi instanciada a ontologia de domínio OntoREAV com requisitos de um projeto de carregamento de software em sistema *Loadable Airborne System (LAS)* (MARQUES; YELISETTY; BARROS, 2020). O carregamento do software ocorre quando uma nova versão corrige problemas identificados anteriormente, incluindo novos recursos ou melhorias. No trabalho de Marques (2020), ele propôs um *framework* com 23 requisitos específicos para carregamento de *software* para esses tipos de sistemas. Foram utilizadas

normas correlacionadas ao *framework* proposto, ou seja, normas específicas para o desenvolvimento de sistemas aviônicos.

ABox do Cenário 1: O Quadro 22 apresenta as instâncias do cenário do domínio LAS.

**Quadro 22 – ABox Instâncias Cenário 1.**

<i>Ac</i> <sub>1</sub>	<p>SistemaEmbarcado(avionico-FAB), Software(software-LAS), Hardware(hardware), Ambiente(ambiente), temRecurso(avionico-FAB, software-LAS), temRecurso(avionico-FAB, hardware), temRecurso(aviônico-FAB, ambiente), NormaSistemaEmbarcado(RTCA-DO-178-C), influencia(RTCA-DO-178-C, avionico-FAB), garanteConformidade(Cleyton, RTCA-DO-178-C), nivelRequerido(A, software-LAS), Requisito(BLR-1), Requisito(BLR-2), Requisito(BLR-3), Requisito(MCR-6), Requisito(MCR-7), Requisito(MCR-8), Requisito(ACR-10), Requisito(ACR-11), Requisito(ACR-12), Requisito(VER- 20), Requisito(VER-21), Requisito(VER-22), Requisito(R01-Custo), define(Fernanda, BLR-1), define(Fernanda, BLR-2), define(Fernanda, BLR-3), define(Fernanda, VER-20), define(Fernanda, VER-21), define(Fernanda, VER- 22), define(Johnny, MCR-6), define(Johnny, MCR-7), define(Johnny, MCR-8), define(Johnny, ACR-10), define(Johnny, ACR-11), define(Johnny, ACR-12), temRequisito(software-LAS, BLR-1), temRequisito(software-LAS, BLR-2), temRequisito(software-LAS, BLR-3), temRequisito(software-LAS, MCR-6), temRequisito(software-LAS, MCR-7), temRequisito(software-LAS, MCR-8), temRequisito(software-LAS, ACR-10), temRequisito(software-LAS, ACR-11), temRequisito(software-LAS, ACR-12) temRequisito(software-LAS, VER-20), temRequisito(software-LAS, VER-21), temRequisito(software-LAS, VER-22), causaImpactoPositivoEm(BLR-2, MCR- 6), causaImpactoNegativoEm(ACR-10, R01-Custo), causaImpactoNegativoEm(MCR-6, R01-Custo), causaImpactoNegativoEm(ACR- 10, R01-Custo),</p>
------------------------	--

**Fonte: A Autora (2021).**

O *framework* dispõe de definições importantes para viabilizar os requisitos, sendo eles:

- *Aircraft Configuration Control System (ACCS)* – sistema de verificação automática para garantir o carregamento adequado da nova versão do *Software* (NSV);
- *Aircraft Display System (ADS)* – sistema que exibirá a Identificação do *Software* (SI) em sua nova versão do *Software*;
- *Cyclic Redundancy Check (CRC)* – método de detecção de erros usado em redes digitais e dispositivos de armazenamento para detectar mudanças acidentais em *strings* de dados;
- *Field Software Loading Procedure (FLSP)* – procedimento que descreve o carregamento de uma NSV;
- *Loading Computer (LC)* – computador responsável por carregar uma NSV;
- *Loading Interface (LI)* – interface física para carregamento de *software*;
- *Logbook (LB)* – livro de registro de operações de manutenção de aeronaves;
- *Manual Check Procedure (MCP)* – procedimento de verificação manual (humana) para garantir o carregamento adequado da NSV;
- *New Software Version (NSV)* – uma versão mais recente do *software*.
- *Previous Software Version (PSV)* – Versão do *software* antes de carregar a nova versão;
- *Software Identification (SI)* – identificação de uma versão de *software*.
- *Supplier (SUP)* – entidade responsável pelo fornecimento do NSV. Pode ser um fabricante de aeronaves ou um fornecedor de sistema.

O Quadro 23 apresenta os requisitos utilizados e sua descrição.

**Quadro 23 – Requisitos Loadable Airborne System.**

<b>Requisito</b>	<b>Descrição</b>	<b>Tipo</b>
BLR-1	O LAS deve possuir uma LI para permitir o carregamento do NVS	Usabilidade
BLR-2	A SUP desenvolverá o FLSP.	Usabilidade
BLR-3	O SUP deve identificar NSV.	Segurança
MCR-6	O FLSP deve incluir o MCP.	Usabilidade
MCR-7	O MCP deve incluir o SI esperado do NSV.	Usabilidade
MCR-8	O MCP deve incluir o CRC esperado do NSV.	Manutenibilidade

ACR-10	O ACCS deve ser capaz de garantir que a versão carregada está completa, comparando o CRC carregado com o CRC esperado.	Segurança
ACR-11	O ACCS deve ser capaz de garantir que a versão carregada está correta.	Confiabilidade
ACR-12	O ACCS deve se comunicar com o ADS se a versão carregada tiver um CRC diferente do esperado.	Confiabilidade
VER-20	O ACCS deve comparar se todos os LAS têm o mesmo SI.	Confiabilidade
VER-21	O ACCS deve notificar o ADS se o LAS tem SI diferente.	Confiabilidade
VER-22	O MCP deve conter instruções para verificar se todos os LAS têm o mesmo SI.	Manutenibilidade

**Fonte: A Autora (2021).**

No Quadro 24 são apresentadas os ABox das inferências realizadas para o Cenário 1. Essas inferências tem como objetivo validar a ontologia para demonstrar a adequação ao domínio aviônico. É importante salientar que os resultados das inferências serão apresentados em ABox. Nesse processo levamos em consideração os axiomas apresentados no Quadro 22 e os axiomas terminológicos dispostos nas Ontologias OntoREES e OntoREAV.

**Quadro 24 – Inferências ABox 1.**

define(fernanda, BLR-1), Requisito(BLR-1)  = Stakeholder(fernanda)
define(fernanda, BLR-2), Requisito(BLR-2)  = Stakeholder(fernanda)
define(fernanda, BLR-3), Requisito(BLR-3)  = Stakeholder(fernanda)
define(fernanda, VER-20), Requisito(VER-20)  = Stakeholder(fernanda)
define(fernanda, VER-21), Requisito(VER-21)  = Stakeholder(fernanda)
define(fernanda, VER-22), Requisito(VER-22)  = Stakeholder(fernanda)
define(johnny, MCR-6), Requisito(MCR-6)  = Stakeholder(johnny)
define(johnny, MCR-7), Requisito(MCR-7)  = Stakeholder(johnny)
define(johnny, MCR-8), Requisito(MCR-8)  = Stakeholder(johnny)
define(johnny, ACR-10), Requisito(ACR-10)  = Stakeholder(johnny)
define(johnny, ACR-11), Requisito(ACR-11)  = Stakeholder(johnny)
define(johnny, ACR-12), Requisito(ACR-12)  = Stakeholder(johnny)

**Fonte: A Autora (2021).**

Conforme apresentado no Quadro 24, nessas primeiras inferências, temos que o Fernanda e Johnny reúne as condições necessárias para ser considerado um Stakeholder, pois, segundo os axiomas da OntoREES, é considerado um Stakeholder quando define algum requisito. Também é possível inferir cada Stakeholder define alguns requisitos, sendo eles:

Johnny – MCR-6, MCR-7, MCR-8, ACR-10, ACR-11 e ACR-12

Fernanda – BLR-1, BLR-2, BLR-3, VER-10, VER-21 e VER-22

**Quadro 25 – Inferências ABox 2.**

$\text{temRecurso}(\text{avionico-FAB}, \text{software-LAS}), \text{temRecurso}(\text{avionico-FAB}, \text{hardware}),$ $\text{temRecurso}(\text{avionico-FAB}, \text{ambiente}), \text{Software}(\text{software-LAS}),$ $\text{Hardware}(\text{hardware}), \text{Ambiente}(\text{ambiente}) \models \text{SistemaEmbarcado}(\text{avionico-FAB})$
--

**Fonte: A Autora (2021).**

Conforme apresentado no Quadro 25, os recursos software-LAS, hardware e ambiente são considerados recursos de avionico-FAB. Diante disso, segundo os axiomas da OntoREES, avionico-FAB é considerado um Sistema Embarcado por ter no mínimo um Hardware, Software e Ambiente.

**Quadro 26 – Inferências ABox 3.**

$\text{temRequisitoNF}(\text{software-LAS}, \text{BLR-1}) \models \text{RequisitoNaoFuncional}(\text{BLR-1})$
$\text{temRequisitoNF}(\text{software-LAS}, \text{BLR-2}) \models \text{RequisitoNaoFuncional}(\text{BLR-2})$
$\text{temRequisitoNF}(\text{software-LAS}, \text{BLR-3}) \models \text{RequisitoNaoFuncional}(\text{BLR-3})$
$\text{temRequisitoNF}(\text{software-LAS}, \text{MCR-6}) \models \text{RequisitoNaoFuncional}(\text{MCR-6})$
$\text{temRequisitoNF}(\text{software-LAS}, \text{MCR-7}) \models \text{RequisitoNaoFuncional}(\text{MCR-7})$
$\text{temRequisitoNF}(\text{software-LAS}, \text{MCR-8}) \models \text{RequisitoNaoFuncional}(\text{MCR-8})$
$\text{temRequisitoNF}(\text{software-LAS}, \text{ACR-10}) \models \text{RequisitoNaoFuncional}(\text{ACR-10})$
$\text{temRequisitoNF}(\text{software-LAS}, \text{ACR-11}) \models \text{RequisitoNaoFuncional}(\text{ACR-11})$
$\text{temRequisitoNF}(\text{software-LAS}, \text{ACR-12}) \models \text{RequisitoNaoFuncional}(\text{ACR-12})$
$\text{temRequisitoNF}(\text{software-LAS}, \text{VER-20}) \models \text{RequisitoNaoFuncional}(\text{VER-20})$
$\text{temRequisitoNF}(\text{software-LAS}, \text{VER-21}) \models \text{RequisitoNaoFuncional}(\text{VER-21})$
$\text{temRequisitoNF}(\text{software-LAS}, \text{VER-22}) \models \text{RequisitoNaoFuncional}(\text{VER-22})$

**Fonte: A Autora (2021).**

De acordo com o Quadro 26, podemos inferir que BLR-1, BRL-2, BLR-3, MCR-6, MCR-7, MCR-8, ACR-10, ACR-11, ACR-12, VER-20, VER-21 e VER-22 são requisitos do recurso software-LAS. É importante destacar a importância dessa inferência, pois consegue classificar os requisitos de acordo com cada recurso. Alguns autores e especialistas desse domínio citam que, em projetos de sistemas embarcados, os requisitos precisam ser separados em requisitos de *hardware, software e ambiente* (WOLF, 2012) (KOOPTMAN, 2010).

**Quadro 27 – Inferências ABox 4.**

influencia(RTCA-DO-178-C, avionico-FAB), NormaSoftware(RTCA-DO-178-C)  = Norma(RTCA-DO-178-C)
nivelRequerido(A, software-LAS), Nivel(A)  = A (catastrófica)
garanteConformidade(cleyton, RTCA-DO-178-C)  = EngenheiroCertificacao(cleyton)

**Fonte: A Autora (2021).**

De acordo com as Inferências do Quadro 27, percebe-se que uma norma influencia o sistema embarcado (aviônico-FAB). A norma definida é a RTCA-DO-178-C, essa norma possui objetivos específicos para cada nível, sendo assim, o nível de criticidade A é considerado catastrófico e precisa atender 71 objetivos associados. Ainda não é possível, nas ontologias propostas, trabalhar detalhadamente cada objetivo associado, mas consegue mostrar qual o nível de criticidade. Diante disso, como trabalhos futuros, iremos propor novos conceitos levando em consideração detalhes de algumas normas de sistemas aviônicos. Por fim, na OntoREAV, foi definido um agente Engenheiro de Certificação, esse agente é responsável apenas por garantir a conformidade com as normas. Portanto, nessa inferência Cleyton é considerado um Engenheiro de Certificação porque ele garante a conformidade da norma RTCA-DO-178-C.

**Quadro 28 – Inferências ABox 5.**

causaImpactoPositivo(BLR-2, MCR-6)  = Requisito(BLR-2)
causaImpactoNegativo(ACR-10, R01-Custo)  = Requisito(ACR-10)
causaImpactoNegativoEm(MCR-6, R01-Custo)  = Requisito(MCR-6)

**Fonte: A Autora (2021).**

O Quadro 28 apresenta as inferências para os conceitos *causaImpactoPositivo* e *causaImpactoNegativo*. Os impactos definidos foram classificados de acordo com as respostas

do questionário aplicado na seção 6.2. Nessa consulta é possível identificar que BLR-2 é um requisito e causa impacto positivo no requisito de MCR-6. O requisito MCR-6 causa um impacto negativo no requisito R01-custo, assim como o requisito ACR-10 também causa um impacto negativo em R01-custo. Um fator interessante com a ontologia proposta é a rastreabilidade dos requisitos, é possível rastrear quais requisitos tem impactos em outros (positivos e/ou negativos) e descrever qual a causa. Na ER a rastreabilidade de requisitos é uma parte importante do processo de engenharia de sistemas, pois garante que todos os requisitos foram considerados adequadamente durante cada fase do desenvolvimento, e que não há "brechas" de escopo no sistema desenvolvido devido a requisitos perdidos. Isso também garante que todos os requisitos sejam internamente consistentes uns com os outros.

#### 7.4 SÍNTESE DO CAPÍTULO

Neste capítulo, abordamos os dados referentes a validação/verificação das ontologias. Foram realizados três tipos de validação/verificação, sendo elas: empírica; padrões e anti-padrões e inferências ABox. A primeira foi realizada uma validação com especialista do domínio, a segunda analisando na literatura quais padrões existentes da OntoUML e por fim realizamos uma validação utilizando inferências em ABox, a partir de um cenário real de aviônicos. Com a verificação foi possível verificar que as ontologias são consistentes e que o modelo conceitual está correto. A validação mostrou que as ontologias propostas são aceitáveis aos domínios propostos.

## 8 CONSIDERAÇÕES FINAIS

Este capítulo descreve as considerações finais deste estudo, apresentando as conclusões, limitações do trabalho e propostas para trabalhos futuros.

### 8.1 CONCLUSÕES

Os sistemas embarcados, nos últimos anos, têm sido amplamente utilizados em diversas áreas e o número de aplicações tem crescido consideravelmente. À medida em que novas tecnologias são criadas, esses sistemas se tornam cada vez mais complexos. Porém, a especificação de tais sistemas, muitas vezes, têm sido descuidada, em função da necessidade de agilidade no processo de desenvolvimento e de sua simplicidade aparente. Assim, a má especificação e a ausência de documentação suficiente acabam sendo fatores de insucesso para esses sistemas.

Podemos perceber que, por se tratar de uma área com vários domínios, cada um com características diferentes, e composta por profissionais com um conhecimento maior de *hardware*, mas com baixo aproveitamento pela área de ER, a engenharia de requisitos para especificação de sistemas embarcados é um campo que precisa de uma atenção maior quando comparado aos sistemas de propósito geral. Isso tem levado a comunidade acadêmica a explorar e estabelecer novas abordagens para o desenvolvimento de SE e integrá-las à diferentes áreas do conhecimento, tais como Ontologias.

Ontologias fornecem uniformidade aos conceitos, no que diz respeito à sintaxe e semântica, facilitando a comunicação em diversos domínios. A criação de uma ontologia específica para o domínio de sistemas embarcados pode contribuir para o sucesso do desenvolvimento de tais sistemas, pois apresenta uma formalidade para representar os conceitos internos do domínio e as relações entre eles. Diante da dificuldade em encontrar abordagens de engenharia de requisitos específicas para SE, foi realizado um mapeamento sistemático da literatura, uma revisão sistemática da literatura, um *Survey* e entrevistas com especialistas do domínio, na qual foi possível identificar trabalhos relevantes que ajudaram a responder às questões de pesquisa estabelecidas sobre a temática de pesquisa, propor uma ontologia *core* de requisitos para SE (OntoREES) e uma ontologia de domínio para sistemas embarcados aviônicos (OntoREAV). Portanto, buscamos resolver o problema de como formalizar a representação do conhecimento de requisitos em sistema embarcados e realizar algumas inferências lógicas para simulações de projetos desses sistemas, utilizando os

padrões abertos da Web Semântica. Diante disso, com as ontologias propostas podemos refutar a hipótese apresentada na seção 1.4.4, “a construção de Ontologias para Especificação de Requisitos em Sistemas Embarcados através das tecnologias abertas da Web Semântica contribui para que seja formalizado o conhecimento inerente aos requisitos de sistemas embarcados”. Consequentemente, isso garante que a elicitacão e especificacão de requisitos seja realizada de forma mais coerente e correta, garantindo um maior controle e qualidade ao sistema.

## 8.2 CONTRIBUIÇÕES

Destacam-se como principais contribuicões desse trabalho:

- **Revisão sistemática da Literatura** – a partir da realizacão do MSL (SOUSA et al., 2016) e RSL foi possível identificar o estado da arte sobre ontologia no desenvolvimento de sistemas embarcados. Também, apresenta lacunas na temática de pesquisa, servindo como indicacão de trabalhos futuros para uma pesquisa mais aprofundada, com o intuito de se confirmar as deficiências ou, principalmente, de resolvê-las. Com essas revisões, foi possível compreender o estado da arte e delinear os objetivos desta tese. Também foi possível identificar que não existem ontologias de requisitos específicas para o domínio de sistemas embarcados. Diante disso, acreditamos, que esta tese é pioneira na temática de pesquisa.
- **Survey** – foi elaborado um *survey* com especialistas do domínio. O objetivo foi investigar o processo de engenharia de requisitos no desenvolvimento de sistemas embarcados.
- **Ontologia core de requisitos** – baseado no MSL, RSL e Survey, propomos uma ontologia de requisitos para sistemas embarcados (OntoREES). A ontologia descreve os principais conceitos envolvidos no desenvolvimento de um SE. Nesta, trataram-se conceitos centrais do domínio de SE, tais como: agentes, recursos, normas, requisitos funcionais e não funcionais e os relacionamentos entre esses. A OntoREES mostrou-se adequada para modelar o domínio de SE.
- **Ontologia de domínio** – a partir da OntoREES e conversas com um especialista no domínio de sistema embarcado aviônico, foi proposta uma ontologia de domínio para o setor aviônico (OntoREAV). Essa ontologia ajudou a ratificar a adequabilidade da OntoREES como uma ontologia central que pode ser estendida para conceituar outros

domínios de sistemas embarcados.

- **Validação** – a validação estática que realizamos com a ajuda de um especialista, do domínio de SE, nos forneceu contribuições valiosas para a nossa ontologia e uma versão melhorada foi elaborada (apresentada no 6). A versão melhorada foi usada para criar a ontologia de domínio (OntoREAV), ou seja, serviu como ontologia central para especializar a ontologia para o domínio de sistema embarcado aviônico. Assim como foi criada a OntoREAV, outras ontologias de domínios poderão ser propostas a partir da OntoREES. A validação da OntoREAV contribuiu para levantar *feedback* relevante, corroborando com a ideia de que a ontologia proposta é de boa qualidade e utilizável para o domínio.

### 8.3 LIMITAÇÕES E AMEAÇAS À VALIDADE

Algumas limitações foram percebidas nesta tese. Com relação a Revisão Sistemática da Literatura, uma das principais limitações é os vieses introduzidos pelo pesquisador nos processos de seleção, as imprecisões na extração dos dados e a cobertura da pesquisa. Duas medidas foram tomadas para minimizar estas limitações:

- A elaboração e utilização de um protocolo de pesquisa;
- As fases de busca, seleção e extração dos dados foram realizadas em pares e, depois de finalizada cada etapa, os dados foram confrontados, minimizando o viés do pesquisador e a extração incorreta dos dados.

Acreditamos que a utilização de um protocolo de pesquisa definido antes da realização desta pesquisa e a execução por pares de pesquisadores minimizaram as limitações identificadas.

No que concerne as Ontologias, uma das maiores limitações foi compreender a complexidade da ontologia de fundamentação UFO e o enorme escopo dos Sistemas Embarcados. Diante disso, alguns conceitos importantes relacionados ao domínio podem não ter sido contemplados.

### 8.4 TRABALHOS FUTUROS

A relevância de um trabalho pode ser avaliada também pelas oportunidades de trabalhos futuros que ele provê. Portanto, buscando obter tal importância, a partir deste estudo

identificamos alguns direcionamentos para novas pesquisas na área de ontologias de requisitos para sistemas embarcados:

1. **Ferramenta** – desenvolver uma ferramenta específica para inferências no domínio de sistemas embarcados, utilizando como base de conhecimento as ontologias desenvolvidas na lógica descritiva. Com a ferramenta, será possível ao usuário comum compreender os requisitos, *stakeholders* e normas que fazem parte de um projeto. Além disso, será possível fazer a rastreabilidade dos requisitos com relação aos impactos positivos e negativos que existem entre si.
2. **Anti-Padrões** – um antipadrão é uma decisão de modelagem propensa a erros recorrentes. Diante disso, a partir do catálogo de anti-padrões semânticos proposto por Guizzardi e Sales (SALES, 2014), vamos realizar uma validação das ontologias propostas utilizando esses anti-padrões.
3. **Ontologias de domínios (médico e automotivos)** – concentramos nossa ontologia de domínio na área de sistemas aviônicos, porém outros domínios de sistemas embarcados têm chamado atenção por especialistas do domínio. Portanto, pretendemos propor mais duas ontologias de domínio para área médica e automotiva, pois foram os outros domínios mais citados na RSL e no *Survey*.
4. **Expandir a OntoREAV** – pretendemos expandir a OntoREAV acrescentando novos conceitos e explorar as normas específicas para esse domínio. Percebemos que todas as normas possuem características específicas e elas possuem um impacto representativo no desenvolvimento desses sistemas. Portanto, existe uma necessidade estudar melhor cada norma e propor novos conceitos.

## 8.5 PUBLICAÇÕES

- Em Sousa (2016) apresentamos um mapeamento sistemático da literatura. O objetivo foi identificar se ontologias estão sendo utilizadas para o desenvolvimento de sistemas embarcados. **Use of Ontologies in Embedded Systems: A Systematic Mapping**. 10th International Conference on the Quality of Information and Communications Technology (QUATIC). **Esse artigo ficou entre os seis melhores da conferência.**

- Em Sousa (2020) apresentamos uma revisão sistemática da literatura. **Os Benefícios do Uso de Ontologias em Sistemas Embarcados**. Anais do WER20 - Workshop em Engenharia de Requisitos, São Paulo, Brasil, Agosto 25-28, 2020.
- Em paralelo, outras publicações como co-autora:
  - Pereira, T; Albuquerque, D; Sousa, A, Silva; R, Alencar; and Castro, J. **A Metamodel to Guide a Requirements Elicitation Process for Embedded Systems**. 11th International Conference on the Quality of Information and Communications Technology (QUATIC), 2018.
  - Danyllo Albuquerque, Jaelson Castro, Aeda Sousa. **A Requirements Definition Framework for the Robotic Systems Domain - An Exploratory Study**. Anais do WER18 - Workshop em Engenharia de Requisitos, Rio de Janeiro, Brasil, Setembro 05-06, 2018
  - Pereira, T. C. ; Albuquerque, D. ; Sousa, A. M. C. B. ; Alencar, F. M. R. ; Castro, J. **Retrospective and Trends in Requirements Engineering for Embedded Systems: A Systematic Literature Review**. In: 20th Ibero-American Conference on Software Engineering., 2017, Buenos Aires. Workshop em Engenharia de Requisitos (WER), 2017.
  - Pereira, T. C. ; Albuquerque, D. ; Sousa, A. M. C. B. ; Alencar, F. M. R. ; Castro, J. **A Metamodel for a Requirements Engineering Process of Embedded Systems**. In: Brazilian Symposium on Computing Systems Engineering (SBESC), 2016, João Pessoa. Brazilian Symposium on Computing Systems Engineering (SBESC), 2016.
  - Heineck, T. ; Goncalves, E. ; Sousa, A. M. C. B. ; Oliveira, M. A. ; Castro, J. . **Model-Driven Development in Robotics Domain: a Systematic Literature Review**. In: Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software, 2016, Maringá-PR. Anais do X Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software, 2016.

## REFERÊNCIAS

- ALRUMAIH, Hala; MIRZA, Abdulrahman; ALSALAMAH, Hessah. Domain Ontology for Requirements Classification in Requirements Engineering Context. **IEEE Access**, [S. l.], v. 8, p. 89899–89908, 2020. DOI: 10.1109/ACCESS.2020.2993838.
- ANDRÉ, Oliveira; ANDRADE, Fernando. **Sistemas embarcados: hardware e firmware na prática**. 2. ed. [s.l.: s.n.].
- BAADER, Franz; HORROCKS, Ian; LUTZ, Carsten; SATTLER, Uli. **An Introduction to Description Logic**. Cambridge: Cambridge University Press, 2017. DOI: 10.1017/9781139025355.
- BARCELOS, Pedro Paulo F.; DOS SANTOS, Victor Amorim; SILVA, Freddy Brasileiro; MONTEIRO, Maxwell E.; GARCIA, Anilton Salles. An automated transformation from OntoUML to OWL and SWRL. **CEUR Workshop Proceedings**, [S. l.], v. 1041, n. September 2014, p. 130–141, 2013. DOI: 10.13140/RG.2.1.1007.5923.
- BELGAMO, Anderson. **Um estudo comparativo sobre as principais técnicas de elicitação de requisitos do software**. 2000. Universidade Metodista de Piracicaba, Piracicaba, 2000.
- BENNACEUR, Amel; TUN, Thein Than; YU, Yijun; NUSEIBEH, Bashar. Requirements engineering. *In: Handbook of Software Engineering*. DOI: 10.1007/978-3-030-00262-6\_2.
- BERANDER, Patrik; ANDREWS, Anneliese. Requirements prioritization. *In: Engineering and managing software requirements*. [s.l.] : Springer, 2005. p. 69–94.
- BICCHIERAI, Irene; BUCCI, Giacomo; NOCENTINI, Carlo; VICARIO, Enrico. Integrating metrics in an ontological framework supporting SW-FMEA. *In: INTERNATIONAL WORKSHOP ON EMERGING TRENDS IN SOFTWARE METRICS*, 3., 2012, Zurique, Suíça. **Proceedings** [...]. Zurique: [IEEE], 2012. p. 35–41.
- BORGES RUY, Fabiano; FALBO, Ricardo de Almeida; BARCELLOS, Monalessa Perini; COSTA, Simone Dornelas; GUIZZARDI, Giancarlo. SEON: A software engineering ontology network. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, [S. l.], v. 10024 LNAI, p. 527–542, 2016. DOI: 10.1007/978-3-319-49004-5\_34.
- BOWEN, Judy; HINZE, Annika. Using ontologies to reason about the usability of interactive medical devices in multiple situations of use. *In: PROCEEDINGS OF THE 4TH ACM SIGCHI SYMPOSIUM ON ENGINEERING INTERACTIVE COMPUTING SYSTEMS 2012*, **Anais** [...]. [s.l.: s.n.] p. 247–256.
- BRAUN, Peter; BROY, Manfred; HOUDEK, Frank; KIRCHMAYR, Matthias; MÜLLER, Mark; PENZENSTADLER, Birgit; POHL, Klaus; WEYER, Thorsten. Guiding requirements engineering for software-intensive embedded systems in the automotive industry. **Computer Science - Research and Development**, [S. l.], v. 29, n. 1, p. 21–43, 2010. DOI: 10.1007/s00450-010-0136-y.
- CORCHO, Oscar; FERNÁNDEZ-LÓPEZ, Mariano; GÓMEZ-PÉREZ, Asunción.

Ontological engineering: principles, methods, tools and languages. *In: Ontologies for software engineering and software technology*. [s.l.] : Springer, 2006. p. 1–48.

CORCHO, Oscar; FERNÁNDEZ-LÓPEZ, Mariano; GÓMEZ-PÉREZ, Asunción; LÓPEZ-CIMA, Angel. Building legal ontologies with METHONTOLOGY and WebODE. *In: Law and the semantic web*. [s.l.] : Springer, 2005. p. 142–157.

COSTA, Adriano Ferraz Da. Arandu, um Chatbot para construção de ontologias guiado por uma ontologia de topo. *[S. l.]*, 2020.

DARAMOLA, Olawande; STÅLHANE, Tor; MOSER, Thomas; BIFFL, Stefan. A conceptual framework for semantic case-based safety analysis. *In: EMERGING TECHNOLOGIES & FACTORY AUTOMATION (ETFA), 2011 IEEE 16TH CONFERENCE ON 2011, Anais [...]*. [s.l: s.n.] p. 1–8.

DAUN, Marian; BRINGS, Jennifer; WEYER, Thorsten; TENBERGEN, Bastian. Fostering concurrent engineering of cyber-physical systems: A proposal for an ontological context framework. **2016 3rd International Workshop on Emerging Ideas and Trends in Engineering of Cyber-Physical Systems, EITEC 2016**, *[S. l.]*, p. 5–10, 2016. DOI: 10.1109/EITEC.2016.7503689.

DE ALMEIDA FALBO, Ricardo; BARCELLOS, Monalessa Perini; NARDI, Julio Cesar; GUIZZARDI, Giancarlo. Organizing ontology design patterns as ontology pattern languages. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, *[S. l.]*, v. 7882 LNCS, p. 61–75, 2013. DOI: 10.1007/978-3-642-38288-8-5.

DERMEVAL, Diego; VILELA, Jéssyka; BITTENCOURT, Ig Ibert; CASTRO, Jaelson; ISOTANI, Seiji; BRITO, Patrick; SILVA, Alan. Applications of ontologies in requirements engineering: a systematic review of the literature. **Requirements Engineering**, *[S. l.]*, v. 21, n. 4, p. 405–437, 2016. DOI: 10.1007/s00766-015-0222-6.

DIACONESCU, Ion Mircea; WAGNER, Gerd. Towards a general framework for modeling, simulating and building sensor/actuator systems and robots for the Web of Things. **CEUR Workshop Proceedings**, *[S. l.]*, v. 1319, p. 30–41, 2014.

DYBÅ, Tore; KITCHENHAM, Barbara a; JØRGENSEN, Magne. Evidence-based Software Engineering for Practitioners \* Aim and Methodology of Evidence-Based Software Engineering. **Search**, *[S. l.]*, n. 1325, p. 1–11, 2004.

FALBO, Ricardo A.; GUIZZARDI, Giancarlo; GANGEMI, Aldo; PRESUTTI, Valentina. Ontology patterns: Clarifying concepts and terminology. **CEUR Workshop Proceedings**, *[S. l.]*, v. 1188, 2013.

FONSECA, Claudenir M.; PORELLO, Daniele; GUIZZARDI, Giancarlo; ALMEIDA, João Paulo A.; GUARINO, Nicola. Relations in ontology-driven conceptual modeling. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, *[S. l.]*, v. 11788 LNCS, n. i, p. 28–42, 2019. DOI: 10.1007/978-3-030-33223-5\_4.

GORSCHKEK, Tony; GARRE, Per; LARSSON, Stig; WOHLIN, Claes. A model for technology transfer in practice. **IEEE Software**, *[S. l.]*, v. 23, n. 6, p. 88–95, 2006. DOI:

10.1109/MS.2006.147.

GRAAF, Bas; LORMANS, Marco; TOETENEL, Hans. Engineering : The State of the Practice. **IEEE Computer Society**, [S. l.], p. 61–69, 2003. DOI: 10.1109/MS.2003.1241368.

GRUBER, Tom. Collective knowledge systems: Where the Social Web meets the Semantic Web. **Journal of Web Semantics**, [S. l.], v. 6, n. 1, p. 4–13, 2008. DOI: <https://doi.org/10.1016/j.websem.2007.11.011>. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1570826807000583>.

GUARINO, Nicola; WELTY, Christopher. Ontological analysis of taxonomic relationships. *In: INTERNATIONAL CONFERENCE ON CONCEPTUAL MODELING 2000, Anais [...]*. [s.l: s.n.] p. 210–224.

GUIZZARDI, Giancarlo. Ontological foundations for structural conceptual models. [S. l.], 2005.

GUIZZARDI, Giancarlo. **Ontological foundations for conceptual modeling with applications**. [s.l: s.n.]. v. 7328 LNCS DOI: 10.1007/978-3-642-31095-9\_45.

GUIZZARDI, Giancarlo; ALMEIDA, João Paulo; GUIZZARDI, Renata S. S.; BARCELLOS, Monalessa Perini; FALBO, Ricardo. Ontologias de fundamentação, modelagem conceitual e interoperabilidade semântica. **CEUR Workshop Proceedings**, [S. l.], v. 728, 2011.

GUIZZARDI, Giancarlo; FALBO, Ricardo; GUIZZARDI, Renata S. S. Grounding software domain ontologies in the Unified foundational ontology (UFO): The case of the ODE software process ontology. **Memorias de la 11th Conferencia Iberoamericana de Software Engineering - CIbSE 2008**, [S. l.], n. i, 2008.

GUIZZARDI, Giancarlo; FONSECA, Claudenir M.; ALMEIDA, João Paulo A.; SALES, Tiago Prince; BENEVIDES, Alessandro Botti; PORELLO, Daniele. Types and taxonomic structures in conceptual modeling: A novel ontological theory and engineering support. **Data & Knowledge Engineering**, [S. l.], v. 134, n. May, p. 101891, 2021. DOI: 10.1016/j.datak.2021.101891. Disponível em: <https://doi.org/10.1016/j.datak.2021.101891>.

GUIZZARDI, Giancarlo; FONSECA, Claudenir M.; BENEVIDES, Alessandro Botti; ALMEIDA, João Paulo A.; PORELLO, Daniele; SALES, Tiago Prince. Endurant types in ontology-driven conceptual modeling: Towards ontoUML 2.0. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, [S. l.], v. 11157 LNCS, p. 136–150, 2018. DOI: 10.1007/978-3-030-00847-5\_12.

GUIZZARDI, Giancarlo; SALES, Tiago Prince. Detection, simulation and elimination of semantic anti-patterns in ontology-driven conceptual models. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, [S. l.], v. 8824, p. 363–376, 2014. DOI: 10.1007/978-3-319-12206-9\_30.

GUIZZARDI, Giancarlo; SALES, Tiago Prince. Anti-patterns in Ontology-driven Conceptual Modeling: The Case of Role Modeling in OntoUML. **Ontology Engineering with Ontology Design Patterns: Foundations and Applications**, [S. l.], v. 25, n. August, p. 161, 2016. DOI: 10.3233/978-1-61499-676-7-161. Disponível em:

[https://books.google.com/books?hl=en&lr=lang\\_en&id=9hChDQAAQBAJ&oi=fnd&pg=PA161&dq=%22ontology+of+cyber+security%22+OR+%22ontology+of+cybersecurity%22+OR+%22cybersecurity+ontology%22+OR+%22cyber+security+ontology%22+OR+%22ontology+for+vulnerability%22+OR+](https://books.google.com/books?hl=en&lr=lang_en&id=9hChDQAAQBAJ&oi=fnd&pg=PA161&dq=%22ontology+of+cyber+security%22+OR+%22ontology+of+cybersecurity%22+OR+%22cybersecurity+ontology%22+OR+%22cyber+security+ontology%22+OR+%22ontology+for+vulnerability%22+OR+).

HERRE, Heinrich. General Formal Ontology (GFO): A foundational ontology for conceptual modelling. *In: Theory and applications of ontology: computer applications*. [s.l.] : Springer, 2010. p. 297–345.

HOFTBERGER, Oliver; OBERMAISSER, Roman. Ontology-based runtime reconfiguration of distributed embedded real-time systems. *In: OBJECT/COMPONENT/SERVICE-ORIENTED REAL-TIME DISTRIBUTED COMPUTING (ISORC), 2013 IEEE 16TH INTERNATIONAL SYMPOSIUM ON 2013, Anais [...]*. [s.l.: s.n.] p. 1–9.

HORROCKS, Ian; SATTLER, Ulrike; TOBIES, Stephan. Practical reasoning for expressive description logics. *In: INTERNATIONAL CONFERENCE ON LOGIC FOR PROGRAMMING ARTIFICIAL INTELLIGENCE AND REASONING 1999, Anais [...]*. [s.l.: s.n.] p. 161–180.

JAFER, Shafagh; CHHAYA, Bharvi; DURAK, Umut. OWL ontology to ecore metamodel transformation for designing a domain specific language to develop aviation scenarios. *Simulation Series, [S. l.]*, v. 49, n. 7, p. 23–33, 2017. DOI: 10.22360/springsim.2017.mod4sim.003.

JAFER, Shafagh; CHHAYA, Bharvi; DURAK, Umut. OWL ontology to Ecore metamodel transformation for designing a domain specific language to develop aviation scenarios. *In: Proceedings of the Symposium on Model-driven Approaches for Simulation Engineering*. Society for Computer Simulation International, 2017. p. 3.

JIMENEZ, M.; PALOMERA, Rogelio; COUVERTIER, Isidoro. **Introduction to embedded systems: Using microcontrollers and the MSP430**. [s.l.: s.n.]. DOI: 10.1007/978-1-4614-3143-5.

KITCHENHAM, Barbara. Procedures for performing systematic reviews. **Keele, UK, Keele University, [S. l.], v. 33, n. 2004, p. 1–26, 2004.**

KITCHENHAM, Barbara; BRERETON, O. Pearl; BUDGEN, David; TURNER, Mark; BAILEY, John; LINKMAN, Stephen. Systematic literature reviews in software engineering-- a systematic literature review. *Information and software technology, [S. l.]*, v. 51, n. 1, p. 7–15, 2009.

KOAY, Nigel; KATARIA, Pavandeep; JURIC, Radmilla. Semantic management of nonfunctional requirements in an e-Health system. *Telemedicine and e-Health, [S. l.]*, v. 16, n. 4, p. 461–471, 2010. DOI: 10.1089/tmj.2009.0120.

KOOPMAN, Philip. **Better embedded system software**. [s.l.] : Drumnadrochit Press, 2010.

KRÖTZSCH, Markus; SIMANČÍK, František; HORROCKS, Ian. A description logic primer. *Perspectives on Ontology Learning, [S. l.]*, v. 18, n. June, p. 3–20, 2014.

LIN, Xi; ZHANG, Hehua; GU, Ming. OntCheck: An ontology-driven static correctness checking tool for component-based models. *Journal of Applied Mathematics, [S. l.]*, v.

2013, 2013. DOI: 10.1155/2013/934349.

LIN, Xi; ZHANG, Hehua; GU, Ming. OntCheck: An ontology-driven static correctness checking tool for component-based models. **Journal of Applied Mathematics**, v. 2013, 2013.

LIU, Lin; AOYAMA, Mikio. **Requirements engineering in the Big Data Era**. [s.l: s.n.]. v. 558 DOI: 10.1007/978-3-662-48634-4. Disponível em: <http://link.springer.com/10.1007/978-3-662-48634-4>.

MAFRA, Sn; TRAVASSOS, Gh. Estudos Primários e Secundários apoiando a busca por Evidência em Engenharia de Software. ... / **06–Programa de Engenharia de Sistemas e ...**, [S. l.], p. 32, 2006.

MARQUES, Johnny Cardoso. Uma análise das características de especificação de requisitos de software em normas de ambientes regulados. **Anais do WER 2019 - Workshop em Engenharia de Requisitos**, [S. l.], n. August, 2019.

MARQUES, Johnny; YELISETTY, Sarasuaty; BARROS, Lilian. A Framework for Loadable Airborne Systems. [S. l.], n. 10, p. 1–10, 2020.

MCGUINNESS, Deborah L.; FIKES, Richard; HENDLER, James; STEIN, Lynn Andrea. DAML+ OIL: an ontology language for the Semantic Web. **IEEE Intelligent Systems**, [S. l.], v. 17, n. 5, p. 72–80, 2002.

MEZHUYEV, Vitaliy; SPUTH, Bernhard; VERHULST, Eric. Interacting entities modelling methodology for robust systems design. **Proceedings - 2nd International Conference on Advances in System Testing and Validation Lifecycle, VALID 2010**, [S. l.], p. 75–80, 2010. DOI: 10.1109/VALID.2010.24.

MIAN, Paula; CONTE, Tayana; NATALI, Ana; BIOLCHINI, Jorge; TRAVASSOS, Guilherme. A Systematic Review Process for Software Engineering. **Empirical Software Engineering**, [S. l.], v. 32, p. 1–6, 2007. DOI: 10.1145/1241572.1241584.

MUSAVI, Seyyedeh Atefeh; HASHEMI, Mahmoud Reza. An Ontology-Based Method for HW/SW Architecture Reconstruction. **IEEE Transactions on Computers**, [S. l.], v. 68, n. 7, p. 1007–1018, 2019. DOI: 10.1109/TC.2019.2895329.

NEGRI, Elisa; FUMAGALLI, Luca; GARETTI, Marco; TANCA, Letizia. Requirements and languages for the semantic representation of manufacturing systems. **Computers in Industry**, [S. l.], v. 81, p. 55–66, 2016. DOI: 10.1016/j.compind.2015.10.009. Disponível em: <http://dx.doi.org/10.1016/j.compind.2015.10.009>.

NETO, Dario Almudi. A Requirements Specification Template of a Communication Network Based on CAN Protocol to Automotive Embedded Systems. [S. l.], v. 10, n. 3, p. 143–150, 2010.

OLIVEIRA, Ítalo José da Silva. DIREITO, LÓGICA E INTELIGÊNCIA ARTIFICIAL: por quê, como e em que medida automatizar a solução judicial de conflitos no Brasil. [S. l.], 2019.

OSHANA, Robert; KRAELING, Mark. **Software engineering for embedded systems:**

**Methods, practical techniques, and applications.** [s.l.] : Newnes, 2019.

OSSADA, Jaime. GERSE – GUIA DE ELICITAÇÃO DE REQUISITOS PARA SISTEMAS EMBARCADOS DE PEQUENO E MÉDIO PORTE. [S. l.], 2010.

OSSADA, Jc; MARTINS, Leg. Um Estudo de Campo sobre o Estado da Prática da Elicitação de Requisitos em Sistemas Embarcados. [S. l.], p. 12, 2010.

OSSADA, Jc; MARTINS, Leg; RANIERI, Bs; BELGAMO, A. GERSE: Guia de Elicitação de Requisitos para Sistemas Embarcados. [S. l.], 2012.

PAI, Madhukar; MCCULLOCH, Michael; GORMAN, Jennifer D.; PAI, Nitika; ENANORIA, Wayne; KENNEDY, Gail; THARYAN, Prathap; COLFORD JR., J. M.; COLFORD, John M. Systematic reviews and meta-analyses: an illustrated, step-by-step guide. **Natl.Med J India**, [S. l.], v. 17, n. 2, p. 86–95, 2004.

PEASE, Adam. **Ontology: A practical guide.** [s.l.] : Articulate Software Press, 2011.

PEREIRA, Tarcisio; SOUSA, Aeda; OLIVEIRA, Reinaldo; ALBUQUERQUE, Deivson; ALENCAR, Fernanda; CASTRO, Jaelson. A metamodel to guide a requirements elicitation process for embedded systems. **Proceedings - 2018 International Conference on the Quality of Information and Communications Technology, QUATIC 2018**, [S. l.], p. 101–109, 2018. DOI: 10.1109/QUATIC.2018.00023.

PFLEEGER, Shari Lawrence; KITCHENHAM, Barbara A. Principles of survey research: part 1: turning lemons into lemonade. **ACM SIGSOFT Software Engineering Notes**, [S. l.], v. 26, n. 6, p. 16–18, 2001.

POHL, Klaus. **Requirements engineering fundamentals: a study guide for the certified professional for requirements engineering exam-foundation level-IREB compliant.** [s.l.] : Rocky Nook, Inc., 2016.

PRESSMAN, Roger; MAXIM, Bruce. **Engenharia de Software-8ª Edição.** [s.l.] : McGraw Hill Brasil, 2016.

PRESTES, Edson et al. Towards a core ontology for robotics and automation. **Robotics and Autonomous Systems**, [S. l.], v. 61, n. 11, p. 1193–1204, 2013. DOI: 10.1016/j.robot.2013.04.005.

PROVDANOV, Cleber Cristiano; FREITAS, Ernani Cesar De. **Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico** Novo Hamburgo: Feevale, 2013.

RAUTENBERG, Sandro; TODESCO, Jose Leomar; STEIL, Andrea V.; GAUTHIER, Fernando A. O. Uma Metodologia para o Desenvolvimento de Ontologias A Methodology for the Development of Ontologies. **Revista Ciências Exatas e Naturais**, [S. l.], v. 10, n. 2, p. 237–262, 2008.

REITER, Raymond. On Closed World Data Bases. *In*: GALLAIRE, Hervé; MINKER, Jack (org.). **Logic and Data Bases**. Boston, MA: Springer US, 1978. p. 55–76. DOI: 10.1007/978-1-4684-3384-5\_3. Disponível em: [https://doi.org/10.1007/978-1-4684-3384-5\\_3](https://doi.org/10.1007/978-1-4684-3384-5_3).

RIBEIRO, Lcm; RAMOS, Cs. **Definição de um processo de Engenharia de Requisitos para software embarcado na indústria automotiva baseada em uma Arquitetura de Processos de VII Workshop Anual do ...**, 2011.

RODRIGUES, Cleyton Mário de Oliveira (Universidade Federal de Pernambuco-UFPE). **UMA ABORDAGEM ONTOLÓGICA PARA SIMULAÇÃO DE AÇÃO LEGAL E CONSISTÊNCIA SEMÂNTICA APLICADA À Recife**. [S. l.], 2019.

ROUSSEY, Catherine; PINET, Francois; KANG, Myoung Ah; CORCHO, Oscar. An introduction to ontologies and ontology engineering. *In: Ontologies in Urban development projects*. [s.l.] : Springer, 2011. p. 9–38.

SAEED, Nazeer T. Mohamme.; WEBER, Christian; MALLAK, Ahlam; FATHI, Madjid; OBERMAISSER, Roman; KUHNERT, Klaus Dieter. ADISTES ontology for active diagnosis of sensors and actuators in distributed embedded systems. **IEEE International Conference on Electro Information Technology**, [S. l.], v. 2019- May, p. 572–577, 2019. DOI: 10.1109/EIT.2019.8834013.

SALES, Tiago Prince. **Ontology Validation for Managers**. 2014. UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO, [S. l.], 2014.

SAMPAIO, R. ..; MANCINI, M. .. Estudos de revisão sistemática : um guia para síntese. **Revista Brasileira de Fisioterapia**, [S. l.], v. 11, n. 1, p. 83–89, 2007.

SANTANDER, Victor. Integrando Modelagem Organizacional com Modelagem Funcional. [S. l.], 2002.

SCHMIT, Matt; BRICENO, Simon; COLLINS, Kyle; MAVRIS, Dimitri; LYNCH, Kevin; BALL, George. Semantic design space refinement for model-based systems engineering. **10th Annual International Systems Conference, SysCon 2016 - Proceedings**, [S. l.], 2016. DOI: 10.1109/SYSCON.2016.7490579.

SEVERINO, Antônio Joaquim. **Metodologia do trabalho científico**. [s.l.] : Cortez editora, 2017.

SOMMERVILLE, Ian. **Engenharia de Software**, 2007.

SOMMERVILLE, Ian. **Engenharia de Software**. edSão Paulo, SP: Pearson Prentice Hall, , 2011.

SOMMERVILLE, Ian; KOTONYA, Gerald. Requirements engineering: processes and techniques. [S. l.], 1998.

SOUSA, A.; COUTO, T.; AGRA, C.; ALENCAR, F. Use of Ontologies in Embedded Systems: A Systematic Mapping. *In: 2016 10TH INTERNATIONAL CONFERENCE ON THE QUALITY OF INFORMATION AND COMMUNICATIONS TECHNOLOGY (QUATIC) 2016*, **Anais [...]**. [s.l.: s.n.] p. 1–8. DOI: 10.1109/QUATIC.2016.011.

SOUSA, Aêda; AGRA, Celso; MELO, Josenildo; ALENCAR, Fernanda. Elicitação e Especificação de Requisitos em Sistemas Embarcados : Uma Revisão Sistemática. [S. l.], 2015.

STARON, Mirosław. Requirements engineering for automotive embedded systems. *In: Automotive Systems and Software Engineering*. [s.l.] : Springer, 2019. p. 11–28.

ŠTUIKYS, Vytautas; DAMAŠEVIČIUS, Robertas. DESIGN OF ONTOLOGY-BASED GENERATIVE COMPONENTS USING ENRICHED FEATURE DIAGRAMS AND METAPROGRAMMING. *Information Technology and Control*, [S. l.], v. 37, n. 4, 2015.

SURYAWANSHI, Yogita; QIU, Haonan; AYARA, Adel; GLIMM, Birte. An ontological model for map data in automotive systems. **Proceedings - IEEE 2nd International Conference on Artificial Intelligence and Knowledge Engineering, AIKE 2019**, [S. l.], p. 140–147, 2019. DOI: 10.1109/AIKE.2019.00034.

TAN, He; ISMAIL, Muhammad; TARASOV, Vladimir; ADLEMO, Anders; JOHANSSON, Mats. Development and Evaluation of a Software Requirements Ontology. **SKY 2016 - 7th International Workshop on Software Knowledge, Proceedings - In conjunction with IC3K 2016**, [S. l.], p. 11–18, 2016. DOI: 10.5220/0006079300110018.

THAMBOULIDIS, Kleanthis C.; DOUKAS, George S.; KOUMOUTSOS, Giannis V. Device modeling for a flexible embedded systems development process. **Proceedings - 10th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing, ISORC 2007**, [S. l.], p. 337–342, 2007. DOI: 10.1109/ISORC.2007.30.

TONIOLO, Cristiano. PRRESE – PROCESSO DE REUSO DE REQUISITOS NÃOFUNCIONAIS PARA SISTEMAS EMBARCADOS USANDO O NFRFRAMEWORK. [S. l.], 2012.

TRAVASSOS, Guilherme Horta; GUROV, Dmytro; AMARAL, EAGG. Introdução à engenharia de software experimental. [S. l.], 2002.

VASILEVSKAYA, Maria; GUNAWAN, Linda Ariani; NADJM-TEHRANI, Simin; HERRMANN, Peter. Integrating security mechanisms into embedded systems by domain-specific modelling. **Security and Communication Networks**, [S. l.], v. 7, n. 12, p. 2815–2832, 2013.

VASILEVSKAYA, Maria; NADJM-TEHRANI, Simin. Support for cross-domain composition of embedded systems using MARTE models. **ACM SIGBED Review**, [S. l.], v. 12, n. 1, p. 37–45, 2015. DOI: 10.1145/2752801.2752806.

VENTICINQUE, Alessio; MAZZOCCA, Nicola; VENTICINQUE, Salvatore. A Semantic Support for Testing Activities of Safety-Critical Embedded Systems. *In: COMPLEX, INTELLIGENT AND SOFTWARE INTENSIVE SYSTEMS (CISIS), 2014 EIGHTH INTERNATIONAL CONFERENCE ON 2014*, **Anais [...]**. [s.l: s.n.] p. 576–581.

WOHLIN, Claes; RUNESON, Per; HÖST, Martin; OHLSSON, Magnus C.; REGNELL, Björn; WESSLÉN, Anders. **Experimentation in software engineering**. [s.l: s.n.]. v. 9783642290 DOI: 10.1007/978-3-642-29044-2.

WOLF, Marilyn. **Computers as components: principles of embedded computing system design**. [s.l.] : Elsevier, 2012.

YEN, I. Ling; GOLUGURI, Jayabharath; BASTANI, Farokh; KHAN, Latifur; LINN, John. A component-based approach for embedded software development. **Proceedings - 5th IEEE**

**International Symposium on Object-Oriented Real-Time Distributed Computing, ISORC 2002**, [S. l.], p. 402–410, 2002. DOI: 10.1109/ISORC.2002.1003805.

YU, Liyang. **A developer's guide to the semantic Web**. [s.l.] : Springer Science & Business Media, 2011.

ZESHAN, Furkh et al. Ontology-based service discovery framework for dynamic environments. **IET Software**, [S. l.], v. 11, n. 2, p. 64–74, 2017. DOI: 10.1049/iet-sen.2016.0048.

TAN, He et al. Development and evaluation of a software requirements ontology. In: **7th International Workshop on Software Knowledge-SKY 2016 in conjunction with the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management-IC3K 2016**. 2016.

THAMBOULIDIS, Kleanthis C.; DOUKAS, George S.; KOUMOUTSOS, Giannis V. Device Modeling for a Flexible Embedded Systems Development Process. In: **null**. IEEE, 2007. p. 337-343.

VASILEVSKAYA, Maria et al. Integrating security mechanisms into embedded systems by domain-specific modelling. **Security and Communication Networks**, v. 7, n. 12, p. 2815-2832, 2014.

VASILEVSKAYA, Maria; NADJM-TEHRANI, Simin. Support for cross-domain composition of embedded systems using MARTE models. **ACM SIGBED Review**, v. 12, n. 1, p. 37-45, 2015.

VENTICINQUE, Alessio; MAZZOCCA, Nicola; VENTICINQUE, Salvatore. A semantic support for testing activities of safety-critical embedded systems. In: **Complex, Intelligent and Software Intensive Systems (CISIS), 2014 Eighth International Conference on**. IEEE, 2014. p. 576-581.

YEN, I-Ling et al. A component-based approach for embedded software development. In: **Object-Oriented Real-Time Distributed Computing, 2002.(ISORC 2002). Proceedings. Fifth IEEE International Symposium on**. IEEE, 2002. p. 402-410.

ZESHAN, Furkh et al. Ontology-based service discovery framework for dynamic environments. **IET Software**, v. 11, n. 2, p. 64-74, 2017.

ZHAN, Fengjie et al. A Kind of Safety Requirements Description Method of the Embedded Software Based on Ontology. In: **Requirements Engineering in the Big Data Era**. Springer, Berlin, Heidelberg, 2015. p. 126-134.

## APÊNDICE A – PROTOCOLO DE PESQUISA DA REVISÃO SISTEMÁTICA

PROTOCOLO DE REVISÃO SISTEMÁTICA (PRS)		Versão: 02
Foco da pesquisa	O foco desta pesquisa está em analisar o uso de ontologias no desenvolvimento de sistemas embarcados.	
Amplitude da revisão sistemática	<b>Questão de pesquisa</b> Como as ontologias auxiliam o desenvolvimento de sistemas embarcados?	
	<b>Palavras-chaves relacionadas às questões de pesquisa</b> Devido ao foco da pesquisa abranger um conteúdo de relevância mundial, evidentemente as palavras-chaves serão formuladas no idioma universal Inglês. As palavras-chave foram definidas e devidamente organizadas de acordo com os termos encontrados no foco da pesquisa. Ao final foram definidas 7 (sete) palavras-chave para a pesquisa.	
	<ul style="list-style-type: none"> <li>• ontology , ontologies</li> <li>• embedded systems, Safety Critical Systems, Real Time Systems, Embedded Product.</li> </ul>	
	<b>Intervenção</b> A pesquisa pretende intervir nas ontologias e sistemas embarcados.	
<b>Efeito</b> Apresentar evidências científica que comprovem a necessidade do uso de ontologias para melhoria no processo de desenvolvimento dos sistemas embarcados.		
<b>Métricas de Resultados</b> Pelo menos cinco trabalhos relacionados à Ontologias específicas para SE. Pelo menos cinco trabalhos relacionados aos benefícios das ontologias em SE.		
<b>População</b> A população aplicável a esta pesquisa pode ser resumida em estudos primários.		
<b>Aplicação</b> Será aplicada a partir do protocolo de pesquisa onde será revisado por dois pesquisadores.		
Seleção das fontes de pesquisa	<b>Crítérios de inclusão e exclusão</b> Os seguintes critérios devem nortear a inclusão e exclusão das fontes de busca dos trabalhos. <ul style="list-style-type: none"> <li>• As fontes devem estar na web, com exceção apenas de livros que podem ser impressos;</li> <li>• As fontes devem disponibilizar os trabalhos na íntegra e gratuitamente para fins de pesquisa;</li> <li>• As fontes devem possuir mecanismos avançados de busca que permitam a combinação de palavras-chave com os termos relacionais “AND” e “OR”;</li> <li>• As fontes devem ser de renome científico-acadêmico mundial, com exceção de sites web de universidades, caso seja necessário, que contenham os mecanismos de busca exigidos.</li> </ul>	
	<b>Procedimentos para seleção das fontes</b> As fontes serão selecionadas por meio de testes com as palavras-chave já citadas. Caso retornem resultados satisfatórios ao teste, elas serão incluídas, ao contrário serão excluídas (descartadas).	
	<b>Idiomas de estudos</b> O idioma aplicado na pesquisa e observação de trabalhos deste estudo foi o Inglês.	

	<b>Identificação das fontes</b>	<p><b>Método de pesquisa:</b> a busca por trabalhos será realizada de forma eletrônica, através de mecanismos de busca de sites web especializados e de renome científico-acadêmico, podendo ser utilizados também sites de universidades que contenham esses mecanismos disponíveis;</p> <p><b>Strings de busca:</b> a strings ou frase de busca é baseada nas palavras-chave já citadas. Essa string será aplicada de acordo com a disponibilidade técnica de estratégia de busca do mecanismo a ser utilizado, podendo sofrer pequenas adaptações para que o mecanismo consiga executá-las. A strings é a seguintes:</p> <p><b>Inglês:</b>  (1) (("ontology" OR "ontologies") AND ("embedded system" OR "embedded software" OR "real time system" OR "safety critical systems" OR "embedded product"))</p> <p><b>Lista de fontes de busca</b>  Os seguintes sites com mecanismos de busca serão inicialmente utilizados para pesquisa dos trabalhos:</p> <ul style="list-style-type: none"> <li>• <i>ACM Digital Library</i></li> <li>• <i>IEEE Xplore</i></li> <li>• <i>Web of Science</i></li> <li>• <i>ScienceDirect;</i></li> <li>• <i>Scopus</i></li> </ul>
Seleção dos trabalhos pesquisados	<p><b>Definição dos trabalhos</b>  Os seguintes critérios devem nortear a inclusão e exclusão dos trabalhos.</p> <ul style="list-style-type: none"> <li>• Devem ser exclusivamente advindos da internet</li> <li>• Estudos publicados entre 2000 a 2019</li> <li>• Os trabalhos devem estar disponibilizados em sites web especializados e de relevância científico-acadêmico</li> <li>• Os trabalhos devem estar disponibilizados por completo</li> <li>• Os trabalhos devem demonstrar algum embasamento científico que comprove os seus resultados</li> </ul> <p><b>Procedimentos para seleção dos trabalhos</b>  Serão aplicadas cinco etapas para a seleção definitiva dos estudos a serem avaliados no contexto da pesquisa. Os seguintes procedimentos vão proporcionar filtrar os trabalhos mais relevantes para a pesquisa:</p> <ul style="list-style-type: none"> <li>• ETAPA I – Realizar pesquisas de acordo com as strings de busca definidas para a pesquisa. Em seguida o pesquisador verificará a data de publicação de todos os trabalhos. Os trabalhos que estiverem dentro do período de data especificado serão selecionados, os demais serão descartados.</li> <li>• ETAPA II – O pesquisador irá ler independentemente o título e o resumo de todos os trabalhos selecionados no primeiro passo. Depois de lidos, serão selecionados para a próxima etapa apenas os trabalhos relevantes e condizentes com o tema, os demais serão descartados.</li> <li>• ETAPA III – Nessa etapa o pesquisador irá ler independentemente a introdução e a conclusão dos trabalhos. Em seguida serão selecionados os trabalhos relevantes, os demais serão descartados.</li> <li>• ETAPA IV – Nessa Etapa os trabalhos selecionados até agora serão lidos completamente pelo pesquisador.</li> <li>• ETAPA V – Nesta última etapa os trabalhos lidos serão considerados aptos (depois de aplicados os critérios de qualidade) e será fichado no formulário de aprovação dos estudos.</li> </ul>	

## APÊNDICE B – E-MAIL

Dear,

I am a doctoral student in Electrical Engineering at UFPE (Department of electronics and systems) and I am developing an academic research about the requirements engineering in the process of developing Embedded Systems.

Your collaboration is extremely important for the conclusion of the ongoing study.

The survey can be accessed at the link below:

[\*\*SURVEY LINK\*\*](#)

I thank you for the attention, ensuring the ethical commitment to preserve the collected information through the applied survey.

I am available for any information!

Best regards,

Aêda Sousa

## APÊNDICE C – CARTA CONVITE

### **Requirements Engineering in the Development of Embedded Systems**

You are invited to participate in this survey to investigate the requirements engineering process in the development of embedded systems. Please read this form carefully and feel free to ask the researchers any questions you may have.

#### **Researchers:**

Aêda Sousa – Federal University of Pernambuco (UFPE) – Department of Electronics and Systems. – Brazil - [aedasousa@gmail.com](mailto:aedasousa@gmail.com)

Fernanda Alencar – Federal University of Pernambuco (UFPE) – Department of Electronics and Systems – Brazil - [fernandaalenc@gmail.com](mailto:fernandaalenc@gmail.com)

**Confidentiality:** by participating in this study, you acknowledge and agree that your responses and contributions will be anonymously recorded for the purpose of a more reliable data analysis.

**Dissemination of results:** the aggregate results of this study will appear in articles published in conferences and scientific journals.

**Right to Withdraw:** the participation in this study is voluntary and you may decide not to participate at any time or choose not to answer any questions you do not feel comfortable. The survey responses will remain anonymous. As the search is anonymous, it cannot be removed once it is submitted.

**Questions:** if you have any questions about the study, do not hesitate to ask at any point, you can also contact the researchers if you have questions at a later time.

**Follow-up:** if you would like to know the results of this study, you can contact the researchers using the contact information provided above.

**Consent to Participate:** by completing and submitting this survey, your free and informed consent is implied and indicates that you understand the conditions of participation above in this study.

\*The average time to answer this questionnaire is 20 minutes.

## APÊNDICE D – QUESTIONÁRIO

### Definition of the Requirements in Embedded Systems

#### Understanding (elicitation) the requirements

1. Are the requirements lifted at the beginning of an embedded systems project? (Yes / No)
2. Who is responsible for doing this?
3. Do customers participate in this definition? (Yes / No)
4. Do you use any form to assist in the requirements definition process? (Yes / No)  
if the answer for **No**:
  - What is the reason for no using it?
5. What form to understand (elicitation) requirements is used?
6. In which process stages of the requirements definition process is it applied? How is it used?
7. Is there any way to define the hardware requirements? How is this form?
8. Is there any way to define the software requirements? How is this form?
9. Is there any way to define the user requirements? How is this form?
10. Is there any way to define the system requirements? How is this form?
11. Is there any way to define the reliability requirements? How is this form?
12. Is there any way to define the security requirements? How is this form?

#### Requirements Analysis

13. How are the inconsistencies in raised requirements or absences requirements analyzed?
14. How is the completeness of the raised requirements analyzed?
15. Are there any models for representing the requirements that compose the scope of what will be developed?
16. Is there any procedure for implementing the analysis of the defined requirements? What is this procedure?
17. How are the set of requirements that will be part of the project scope negotiated?
18. Is there any procedure for prioritizing the requirements that will compose a version of the project?

### **Requirements Specification**

19. How are the requirements to be implemented documented? How are documented the requirements that will be implemented?
20. Is any computational tool used to support the requirements record?
21. Who reads the generated documentation?
22. Is there any management over the versions of this document?

### **Verification and Validation of requirements**

23. Are the raised requirements and that compose the elements of the project scope verified?
24. How are the raised requirements and that compose the elements of the project scope verified?
25. Are the raised requirements and that compose the elements of the project scope validated/approved with the client/final user?
26. How are the raised requirements and that compose the elements of the project scope validated/approved?

### **Assessment (Evaluation) of Requirements**

27. Are the requirements changes throughout the development project managed and implemented)? Is there any procedure?
28. Is the evolution of the requirements thought out?
29. Is there any procedure to deal with the evolution of the requirements?

### **Requirements**

30. Do you think it is important to define the requirements in the development of embedded systems? (Yes / No) – Justify
31. What are the main or fundamental requirements to be considered in the development of embedded systems?
  - Name of requirements, Description e Type (functional e nonfunctional)

32. Is the system delivered within the stipulated time? Deadline set

- Over 80%
- Between 61% and 80%
- Between 40 and 60%
- Below 40%

33. Is the system delivered at the originally planned cost?

- Over 80%
- Between 61% and 80%
- Between 40 and 60%
- Below 40%

## APÊNDICE E – GLOSSÁRIO DE TERMOS

<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>
Altura	Expressa a altura em metro e centímetro	<b>Conceito</b>
Ambiente	local onde acontece a operação do sistema	<b>Conceito</b>
Analógico/Digital	dispositivo que converte valores de um sinal analógico em sinal digital	<b>Conceito</b>
Atuador	componente físico que recebe comandos do sistema para atuar no contexto do sistema embarcado	<b>Conceito</b>
Bateria	coleção de células de energia elétrica que fornece energia para o sistema e componentes	<b>Conceito</b>
Botões	componente físico de entrada de dados utilizado para interface com o usuário	<b>Conceito</b>
CarregamentoSoftware	Conceito relacionado a atualização do software	<b>Conceito</b>
causaImpactoNegativo	associação que relaciona <i>RequisitoQualidade</i> a <i>RequisitoFuncional</i> e <i>Fisico</i> em <i>RequisitoQualidade</i> . Essa associação pode definir se um requisito causa impacto negativo em outro requisito.	<b>Relação</b>
causaImpactoPositivo	associação que relaciona <i>RequisitoQualidade</i> a <i>RequisitoFuncional</i> e <i>Fisico</i> em <i>RequisitoQualidade</i> . Essa associação pode definir se um requisito causa impacto positivo em outro requisito.	<b>Relação</b>
Componentes	partes físicas que compõem o sistema embarcado	<b>Conceito</b>

Comunicação	consiste na forma de comunicação entre os componentes	<b>Conceito</b>
Confiabilidade	capacidade do sistema de fornecer serviços conforme especificado	<b>Conceito</b>
ConsumoEnergia	definem estratégias para o gerenciamento e consumo de energia do sistema e componentes	<b>Conceito</b>
Conversores	dispositivos que convertem valores de um sensor para atuador	<b>Conceito</b>
Custo	define o custo monetário do sistema e componentes	<b>Conceito</b>
define	associação que relaciona um <i>Stakeholder</i> a um <i>Requisito</i>	<b>Relação</b>
Desempenho	tempo de resposta e taxa de vazão dos dados, que refletem a capacidade de processamento do sistema	<b>Conceito</b>
Digital/Analógico	dispositivo que converte valores de um sinal digital em sinal analógico	<b>Conceito</b>
DisplayLCD	interface de comunicação visual. Utiliza o número de caracteres por linhas e o número de linhas para a comunicação	<b>Conceito</b>
DisplaySeteSegmentos	interface de comunicação visual	<b>Conceito</b>
éImpactadoNegativamentePor	Associação que relaciona os impactos negativos entre os requisitos	<b>Relação</b>
éImpactadoPositivamentePor	Associação que relaciona os impactos positivos entre os requisitos	<b>Relação</b>
éInfluenciadoPor	Associação que relaciona <i>Hardware</i> , <i>Software</i> e <i>SistemaEmbarcado</i> a Norma	<b>Relação</b>
éRecursoDe		<b>Relação</b>
éRequisito	associação que define um <i>Requisito</i>	<b>Relação</b>
Físico	descreve os requisitos físicos do	<b>Conceito</b>

	sistema, ou seja, os componentes	
foiDefinidoPor	associa um <i>Requisito</i> ao <i>Stakeholder</i> que o define	<b>Relação</b>
Hardware	equipamento físico usado para processar, armazenar ou transmitir programas ou dados de computador	<b>Conceito</b>
influencia	Associação que relaciona uma <i>Norma</i> a <i>SistemaEmbarcado</i>	<b>Relação</b>
InterfaceUsuarioEntrada	responsável pela entrada de dados	<b>Conceito</b>
InterfaceUsuarioSaída	responsável pela saída dos dados	<b>Conceito</b>
Largura	Expressa a largura em metro e centímetro	<b>Conceito</b>
Manutenibilidade	capacidade do sistema ser reparado ou substituído	<b>Conceito</b>
MemoriaDados	local onde serão armazenados os registros, as variáveis e todos os espaços reservados para o processamento temporário	<b>Conceito</b>
MemoriaPrograma	local onde fica armazenado o software do sistema embarcado	<b>Conceito</b>
Microcontrolador	pequeno computador em um único circuito integrado	<b>Conceito</b>
Microprocessador	executa as instruções e cálculos que constituem os programas	<b>Conceito</b>
Norma	procedimentos necessários para que o fabricante de um sistema embarcado demonstre que seu produto foi construído atendendo aos requisitos.	<b>Conceito</b>
NormaHardware	Normas relacionadas a hardware	<b>Conceito</b>
NormaSistemaEmbarcado	Normas relacionadas a Sistema Embarcado	<b>Conceito</b>
NormaSoftware	Normas relacionadas a software	<b>Conceito</b>
Paralela	consiste na transferência de dado de	<b>Conceito</b>

	oito <i>bits</i> (um <i>byte</i> ) independentes, ou seja, existem oito pinos para o envio destes simultaneamente	
Peso	peso adequado para o sistema e componentes	<b>Conceito</b>
Pessoa	é um tipo rígido	<b>Conceito</b>
Portabilidade	capacidade do sistema ser executado sob diferentes sistemas de computação, hardware, software ou combinação de ambos	<b>Conceito</b>
Privacidade	proteção de informações e recursos do sistema contra pessoas não autorizadas	<b>Conceito</b>
Qualidade	descreve as restrições que afeta a qualidade do hardware e do software	<b>Conceito</b>
RequisitoFuncional	descreve uma função do sistema ao nível de hardware e software	<b>Conceito</b>
RequisitoNaoFuncional	descreve as restrições do sistema	<b>Conceito</b>
Requisitos	descreve uma funcionalidade do sistema e envolve requisitos funcionais e não funcionais	<b>Conceito</b>
RequisitosEstruturaFisica	descreve requisitos do ambiente relacionados a temperatura, peso, dimensões e umidade	<b>Conceito</b>
Segurança	capacidade do sistema continuar operando mesmo após uma falha ocorrer	<b>Conceito</b>
Sensor	componente físico que fornece informações atualizadas sobre o contexto em que o sistema está sendo executado	<b>Conceito</b>
Serial	consiste no envio de bits em fila, ou seja, um bit de dados de cada vez. Possui dois canais para transferência de	<b>Conceito</b>

	dados, envio e recebimento	
SistemaEmbarcado	Sistema que executa uma tarefa específica e que é composto por hardware, software e ambiente	<b>Conceito</b>
Software	sequência de instruções escritas para serem interpretadas com o objetivo de executar tarefas específicas. É a parte lógica responsável por mandar instruções para o hardware.	<b>Conceito</b>
Stakeholder	Qualquer pessoa física que define os requisitos	<b>Conceito</b>
StakeholderHardware	pessoa física que seja específica de hardware	<b>Conceito</b>
StakeholderOrganizacao		<b>Conceito</b>
StakeholderSoftware	pessoa física que seja específica de software	<b>Conceito</b>
StakeholderTime		<b>Conceito</b>
TaxaOcorrenciaFalha		
Teclado	componente físico de entrada de dados utilizado para interface com o usuário	<b>Conceito</b>
Temperatura	grandeza física que exprime o nível de calor que o ambiente do sistema embarcado suporta	<b>Conceito</b>
TempoDeResposta	o tempo entre uma ação de entrada de dados no sistema e sua saída	<b>Conceito</b>
TempoMedioFalha	capacidade de um sistema continuar operando sem interrupção quando um ou mais de seus componentes falham	<b>Conceito</b>
temRecursoAmbiente	associação que relaciona <i>SistemaEmbarcado</i> ao recurso <i>Ambiente</i>	<b>Relação</b>
temRecursoHardware	associação que relaciona <i>SistemaEmbarcado</i> ao recurso	<b>Relação</b>

	<i>Hardware</i>	
temRecursoSoftware	associação que relaciona <i>SistemaEmbarcado</i> ao recurso <i>Software</i>	<b>Relação</b>
temRequisito	associação que relaciona <i>Ambiente</i> , <i>Hardware</i> e <i>Software</i> a <i>Requisito</i>	<b>Relação</b>
temRequisitoEstruturaFisica	associação que relaciona o <i>Ambiente e Componentes</i> a <i>RequisitoEstruturaFisica</i>	<b>Relação</b>
temRequisitoF	Associação que vincula um recurso ( <i>Ambiente</i> , <i>Hardware</i> , <i>Software</i> ) especificamente a <i>RequisitosFuncionais</i>	<b>Relação</b>
temRequisitoNF	Associação que vincula um recurso ( <i>Ambiente</i> , <i>Hardware</i> , <i>Software</i> ) especificamente a <i>RequisitosNãoFuncionais</i>	<b>Relação</b>
Umidade	quantidade de vapor de água presente no ambiente	<b>Conceito</b>
Usabilidade	interação do usuário final com o sistema e o esforço necessário para aprender, operar, preparar a entrada e interpretar a saída do sistema	<b>Conceito</b>
Wireless	consiste na forma de comunicação através de uma rede sem fio	<b>Conceito</b>