



MICHAEL TADEU ALVES DE OLIVEIRA

**ESPECIFICAÇÃO DE REQUISITOS DE SOFTWARE
UTILIZANDO PADRÕES NO CONTEXTO DA ENGENHARIA
DE REQUISITOS ORIENTADA A ASPECTOS**

**LAVRAS-MG
2020**

MICHAEL TADEU ALVES DE OLIVEIRA

**ESPECIFICAÇÃO DE REQUISITOS DE SOFTWARE
UTILIZANDO PADRÕES NO CONTEXTO DA ENGENHARIA
DE REQUISITOS ORIENTADA A ASPECTOS**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, área de concentração em Banco de Dados e Engenharia de Software, para a obtenção do título de Mestre.

Prof. Dr. Heitor Augustus Xavier Costa
Orientador

Prof. Dr. Paulo Afonso Parreira Júnior
Coorientador

**LAVRAS-MG
2020**

Ficha catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).

Oliveira, Michael Tadeu Alves de.

Especificação de Requisitos de Software Utilizando Padrões no Contexto da Engenharia de Requisitos Orientada a Aspectos / Michael Tadeu Alves de Oliveira. - 2020.

174 p. : il.

Orientador(a): Heitor Augustus Xavier Costa.

Coorientador(a): Paulo Afonso Parreira Júnior.

Dissertação (mestrado acadêmico) - Universidade Federal de Lavras, 2020.

Bibliografia.

1. Padrões de Requisitos de Software. 2. Engenharia de Requisitos Orientada a Aspectos. 3. Engenharia de Software Baseada em Evidências. I. Costa, Heitor Augustus Xavier. II. Parreira Júnior, Paulo Afonso. III. Título.

MICHAEL TADEU ALVES DE OLIVEIRA

**ESPECIFICAÇÃO DE REQUISITOS DE SOFTWARE UTILIZANDO
PADRÕES NO CONTEXTO DA ENGENHARIA DE REQUISITOS
ORIENTADA A ASPECTOS**

**SOFTWARE REQUIREMENTS SPECIFICATION USING STANDARDS
IN THE ASPECT-ORIENTED REQUIREMENTS ENGINEERING
CONTEXT**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, área de concentração em Banco de Dados e Engenharia de Software, para a obtenção do título de Mestre.

APROVADA em 27 de novembro de 2020

Dr. Rafael Serapilha Durelli - UFLA

Dr. Matheus Carvalho Viana - UFSJ

Prof. Dr. Heitor Augustus Xavier Costa
Orientador

Prof. Dr. Paulo Afonso Parreira Júnior
Coorientador

**LAVRAS-MG
2020**

*À minha esposa Dayanna, ao meu filho Bernardo,
aos meus pais João e Inalva (in memoriam) e à minha sogra Maria Silvana.*

AGRADECIMENTOS

Agradeço primeiramente a Deus por minha saúde, pelas oportunidades oferecidas, pela vida, pela sabedoria, pelos sonhos realizados e pelas pessoas que sempre estiveram em meu caminho, apoiando-me ao longo da jornada. Dentre essas pessoas, gostaria de destacar as seguintes:

À minha esposa Dayanna, pelo amor e apoio incondicional em todos os momentos, demonstrado em palavras e gestos que sempre guardarei por toda a minha vida.

Ao meu filho Bernardo, que é a razão da minha vida, sendo motivo imenso de alegria e um companheiro inseparável de “trabalho”.

Aos meus pais João e Inalva, pelo apoio imenso, ensinamentos, suporte para nunca deixar de lutar pelos meus sonhos e objetivos, e que contribuíram para chegar nesse dia.

Aos meus irmãos Márcio, Léo e Lamarquiana, que sempre estão dispostos a ajudar, seja qual for o momento.

À minha sogra/ madrinha Maria Silvânia e a avó Doralice por sempre estarem dispostas a ajudar, e a confortar nos momentos mais difíceis, atuando como mães que sempre posso contar.

Ao meu grande amigo Elias que, com alegria, passou-me muita sabedoria e companheirismo.

Ao meu orientador Prof^o. Dr^o. Heitor Augustus Xavier Costa e coorientador Prof^o. Dr^o. Paulo Afonso Parreira Júnior, pelo empenho, paciência, dedicação e imenso apoio no desenvolvimento deste trabalho.

Agradeço também à Universidade Federal de Lavras pela a oportunidade. Aos membros da banca, pela disponibilidade e atenção. E a todos que diretamente ou indiretamente fizeram com que este trabalho fosse realizado.

RESUMO

Um interesse de software consiste em um conjunto de requisitos que devem colaborar para um mesmo objetivo. Quando um interesse envolve requisitos que se encontram entrelaçados com requisitos de outros interesses, tem-se um “Interesse Transversal”, e a modularização inadequada desse tipo de interesse pode dificultar o desenvolvimento e a evolução do software. A área de Engenharia de Requisitos Orientada a Aspectos (EROA) oferece estratégias mais adequadas para identificação, classificação, representação e composição de Interesses Transversais. Um problema comumente relatado nos estudos experimentais sobre abordagens para EROA é a baixa efetividade (cobertura e precisão) proporcionada por elas, relacionadas à identificação e à classificação de interesses do software. Uma das possíveis causas para isso é a falta de preocupação com a qualidade do Documento de Requisitos de Software (DRS) antes da execução das atividades de identificação e classificação de interesses, que consiste na descoberta dos interesses a partir dos requisitos funcionais e não-funcionais, classificando-os em interesses base ou ITs. Nesse sentido, o objetivo deste trabalho foi investigar se o uso de Padrões para Requisitos de Software (PRS) para escrever DRS pode contribuir para a melhoria da efetividade da identificação e classificação de interesses em sistemas de software. Um PRS é um artefato que fornece orientações sobre como especificar requisitos funcionais e não-funcionais, de forma que possam ser reusados em contextos e problemas bem definidos. Para isso, realizou-se uma pesquisa documental para explorar o estado da arte sobre PRS e abordagens para EROA. Desenvolveu-se uma abordagem para transformar requisitos em forma de casos de uso para texto narrativo, elaborou-se PRS para CRUD, Segurança, Usabilidade, Desempenho e Portabilidade, bem como um apoio computacional para apoiar a execução da atividade de identificação e classificação de interesses de software. Realizou-se estudos experimentais com DRS de sistemas de software acadêmicos e reais, comparando a efetividade do apoio computacional com o uso de PRS e sem o uso de PRS e verificando os resultados por testes estatísticos e as medidas de cobertura, precisão e *F-Measure*. Os resultados indicaram que a utilização de PRS para identificação e classificação de interesses de software contribui significativamente para o aprimoramento da cobertura, precisão e *F-Measure* do processo de identificação e classificação de interesses a partir de requisitos de software, melhorando a eficiência e eficácia na atividade para EROA.

Palavras-chave: Documentos de Requisitos de Software; Orientação a Aspectos; Engenharia de Requisitos Orientada a Aspectos; Padrões de Requisitos de Software; Identificação e classificação de Interesses.

ABSTRACT

An interest in software consists of a set of requirements that must collaborate towards the same objective. When an interest involves requirements that are intertwined with requirements of other interests, there is a Crosscutting Concern (CCC), and the inadequate modularization of this type of interest can hinder the development and evolution of the software. The Aspect-Oriented Requirements Engineering (AORE) area offers more appropriate strategies for the identification, classification, representation and composition of CCC. A problem commonly reported in experimental studies on approaches to AORE is the low effectiveness (coverage and accuracy) provided by them, related to the identification and classification of software interests. One of the possible causes for this is the lack of concern with the quality of the Software Requirements Document (SRD) before carrying out the activities of identification and classification of interests, which consists of discovering the interests from the functional and non-functional requirements, classifying them into basic concerns or CCC. In this sense, the objective of this work was to investigate whether the use of Software Requirements Pattern (SRP) to write SRD can contribute to improving the effectiveness of the identification and classification of interests in software systems. A SRP is an artifact that guides how to specify functional and non-functional requirements, so that they can be reused in well-defined contexts and problems. To this end, a documentary research was conducted to explore the state of the art on SRP and approaches to AORE. An approach was developed to transform requirements in the form of use cases for narrative text, SRP was elaborated for CRUD, Security, Usability, Performance and Portability, as well as computational support to support the performance of the activity of identification and classification of concerns software. Experimental studies were performed with SRD of academic and real software systems, comparing the effectiveness of the computational tool with the use of SRP and without the use of SRP and verifying the results by statistical tests and the measures of coverage, precision and F-Measure. The results indicated that the use of SRP for the identification and classification of software concerns contributes significantly to the improvement of the coverage, precision and F-Measure of the process of identification and classification of concerns based on software requirements, improving efficiency and effectiveness in the activity for AORE.

Keywords: Software Requirements Documents; Orientation to Aspects; Aspect-Oriented Requirements Engineering; Software Requirements Pattern; Identification and Classification of Concerns.

LISTA DE ILUSTRAÇÕES

Figura 2.1 - Método da Pesquisa.....	9
Figura 3.1 - Publicações por Ano.....	31
Figura 3.2 - Classificação dos Artigos.....	39
Figura 3.3 - Tipos de Veículos de Publicação dos Artigos Analisados.....	42
Figura 5.1 - Representação Gráfica da Abordagem.....	55
Figura 6.1 - Arquitetura de ObasCId-Tool+.....	60
Figura 6.2 - Diagrama de Casos de Uso.....	61
Figura 6.3 - Tabela do Req_Pattern.....	62
Figura 6.4 - Tela Incluir Pesquisador.....	63
Figura 6.5 - Tela Login do Usuário.....	63
Figura 6.6 - Tela de Manutenção de Catálogos de Interesses.....	63
Figura 6.7 - Tela de Inclusão de Catálogo de Interesses.....	64
Figura 6.8 - Tela de Inclusão de Interesses.....	64
Figura 6.9 - Tela de Manutenção de Palavras-Chave para Interesses.....	65
Figura 6.10 - Tela de Inclusão de Documentos de Requisitos de Software (DRS).....	65
Figura 6.11 - Tela de Inclusão de Requisitos.....	65
Figura 6.12 - Tela de Cadastro de novos PRS.....	66
Figura 6.13 - Tela de Inclusão de Requisitos com PRS.....	67
Figura 6.14 - Tela de Inclusão de Unidade de Identificação.....	67
Figura 6.15 - Tela do Resultado: DRS sem PRS.....	68
Figura 7.1 - Cobertura, Precisão F-Measure.....	75

LISTA DE TABELAS

Tabela 3.1 - Questões de Pesquisa (QP) e Justificativas.	15
Tabela 3.2 - Tipos de Estudos (adaptado de WIERINGA <i>et al.</i> , 2006).....	15
Tabela 3.3 - Estrutura para Elaboração da <i>String</i> de Busca Utilizando <i>Framework</i> PICo.	16
Tabela 3.4 - Critérios de Exclusão	17
Tabela 3.5 - <i>String</i> de Busca e Filtros Utilizados nas Bibliotecas.....	17
Tabela 3.6 - Resumo do Resultado do MSL.....	18
Tabela 3.7 - PRS Encontrados no MSL.....	20
Tabela 3.8 - Notações Utilizadas em DRS.....	21
Tabela 3.9 - Tipos de Requisitos Contemplados nos Artigos.....	23
Tabela 3.10 - Contextos.....	25
Tabela 3.11 - Ferramentas/Apoios Computacionais, Abordagens, Métodos ou Normas.....	26
Tabela 3.12 - Tipos de Estudos.....	28
Tabela 3.13 - Formas para Avaliar PRS.	29
Tabela 3.14 - Questões de Pesquisa (QP) e Justificativas.	32
Tabela 3.15 - Estrutura para Elaboração da <i>String</i> de Busca Utilizando <i>Framework</i>	33
Tabela 3.16 - Critérios de Exclusão.	34
Tabela 3.17 - <i>Strings</i> de Busca e Filtros Utilizados nos Repositórios.....	35
Tabela 3.18 - Resumo do Resultado do MSL.....	36
Tabela 3.19 - Abordagens para EROA.....	37
Tabela 3.20 - Paradigmas de Identificação e Classificação de Interesses.	40
Tabela 4.1 - <i>Template</i> de Apresentação do PRS.....	46
Tabela 4.2 - <i>Template</i> para PRS de CRUD - Incluir Informação.	47
Tabela 4.3 - <i>Template</i> para PRS de CRUD - Recuperar Informação.....	48
Tabela 4.4 - <i>Template</i> para PRS de CRUD - Alterar Informação.....	48
Tabela 4.5 - <i>Template</i> para PRS de CRUD - Excluir Informação.....	49
Tabela 4.6 - <i>Template</i> para PRS de Segurança.....	49
Tabela 4.7 - <i>Template</i> para PRS de Desempenho.....	50
Tabela 4.8 - <i>Template</i> para PRS de Usabilidade.....	52
Tabela 4.9 - <i>Template</i> para PRS de Portabilidade.....	52
Tabela 5.1 - Artefatos Consumidos e Gerados/Atualizados.	55
Tabela 5.2 - Modelo de DRS adotado na Abordagem Proposta.....	56
Tabela 5.3 - Trecho do DRS do Sistema de Software Health Watcher (HEALTH....	57
Tabela 5.4 - Trecho de DRS do Sistema de Software Health Watcher.	57
Tabela 5.5 - Requisito do Sistema de Software Health Watcher após a.....	57
Tabela 7.1 - Requisitos Funcionais do Sistema de Software Health Watcher em Texto	73
Tabela 7.2 - Requisitos Funcionais do Sistema de Software Health Watcher com	74
Tabela 7.3 - Definição do objetivo do Estudo Experimental Piloto.	76
Tabela 7.4 - Hipóteses para os Estudos Experimentais.....	76
Tabela 7.5 - Requisitos Funcionais e Requisitos Não-Funcionais do Sistema de Software	77
Tabela 7.6 - Catálogo de Interesses Desenvolvido a partir de Dados Históricos do.....	78
Tabela 7.7 - Execução do Estudo Experimental Piloto.	78
Tabela 7.8 - Resultados do Estudo Experimental Piloto para o Sistema de Software	79
Tabela 7.9 - Sistemas de Software utilizados nos Estudos Experimentais.	81

Tabela 7.10 - Requisitos Funcionais e Não-Funcionais e seus Interesses.	81
Tabela 7.11 - Execução do Estudo Experimental I.....	82
Tabela 7.12 - Resultados do Estudo Experimental I.	83
Tabela 7.13 - Requisitos Funcionais e Não-Funcionais e seus Interesses.	83
Tabela 7.14 - Execução do Estudo Experimental II.	84
Tabela 7.15 - Resultados do Estudo Experimental II.....	84
Tabela 7.16 - Requisitos Funcionais e Não-Funcionais e seus Interesses.	85
Tabela 7.17 - Execução do Estudo Experimental III.....	85
Tabela 7.18 - Resultados do Estudo Experimental III.	86
Tabela 7.19 - Requisitos Funcionais e Não-Funcionais e seus Interesses.	87
Tabela 7.20 - Execução do Estudo Experimental IV.....	87
Tabela 7.21 - Resultados do Estudo Experimental IV.	88
Tabela 7.22 - Requisitos Funcionais e Não-Funcionais e seus Interesses.	88
Tabela 7.23 - Execução do Estudo Experimental V.	90
Tabela 7.24 - Resultados do Estudo Experimental V.....	90
Tabela 7.25 - Interesses Analisados sem PRS.	92
Tabela 7.26 - Interesses Analisados com PRS.....	92
Tabela 7.27 - Análise das Medidas por Sistemas de Software Sem e Com Uso de PRS. .	93
Tabela 7.28 - Teste de Hipótese por Interesse de Software.	94
Tabela 7.29 - Teste de Hipótese por Sistema de Software.	96

LISTA DE SIGLAS

- CRUD** *Create, Read, Update e Delete*
- DRS** Documento de Requisitos de Software
- ER** Engenharia de Requisitos
- EROA** Engenharia de Requisitos Orientada a Aspectos
- GATE** *General Architecture for Text Engineering*
- IDE** *Integrated Development Environment*
- MSL** Mapeamento Sistemático da Literatura
- PABRE** *Pattern-Based Requirements Elicitation*
- PUMConf** *Product Line Use Case Model Configurator*
- PQ** Publication Questions
- PRS** Padrões de Requisitos de Software
- RF** Requisitos Funcionais
- RNF** Requisitos Não-Funcionais
- RQ** *Research Questions*
- SI** Sistemas de Informação

SUMÁRIO

1	INTRODUÇÃO	1
1.1	Motivação	3
1.2	Objetivo	3
1.3	Organização do Trabalho	4
2	MÉTODO DE PESQUISA	7
2.1	Considerações Iniciais	7
2.2	Classificação da Pesquisa	7
2.3	Método de Pesquisa	8
2.4	Considerações Finais	10
3	ESTADO DA ARTE.....	12
3.1	Considerações Iniciais	12
3.2	Método de Pesquisa	12
3.3	Padrões de Requisitos de Software	14
3.3.1	Planejamento.....	14
3.3.2	Execução do MSL	16
3.3.3	Análise dos Resultados	19
3.4	Engenharia de Requisitos Orientada a Aspectos.....	31
3.4.1	Planejamento.....	32
3.4.2	Execução do MSL	34
3.4.3	Análise dos Resultados	36
3.5	Considerações Finais	42
4	PADRÕES DE REQUISITOS DE SOFTWARE.....	45
4.1	Considerações Iniciais	45
4.2	Estrutura do PRS	45
4.3	Seleção dos PRS	46
4.3.1	<i>Template</i> para PRS de CRUD.....	46
4.3.2	<i>Template</i> para PRS de Segurança.....	49
4.3.3	<i>Template</i> para PRS de Desempenho	50
4.3.4	<i>Template</i> para PRS de Usabilidade.....	51
4.3.5	<i>Template</i> para PRS de Portabilidade.....	51
4.4	Considerações Finais	53
5	UMA ABORDAGEM PARA A PADRONIZAÇÃO DE DOCUMENTOS DE REQUISITOS DE SOFTWARE	54
5.1	Considerações Iniciais	54
5.2	Visão Geral da Abordagem	54
5.3	Descrição das Atividades da Abordagem Proposta.....	55
5.3.1	Fase: Seleção do DRS	56
5.3.2	Fase: Transformação do DRS em Texto Narrativo.....	56
5.3.3	Fase: Padronização do DRS	57

5.4	Considerações Finais	58
6	OBASCID-TOOL+: APOIO COMPUTACIONAL À EROA UTILIZANDO PADRÕES DE REQUISITOS DE SOFTWARE.....	59
6.1	Considerações Iniciais	59
6.2	Visão Geral do ObasCId-Tool+	59
6.3	Módulo Gerenciamento de Documentos de Requisitos.....	66
6.4	Módulo Classificação e Identificação de Interesses.....	66
6.5	Considerações Finais	68
7	ESTUDOS EXPERIMENTAIS	69
7.1	Considerações Iniciais	69
7.2	Sistema de Software Health Watcher.....	70
7.2.1	Requisitos Funcionais e Não-Funcionais do Sistema de Software Health Watcher	70
7.3	Medidas Cobertura, Precisão e <i>F-Measure</i>	73
7.4	Estudo Experimental Piloto.....	75
7.4.1	Planejamento do Estudo Experimental Piloto	75
7.4.2	Análise dos Resultados do Estudo Experimental Piloto	78
7.5	Estudos Experimentais.....	80
7.6	Estudo Experimental I - EASYHALF	80
7.6.1	Requisitos Funcionais e Não-Funcionais	80
7.6.2	Planejamento do Estudo Experimental I	81
7.6.3	Análise dos Resultados do Estudo Experimental I.....	82
7.7	Estudo Experimental II - FAKE NEWS APP	82
7.7.1	Requisitos Funcionais e Não-Funcionais	83
7.7.2	Planejamento do Estudo Experimental II.....	83
7.7.3	Análise dos Resultados do Estudo Experimental II	84
7.8	Estudo Experimental III - OBASCID-TOOL	85
7.8.1	Requisitos Funcionais e Não-Funcionais	85
7.8.2	Planejamento do Estudo Experimental III	85
7.8.3	Análise dos Resultados do Estudo Experimental III.....	86
7.9	Estudo Experimental IV - Sistema Gerenciador de Atividades Curriculares (SisGAC).....	86
7.9.1	Requisitos Funcionais e Não-Funcionais	87
7.9.2	Planejamento do Estudo Experimental IV.....	87
7.9.3	Análise dos Resultados do Estudo Experimental IV	87
7.10	Estudo Experimental V - VEREDAS SOL E LARES	88
7.10.1	Requisitos Funcionais e Não-Funcionais	88
7.10.2	Planejamento do Estudo Experimental V	89
7.10.3	Análise dos Resultados do Estudo Experimental V.....	90
7.11	Interesses Analisados nos Estudos Experimentais	91
7.12	Testes de Hipóteses	93
7.13	Ameaças à Validade dos Estudos Experimentais	97
7.14	Considerações Finais	98
8	TRABALHOS RELACIONADOS	99

9	AMEAÇAS À VALIDADE	102
10	CONSIDERAÇÕES FINAIS	104
10.1	Contribuições e Limitações.....	106
10.2	Trabalhos Futuros	107
	REFERÊNCIAS	109
	APÊNDICE A - ARTIGOS RESULTANTES DO MSL SOBRE PRS.....	117
	APÊNDICE B - ARTIGOS RESULTANTES DO MSL SOBRE EROA..	118
	APÊNDICE C - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE HEALTH WATCHER.....	121
	APÊNDICE D - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE HEALTH WATCHER EM TEXTO NARRATIVO	127
	APÊNDICE E - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE HEALTH WATCHER UTILIZANDO PRS	130
	APÊNDICE F - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE EASYHALF.....	133
	APÊNDICE G - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE EASYHALF UTILIZANDO PRS	134
	APÊNDICE H - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE FAKE NEWS APP.....	136
	APÊNDICE I - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE FAKE NEWS APP EM TEXTO NARRATIVO	140
	APÊNDICE J - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE FAKE NEWS APP UTILIZANDO PRS	142
	APÊNDICE K - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE OBASCID-TOOL	144

APÊNDICE L - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE OBASCID-TOOL UTILIZANDO PRS..... 147

APÊNDICE M - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE SISGAC..... 150

APÊNDICE N - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE SISGAC UTILIZANDO PRS 151

APÊNDICE O - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE VEREDAS SOL E LARES 153

APÊNDICE P - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE VEREDAS SOL E LARES UTILIZANDO PRS 157

1 INTRODUÇÃO

Os requisitos de um sistema de software correspondem às descrições do que esse sistema deve fazer, aos serviços que ele oferece e às restrições em sua funcionalidade. Tais requisitos representam a demanda dos clientes desse sistema (SOMMERVILLE, 2019; CHITCHYAN *et al.*, 2005). Em geral, os requisitos são classificados como (SOMMERVILLE, 2019): i) **Requisitos Funcionais (RF)**, os quais descrevem funções que o sistema deve fornecer, como ele deve reagir às entradas específicas e como ele deve se comportar em determinadas situações; e ii) **Requisitos Não-Funcionais (RNF)**, os quais representam restrições impostas às funções oferecidas pelo sistema ou ao processo de desenvolvimento.

Um conjunto de requisitos de software relacionados a uma mesma função denomina-se **Interesse de Software** (HERRERA *et al.*, 2012; SAMPAIO *et al.*, 2007). Por exemplo, o interesse *Segurança* contempla requisitos relacionados ao propósito de “garantir que o sistema de software seja seguro” (PARREIRA JÚNIOR, 2015). Quando um interesse possui requisitos entrelaçados (relacionados) com requisitos de outros interesses, tem-se um **Interesse Transversal** (no restante do trabalho, será apenas **Interesse**).

Para exemplificar esse conceito, são apresentados os seguintes requisitos hipotéticos, extraídos do sistema de software *Health Watcher* (HEALTH WATCHER, 2020): i) REQ1 - Funcionários podem registrar queixas de saúde no sistema de software, a queixa deve ser cadastrada e com a situação ABERTA; ii) REQ2 - O funcionário deve estar autenticado no sistema de software para registrar uma queixa. Percebe-se que o propósito do requisito REQ1 é explicitar um comportamento funcional do sistema de software, relacionado ao interesse “Gerenciamento de Queixas”. O propósito do requisito REQ2, de acordo com sua descrição, é especificar um comportamento não-funcional (uma restrição) referente ao interesse *Segurança*. Nota-se que o requisito REQ1 depende da condição “o funcionário deve estar autenticado no sistema” do requisito REQ2. Nesse caso, o interesse *Segurança* é transversal, pois há um requisito desse interesse que afeta (está entrelaçado a) um requisito de outro interesse do sistema de software.

Ao longo do desenvolvimento de sistemas de software, é desejável a separação dos interesses presentes no sistema, de forma que cada interesse possa ser encapsulado em um módulo específico, que satisfaça somente aos requisitos relativos a ele (SAMPALIO *et al.*, 2007). Quando essa separação não é realizada de forma adequada, o desenvolvimento e a evolução do sistema de software podem ser comprometidos. Contudo, realizar essa separação nem sempre

é fácil, principalmente, quando se trata de interesses transversais (HERRERA *et al.*, 2012; SAMPAIO *et al.*, 2007).

Uma etapa fundamental para realizar a separação de interesses de um sistema de software consiste em identificar os interesses existentes e classificá-los como **base**¹ (que não afetam requisitos de outros interesses) ou **transversais**. Nesse contexto, na Engenharia de Requisitos Orientada a Aspectos (EROA), são estudadas abordagens (métodos, técnicas e ferramentas) mais adequadas para a separação de interesses (SAMPAIO *et al.*, 2007). Todavia, estudos experimentais recentes têm relatado baixa efetividade (cobertura e precisão) das abordagens propostas na literatura (PARREIRA JÚNIOR; PENTEADO, 2018; HERRERA *et al.*, 2012; SAMPAIO *et al.*, 2007).

Uma das possíveis causas desse problema é a falta de preocupação com a qualidade do Documento de Requisitos de Software (DRS), antes da execução das atividades de identificação e classificação dos interesses (PARREIRA JÚNIOR; PENTEADO, 2014). Para tentar mitigar esse problema, os Padrões de Requisitos de Software (PRS) são um tipo de artefato de software que pode ser útil, pois fornecem orientações sobre como especificar requisitos funcionais e requisitos não-funcionais, de forma a aprimorar a qualidade desses requisitos (PALOMARES *et al.*, 2011). Por exemplo, o requisito “REQ1 - Funcionários podem registrar queixas de saúde no sistema de software” (apresentado anteriormente) poderia ser descrito utilizando um PRS para inclusão de informações (BARCELOS; PENTEADO, 2017), tal como “Permitir a inclusão de informações de **<entidade>**, contendo os seguintes atributos: **<atributos>**, **<condição>**”. Neste caso, o requisito REQ1 seria reescrito como “Permitir a inclusão de informações de Queixas, contendo os seguintes atributos: tipo da queixa, descrição da queixa, observações e localidade, a queixa deve ser cadastrada e com a situação ABERTA”.

Nota-se que, ao utilizar um PRS, uma parte da descrição do requisito é fixa, permitindo alterações apenas nas *tags* (<...>) pré-definidas. Assim, os PRS podem ser importante ferramenta de apoio à melhoria de qualidade de um DRS, por exemplo, evitando ambiguidades, contradições e incoerências e favorecendo a identificação e classificação mais precisas de interesses de software.

¹ Interesses base (ou não transversais), no contexto da EROA, são aqueles cujos requisitos não estão entrelaçados com requisitos de outros interesses do software (HERRERA *et al.*, 2012; SAMPAIO *et al.*, 2007).

1.1 Motivação

A EROA parte da premissa de que, no ciclo de desenvolvimento, o quanto antes os interesses de um sistema de software forem encontrados, menor será o esforço para sua separação e modularização (PARREIRA JÚNIOR, 2015). Ainda que alguns trabalhos apresentem avanços quanto à efetividade (cobertura e precisão) das abordagens para EROA (PARREIRA JÚNIOR; PENTEADO 2018; HERRERA *et al.*, 2012; SAMPAIO *et al.*, 2007), muitas dessas abordagens ainda consideram que o DRS, utilizado como entrada para o processo de identificação e classificação de interesses, encontra-se bem escrito, sendo uma consideração pouco realística (RAMOS; DE CASTRO, 2005; REIS *et al.*, 2002). Mesmo apresentando qualidade aceitável, é possível que a forma de como DRS foi escrito não favoreça o processo de identificação e classificação de interesses, uma vez que Engenheiros de Software possuem estilo de especificação de requisitos baseado em sua experiência.

1.2 Objetivo

A hipótese deste trabalho considera que **a utilização de PRS para escrever DRS pode contribuir para a melhoria da efetividade da identificação e classificação de interesses em sistemas de software**. Neste trabalho, a efetividade pode ser entendida como maior quantidade de interesses corretamente identificados (cobertura), com menor índice de falsos positivos (precisão).

O conceito de PRS tem sido utilizado em várias áreas (por exemplo, contextos gerais de sistemas de software (WITHALL 2007; ROBERTSON, 1996), contextos particulares, como o de sistemas embarcados (KONRAD; CHENG, 2002) ou sistemas de informação (TOVAL *et al.*, 2002)), porém não há relatos da sua utilização no contexto da EROA, conforme observado por meio de um Mapeamento Sistemático da Literatura (MSL) (detalhado no Capítulo 3).

Verificar a hipótese anterior é o objetivo desse trabalho. Assim, foi elaborada uma abordagem para identificação e classificação de interesses em DRS escritos utilizando PRS. Para avaliar essa abordagem, foi evoluído o apoio computacional para EROA denominado `ObasCId-Tool`² (*Ontologically-based Concern Identification and Classification Tool*) (PARREIRA JÚNIOR; PENTEADO, 2018). `ObasCId-Tool` foi escolhido, pois: i) é baseado em uma das abordagens mais recentes sobre EROA (conforme MSL apresentado no Capítulo 3), tendo apresentado resultados experimentais que aprimoram os existentes no estado-da-arte; ii) é livre, permitindo a concentração de esforços na sua evolução para contemplar a utilização

² Disponível em: <https://gitlab.com/advanse/obascid-tool>

de PRS; e iii) o aluno de mestrado e os seus orientadores possuem conhecimento prévio. Para alcançar o objetivo proposto, os seguintes procedimentos metodológicos foram realizados:

- a) **Extensão do MSL sobre EROA (PARREIRA JÚNIOR; PENTEADO, 2014).** O MSL anterior contemplava abordagens para EROA publicadas até o ano de 2012. Assim, o MSL foi atualizado para conhecer e analisar novas propostas de abordagens para identificação e classificação de interesses;
- b) **Realização de um MSL sobre PRS.** Um novo MSL foi realizado para conhecer o estado da arte sobre PRS e propor uma abordagem para padronizar DRS e aprimorar as atividades para EROA;
- c) **Elaboração de uma abordagem para transformação dos requisitos.** Foi elaborada uma abordagem para transformar requisitos de software escritos em casos de uso em texto narrativo³. Isso foi necessário, pois *ObasCId-Tool* utiliza requisitos descritos em texto narrativo;
- d) **Elaboração de uma abordagem de padronização utilizando PRS.** Foi elaborada uma abordagem de escrita de DRS utilizando PRS;
- e) **Desenvolvimento do apoio computacional *ObasCId-Tool+ 4* (evolução de *ObasCId-Tool*).** *ObasCId-Tool* foi evoluído para armazenar e manter requisitos utilizando PRS;
- f) **Avaliação da abordagem.** Foram realizados experimentos controlados para avaliar a hipótese deste trabalho.

1.3 Organização do Trabalho

No Capítulo 2, é apresentado o método de pesquisa utilizado para a realização deste trabalho, que possui etapas e atividades realizadas com os resultados obtidos.

No Capítulo 3, são apresentados os MSL conduzidos, visando compreender o estado da arte sobre PRS e EROA.

No Capítulo 4, são apresentados os PRS utilizados nos estudos experimentais deste trabalho. Alguns foram selecionados a partir do MSL sobre PRS realizado, identificados no Capítulo 3, e outros que serviram de base para a criação de novos PRS.

³ Neste trabalho, considera-se texto narrativo aquele que narra uma história por meio de uma sequência de fatos (Fonte: <https://www.educamaisbrasil.com.br/enem/lingua-portuguesa/texto-narrativo>)

⁴ Disponível em: <https://github.com/PqES/obascid-tool-plus>

No Capítulo 5, é apresentada a abordagem utilizada para a padronização de DRS. Essa abordagem é utilizada para aprimorar a efetividade das atividades de identificação e classificação de interesses de software.

No Capítulo 6, é apresentado o apoio computacional *ObasCId-Tool+* para a realização do cadastro dos requisitos de software, utilizando PRS, bem como a identificação e a classificação de interesses de software. Além disso, *ObasCId-Tool+* possui o módulo de cadastro dos PRS. Por fim, é apresentado um experimento piloto para avaliar *ObasCId-Tool+*, utilizando PRS para aprimorar as atividades para EROA.

No Capítulo 7, são apresentados os estudos experimentais conduzidos sobre a abordagem de padronização dos requisitos e sobre *ObasCId-Tool+*, bem como o planejamento desses estudos.

No Capítulo 8, são apresentados e discutidos alguns trabalhos relacionados com o escopo deste trabalho.

No Capítulo 9, são apresentadas ameaças à validade deste trabalho, sendo classificadas em Ameaça Externa, Ameaça Interna, Ameaça de Conclusão e Ameaça de Construção.

No Capítulo 10, são apresentadas as conclusões deste trabalho, assim como são discutidas as principais contribuições deste trabalho, destacando as propostas para sua continuidade como sugestões de trabalhos futuros.

No APÊNDICE A e no APÊNDICE B, é apresentada a referência completa dos estudos selecionados nos MSL sobre PRS e EROA, respectivamente.

No APÊNDICE C, no APÊNDICE D e no APÊNDICE E, são apresentados os requisitos funcionais e não-funcionais do sistema de software *Health Watcher* segmentados da seguinte forma: i) no primeiro apêndice, são apresentados os requisitos em formato descritivo de casos de uso; ii) no segundo apêndice, são apresentados os requisitos em formato descritivo de texto narrativo, pois foi aplicado a Transformação do DRS em Texto Narrativo da abordagem apresentada no Capítulo 5; e iii) no terceiro apêndice, são apresentados os requisitos com a utilização dos PRS exibidos no Capítulo 4 deste trabalho. Com isso, no segundo e no terceiro apêndices, os requisitos serviram de *inputs* para o estudo experimental piloto.

No APÊNDICE F e no APÊNDICE G, são apresentados os requisitos funcionais e não-funcionais do sistema de software *EasyHalf*, respectivamente. Assim, no primeiro apêndice, são apresentados os requisitos em formato de texto narrativo e, no segundo apêndice, são apresentados os requisitos com a utilização dos PRS. Ambos serviram de *input* para a execução do **Estudo Experimental I**.

No APÊNDICE H, no APÊNDICE I e no APÊNDICE J, são apresentados os requisitos funcionais e não-funcionais do sistema de software *Fake News App* organizados da seguinte forma: i) no primeiro apêndice, são apresentados os requisitos em formato descritivo de casos de uso; ii) no segundo apêndice, são apresentados os requisitos em formato descritivo de texto narrativo, pois foi aplicado a Transformação do DRS em Texto Narrativo; e iii) no terceiro apêndice, são apresentados os requisitos com a utilização dos PRS. Assim, no segundo e no terceiro apêndices os requisitos serviram de *input* para o **Estudo Experimental II**.

No APÊNDICE K e no APÊNDICE L, são apresentados os requisitos funcionais e não-funcionais do sistema de software *ObasCID-Tool*, respectivamente. Assim, no primeiro apêndice, são apresentados os requisitos em formato de texto narrativo e, no segundo apêndice, são apresentados os requisitos com a utilização dos PRS. Ambos serviram de *input* para a execução do **Estudo Experimental III**.

No APÊNDICE M e no APÊNDICE N, são apresentados os requisitos funcionais e não-funcionais do sistema de software *SisGAC*, respectivamente. Assim, no primeiro apêndice, são apresentados os requisitos em formato de texto narrativo e, no segundo apêndice, são apresentados os requisitos com a utilização dos PRS. Ambos serviram de *input* para a execução do **Estudo Experimental IV**.

No APÊNDICE O e no APÊNDICE P, são apresentados os requisitos funcionais e não-funcionais do sistema de software *Veredas Sol e Lares*, respectivamente. Assim, no primeiro apêndice, são apresentados os requisitos em formato de texto narrativo e, no segundo apêndice, são apresentados os requisitos com a utilização dos PRS. Ambos serviram de *input* para a execução do **Estudo Experimental V**.

2 MÉTODO DE PESQUISA

2.1 Considerações Iniciais

A Metodologia de Pesquisa é o conjunto de métodos, de técnicas e de procedimentos a serem empregados durante a execução do trabalho de pesquisa científica, a fim de obter como resultado um novo produto, processo ou conhecimento (PRODANOV; DE FREITAS, 2013).

Este capítulo está organizado da seguinte forma. Na Seção 2.2, é apresentada a caracterização deste trabalho mediante a classificação dos tipos de pesquisa existentes na literatura. Na Seção 2.3, são apresentadas as etapas para o desenvolvimento deste trabalho (método de pesquisa).

2.2 Classificação da Pesquisa

Uma pesquisa científica pode ser classificada sob vários aspectos (PRODANOV; DE FREITAS, 2013). A seguir, é descrito como este trabalho pode ser classificado:

- a) **Natureza.** A pesquisa pode ser classificada em (i) básica, cujo objetivo é gerar conhecimento para o avanço da ciência, sem aplicação prática prevista, ou (ii) aplicada, cujo objetivo é gerar conhecimentos para aplicação prática, dirigidos à solução de problemas específicos. Este trabalho pode ser classificado como **pesquisa aplicada**, dirigida à geração de conhecimentos, contribuindo com a identificação e classificação de interesses;
- b) **Objetivos.** A pesquisa pode ser classificada como (i) exploratória, quando a pesquisa está na fase preliminar e o objetivo é proporcionar mais informações sobre o assunto, (ii) descritiva, na qual o pesquisador apenas registra e descreve os fatos observados sem interferir, ou (iii) explicativa, em que o pesquisador procura por fatores que contribuem ou determinem a ocorrência de um evento. Este trabalho pode ser classificado como **pesquisa explicativa**, uma vez que PRS são utilizados para verificar a precisão e a cobertura da identificação de interesses;
- c) **Procedimentos Técnicos.** A pesquisa pode ser classificada em (i) bibliográfica, elaborada a partir de materiais publicados sob determinados assuntos, (ii) documental, elaborada a partir de estudos que ainda não receberam tratamento analítico ou podem ser reelaborados de acordo com os objetivos da pesquisa, (iii) experimental, na qual são selecionadas variáveis que podem influenciar determinado objeto de estudo, de modo que elas possam ser observadas, (iv) *survey*, desenvolvida por meio da interrogação direta de um grupo de

peças cujo comportamento tem interesse, (v) de campo, consiste na observação de fatos e de fenômenos, tal como ocorrem espontaneamente, (vi) estudo de caso, envolve o estudo profundo e exaustivo de um ou mais objetivos para permitir mais conhecimento, (vii) *ex-post-facto*, quando a pesquisa ocorre após algum fato, (viii) pesquisa-ação, realizada a fim de fornecer uma solução para determinado problema ou (ix) pesquisa participante, envolve a interação entre pesquisadores e membros das situações investigadas. Este trabalho pode ser classificado como **pesquisa documental**, em que o conhecimento necessário para alcançar o objetivo foi obtido por meio da condução de MSL;

- d) **Forma e Abordagem do Problema.** A pesquisa pode ser classificada em (i) quantitativa, em que as opiniões e as informações são traduzidas em números de modo a classificá-las e analisá-las; ou (b) qualitativa, na qual há relação dinâmica entre o mundo real e o sujeito, não sendo quantificada em números. Este trabalho pode ser classificado como **pesquisa qualitativa e quantitativa**, pois foram realizados MSL para verificar o estado da arte quanto aos temas abordados neste trabalho e foram utilizados recursos e técnicas estatísticas para analisar os dados obtidos nos estudos experimentais realizados.

2.3 Método de Pesquisa

Neste trabalho, o objetivo foi verificar se há melhorias quanto à efetividade (cobertura, precisão e *F-Measure*) do processo de identificação e classificação de interesses, quando PRS são utilizados para escrever DRS. Para tanto, foram seguidas as seguintes etapas (Figura 2.1):

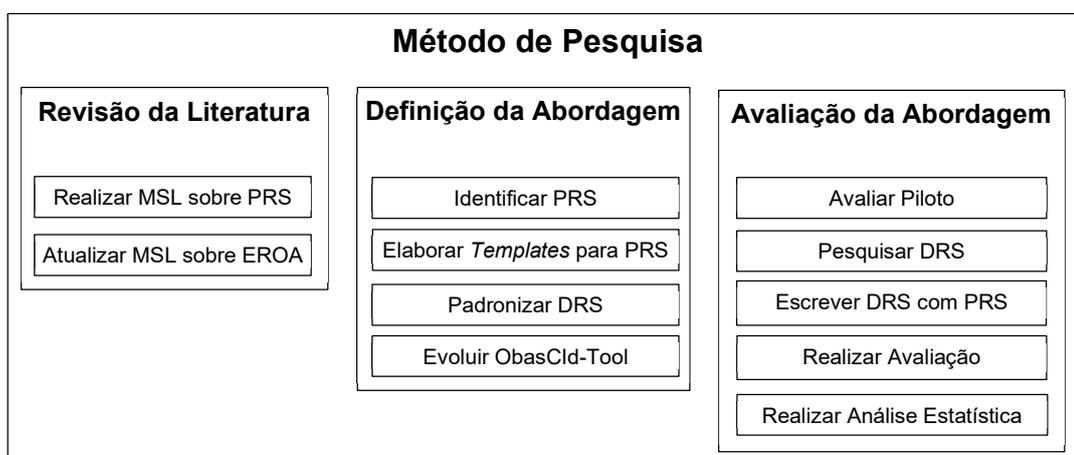
- a) **Etapa 1 - Revisão da Literatura.** Um MSL foi conduzido sobre a utilização de PRS para escrever DRS. Esses PRS foram utilizados para padronizar os requisitos do DRS. Outro MSL foi conduzido sobre EROA para atualizar o MSL realizado anteriormente (PARREIRA JÚNIOR; PENTEADO, 2014). Com o resultado desse MSL, verificou-se a existência de trabalhos recentes sobre o tema, visando à melhoria nas atividades para EROA. Ambos MSL contemplam publicações até outubro de 2020. Essa etapa é composta por duas atividades:

- **Atividade 1 - Realizar MSL sobre PRS.** Nesse MSL, foram compiladas informações sobre (i) a existência de diferentes PRS para diversos cenários de utilização, (ii) a constatação da notação mais utilizada em PRS, (iii) a verificação de como os requisitos funcionais e não funcionais são tratados em PRS, (iv) a identificação do contexto em que PRS estão sendo inseridos, (v) a existência de ferramentas/apoios computacionais para utilização de PRS, (vi) a identificação de

tipos de estudo encontrados para avaliar a utilização de PRS e (vii) a existência de medidas para avaliar a qualidade do DRS elaborado;

- **Atividade 2 - Atualizar MSL sobre EROA.** Nessa atividade, buscou-se conhecer o estado da arte sobre EROA, considerando (i) a existência de abordagens para a EROA para determinadas atividades, (ii) a identificação de tipos de estudo encontrados para avaliar abordagens para EROA e (iii) a compilação e sintetização de paradigmas utilizados para identificar interesses;

Figura 2.1 - Método da Pesquisa.



Fonte: Do autor (2020).

- b) **Etapa 2 - Definição da Abordagem.** A abordagem para padronização de DRS foi desenvolvida para aprimorar a identificação e classificação de interesses, utilizando um apoio computacional. Para isso, os PRS mais presentes na literatura foram utilizados no desenvolvimento do apoio computacional ObasCId-Tool+ para identificar e classificar os interesses. Essa etapa é composta por três atividades:

- **Atividade 1 - Identificar PRS.** A partir dos resultados do MSL sobre PRS, foram identificados PRS mais utilizados na literatura para diversos contextos;
- **Atividade 2 - Elaborar *Templates* para PRS.** Foram elaborados *templates* considerando os PRS mais utilizados na literatura para padronizar requisitos, possibilitando aumento da qualidade dos DRS;
- **Atividade 3 - Padronizar DRS.** Foi elaborada uma abordagem para padronizar DRS, visando transformá-lo em texto narrativo e, posteriormente, utilizar PRS;
- **Atividade 4 - Evoluir ObasCId-Tool.** O apoio computacional ObasCId-Tool foi evoluído, incorporando PRS para escrever DRS. Essa evolução originou o apoio computacional ObasCId-Tool+;

c) **Etapa 3 - Avaliação da Abordagem.** Foram realizados experimentos para avaliar a qualidade dos PRS, elaborados na etapa anterior e utilizados para escrever DRS. Em seguida, o apoio computacional *ObasCId-Tool+* foi avaliado utilizando DRS com e sem a utilização de PRS, analisando a cobertura e a precisão na identificação e classificação de interesses. Foram identificadas outras ferramentas/apoios computacionais que realizam identificação e classificação de interesses para realizar avaliação externa (avaliação entre duas ou mais ferramentas computacionais). Além disso, DRS com e sem a utilização de PRS foram avaliados utilizando *ObasCId-Tool+*, medindo cobertura, precisão e *F-Measure* na identificação e classificação de interesses, ou seja, foi realizada avaliação interna (avaliação da abordagem proposta). Essa etapa é composta por cinco atividades:

- **Atividade 1 - Avaliar Piloto.** Verificar qual a contribuição dos PRS elaborados/definidos anteriormente na melhoria da cobertura e da precisão na identificação e classificação dos interesses na EROA;
- **Atividade 2 - Pesquisar DRS.** Utilizar DRS existentes na literatura para serem utilizados na abordagem de padronização e que servirão como entrada para ferramentas/apoios computacionais;
- **Atividade 3 - Escrever DRS com PRS.** Utilizar o DRS encontrados/pesquisados e transformá-los utilizando os *templates* elaborados/definidos para os PRS;
- **Atividade 4 - Realizar Avaliação.** Utilizar algumas ferramentas/apoios computacionais encontradas no MSL sobre PRS para realizar estudos de avaliação com *ObasCId-Tool+*;
- **Atividade 5 - Realizar Análise Estatística.** Realizar análises com ferramentas/apoios computacionais e o DRS (com e sem utilização dos *templates* desenvolvidos para os PRS), verificando a cobertura e a precisão na identificação e classificação de interesses;

2.4 Considerações Finais

Este trabalho pode ser classificado como uma pesquisa (i) **aplicada**, pois é dirigida à geração de conhecimentos, contribuindo com a identificação e classificação de interesses, (ii) **explicativa**, pois PRS são utilizados para verificar o aumento na precisão e na cobertura da identificação de interesses de um sistema de software, (iii) **documental**, pois conhecimento necessário para alcançar o objetivo foi obtido por meio da condução de MSL, e (iv) **quantitativa e qualitativa**, pois foram utilizados recursos e técnicas estatísticas para analisar

o problema e foram realizados MSL para verificar o estado da arte quanto aos temas abordados neste trabalho.

O trabalho seguiu o método de pesquisa proposto, composto por três etapas (**Revisão da Literatura, Definição da Abordagem e Avaliação da Abordagem**). Cada uma dessas etapas é composta por atividades (2 atividades, 4 atividades e 5 atividades, respectivamente), totalizando 11 atividades necessárias para alcançar o objetivo desta pesquisa.

3 ESTADO DA ARTE

3.1 Considerações Iniciais

Os principais temas relacionados a esta pesquisa são PRS e EROA. A revisão da literatura realizada foi conduzida por meio de dois MSL (PETERSEN *et al.*, 2008). O primeiro MSL sobre PRS permitiu a identificação e a catalogação dos principais PRS utilizados e disponíveis na literatura para diferentes domínios de sistemas de software. O objetivo da realização desse MSL foi prover subsídios teóricos para a definição do problema de pesquisa desta dissertação, bem como propor uma solução. O segundo MSL foi sobre EROA permitiu estender a identificação e a catalogação das principais abordagens para EROA disponíveis na literatura.

Este capítulo está organizado da seguinte forma. Na Seção 3.2, os principais conceitos relacionados aos MSL são apresentados. Na Seção 3.3, os principais conceitos relacionados a PRS são expostos e alguns padrões que contribuíram para a construção da solução proposta na pesquisa são descritos. Na Seção 3.4, a descrição de algumas abordagens para EROA e a abordagem utilizada nos estudos experimentais nesse estudo, identificada como *ObasCId*, são mostradas.

3.2 Método de Pesquisa

Um MSL é uma técnica empregada para identificar, interpretar e avaliar pesquisas de acordo com uma ou mais questões de pesquisa, temáticas ou fenômenos de interesse (KITCHENHAM, 2004). Em um MSL, não é preciso realizar avaliação dos artigos quanto à qualidade, pois o seu resultado é um inventário de artigos sobre a área temática, mapeados para uma classificação (WIERINGA *et al.*, 2006), sendo uma visão geral do escopo da área, que permite descobrir tendências de pesquisa (PETERSEN *et al.*, 2008). Nos MSL realizados neste trabalho, três fases foram adotadas (KEELE, 2007; KITCHENHAM, 2004):

- a) **Planejamento.** É elaborado o protocolo do MSL, contendo a descrição formal da metodologia utilizada. Em geral, nesse protocolo as questões que motivam a execução da pesquisa, a *string* de busca (Tabela 3.3 e Tabela 3.15), as estratégias para coleta das informações e os critérios de exclusão (Tabela 3.4 e Tabela 3.16) dos estudos identificados nos MSL são definidos (KITCHENHAM, 2004);
- b) **Execução.** Envolve a identificação, a seleção e a avaliação dos estudos encontrados, após a aplicação do planejamento inicial nos repositórios de artigos científicos. Para isso, a *string*

de busca é utilizada nesses repositórios, selecionando e classificando os estudos segundo os critérios de exclusão definidos para os MSL;

- c) **Análise dos Resultados.** É realizada uma síntese dos resultados obtidos na fase anterior (DE ALMEIDA BIOLCHINI *et al.*, 2007). Os estudos selecionados são analisados para responder às questões de pesquisa estabelecidas durante o planejamento.

Quanto aos MSL realizados neste trabalho, foram definidos os objetivos dos MSL e as questões de pesquisa a serem abordadas de forma mais detalhada posteriormente. Além disso, foi elaborada uma *string* de busca para a recuperação de documentos, utilizando o *framework* PICO⁵, que possui três elementos (MURDOCH UNIVERSITY, 2018; JOANNA BRIGGS INSTITUTE, 2014):

- a) **Population - População (P)**, trata-se do objeto de estudo em questão;
- b) **Interest - Interesse (I)**, refere-se a um evento, uma atividade, uma experiência ou um processo definido;
- c) **Context - Contexto (Co)**, consiste no cenário ou nas características distintas relacionados ao objeto de estudo.

Para cada elemento do *framework* PICO, quando aplicável, são elencadas palavras-chave e seus respectivos sinônimos, combinando-as por meio de operadores lógicos OR e AND. Nas Seções 3.3.1 e 3.4.1, há exemplos de utilização desse *framework*. Cabe ressaltar que três pesquisadores (Pesquisador A, Pesquisador B e Pesquisador C), foram envolvidos nos MSL, realizando os seguintes procedimentos:

- a) O Pesquisador A utilizou a *string* de busca na máquina (de busca) dos repositórios de artigos científicos selecionados e documentou os resultados no software JabRef⁶;
- b) O Pesquisador A verificou e excluiu os estudos não artigos e os artigos duplicados (com título, autores e resumo iguais). Na identificação de artigos duplicados, foram mantidos os artigos com maior quantidade de palavras-chave, pois descreviam melhor o artigo (conforme definição dos pesquisadores envolvidos);
- c) Os artigos encontrados foram avaliados pelo Pesquisador B, de maneira individual, quanto à aplicação dos critérios de exclusão estabelecidos. Essa avaliação foi realizada por meio da leitura do título, resumo e palavras-chave. Os artigos, cujas avaliações causaram dúvidas

⁵ Foi desenvolvido para identificar palavras-chave e formular *strings* de busca a partir das questões de pesquisa (MURDOCH UNIVERSITY, 2018; JOANNA BRIGGS INSTITUTE, 2014).

⁶ <http://www.jabref.org/>

- quanto à sua inclusão/exclusão, foram incluídos. Os artigos foram documentados em uma lista de artigos incluídos e excluídos com justificativa para sua inclusão/exclusão;
- d) Realizou-se a interseção entre os artigos selecionados pelo Pesquisador A e pelo Pesquisador B, sendo esses artigos documentados. Na ocorrência de algum desacordo sobre a inclusão/exclusão de artigos, os dois pesquisadores discutiram e resolveram qual seria a melhor opção para retirada do artigo. Nos casos em que não houve consenso, o artigo foi incluído. Os artigos excluídos foram documentados em uma lista de artigos excluídos, com justificativa para a sua exclusão;
- e) O Pesquisador C avaliou os artigos excluídos e as justificativas de exclusão e os artigos presentes na interseção. O resultado foi o conjunto de artigos resultantes dos MSL.

3.3 Padrões de Requisitos de Software

A Engenharia de Requisitos (ER) é uma das fases iniciais do ciclo de desenvolvimento de sistemas de software, na qual o DRS é produzido. A qualidade desse documento contribui positivamente para qualidade do sistema de software como um todo e para o desenvolvimento, em menos tempo, garantindo a entrega nos prazos estipulados (BARCELOS; PENTEADO, 2017).

Com o intuito de obter DRS de melhor qualidade, PRS podem fornecer orientações sobre como especificar requisitos funcionais e requisitos não-funcionais. PRS podem ser utilizados em vários tipos de domínios de sistemas de software, trazendo benefícios importantes, tais como, aumento da produtividade, diminuição do tempo para escrever DRS e redução dos custos (CHERNAK, 2012). No MSL descrito nesta seção, o objetivo foi conhecer o estado-da-arte sobre PRS.

3.3.1 Planejamento

Nesta seção, é apresentado o planejamento do MSL realizado neste trabalho, de acordo com o modelo proposto na literatura (KEELE, 2007; KITCHENHAM, 2004). O objetivo foi identificar, classificar e catalogar estudos primários a respeito de PRS. Assim, foram elaboradas oito questões de pesquisa (QP) (Tabela 3.1). Essas questões visam apoiar a classificação das evidências disponíveis que auxiliam na identificação de lacunas e/ou estado da arte frente ao objetivo do MSL (KITCHENHAM *et al.*, 2010). Os tipos de estudos considerados na RQ6 são classificados conforme os tipos apresentados na Tabela 3.2 (WIERINGA *et al.*, 2006), sendo utilizados para orientar o desenvolvimento de MSL (PETERSEN *et al.*, 2008).

Tabela 3.1 - Questões de Pesquisa (QP) e Justificativas.

Sigla	Questão	Justificativa
QP1	Quais são os PRS existentes na literatura?	Conhecer os PRS utilizados/propostos pelos pesquisadores na literatura (estado-da-arte).
QP2	Quais são as notações utilizadas para a elaboração de PRS?	Conhecer as notações utilizadas para descrever os PRS pode auxiliar na escolha da notação mais adequada para utilização com a EROA.
QP3	Como os requisitos funcionais e não-funcionais são tratados nos PRS?	Saber como os requisitos funcionais e requisitos não-funcionais são padronizados nos DRS, para entender como essa padronização pode aprimorar a cobertura e a precisão na identificação de interesses na EROA.
QP4	Em quais contextos os PRS são utilizados/criados?	Identificar os contextos com mais incidência de PRS pode auxiliar na escolha dos padrões a serem utilizados neste trabalho.
QP5	Quais ferramentas/apoios computacionais foram encontradas para dar suporte na criação de PRS?	Caracterizar a utilização de ferramentas/apoios computacionais para a criação/utilização de PRS na elaboração de DRS.
QP6	Quais são os tipos de estudos propostos para PRS?	Distinguir tipos de estudos, por exemplo, trabalhos que propõem novas abordagens, daqueles que estendem as existentes e relatam experiências e opiniões de uso, daqueles baseados em evidências, obtidas a partir de estudos experimentais.
QP7	Quais são as medidas utilizadas para avaliar a qualidade dos PRS?	Conhecer as medidas para avaliar os PRS.
QP8	Qual é a quantidade de publicações por ano nos estudos encontrados?	Observar em quais anos houve mais ocorrências sobre PRS para justificar a relevância do tema deste trabalho.

Fonte: Do autor (2020).

Tabela 3.2 - Tipos de Estudos (adaptado de WIERINGA *et al.*, 2006).

Categoria	Descrição
Estudo de Validação	Apresenta avaliação de uma abordagem proposta em ambientes simulados (laboratórios) com experimentos controlados, estudos de caso ou prova de conceito.
Estudo de Avaliação	Apresenta avaliação prática de uma abordagem proposta com experimentos em ambiente industrial real. Esse tipo de estudo, em geral, é realizado em abordagens mais maduras, cujas forças foram avaliadas por meio de “estudos de validação”.
Solução Original	Apresenta a descrição de uma solução original para um problema. Os potenciais benefícios e aplicabilidade da solução proposta são apresentados por pequenos exemplos e bons argumentos pelos autores do estudo.
Solução Adaptada	Apresenta descrição de uma solução para um problema, mas é uma adaptação de uma solução existente. Uma adaptação pode ser considerada como uma solução suplementar ou uma solução que minimiza certas limitações da abordagem original. Da mesma forma, o tipo de estudo “Solução Original”, os potenciais benefícios e aplicabilidade da solução proposta são mostrados por exemplos e por argumentação dos autores do estudo.
Estudo Filosófico	Delinea uma nova maneira de olhar para as abordagens existentes e as estruturas na forma de taxonomia, estrutura conceitual ou catálogo.
Artigos de Opinião	Expressa a opinião pessoal de um(alguns) pesquisador(es) sobre os benefícios e/ou limitações de uma abordagem particular ou como a abordagem deve ser usada.
Relatos de Experiência	Consiste em depoimentos expressos por profissionais/pesquisadores sobre como as abordagens podem ser utilizadas na prática. É a experiência pessoal do(s) autor(es) do uso de uma abordagem particular.

Fonte: Do autor (2020).

Em seguida, uma *string* de busca para recuperação de documentos foi elaborada, utilizando o *framework* PICO (Tabela 3.3): i) **População (P)** - o objeto de estudo é a ER; ii) **Interesse (I)** - o interesse são padrões (*templates*); e iii) **Contexto (Co)** - no caso desse MSL, o contexto é software. Com os operadores lógicos OR e AND, juntamente com as palavras-chave e respectivos sinônimos, foram aplicados nos elementos do *framework* PICO. Assim, a seguinte *string* de busca foi elaborada:

(*pattern* OR *standard* OR *template* OR *model*) AND ("requirement document" OR "requirement specification") AND (software OR system OR product)

Tabela 3.3 - Estrutura para Elaboração da *String* de Busca Utilizando *Framework* PICO.

PICo	Termos
P	"requirement document" OR "requirement specification"
I	"pattern" OR "standard" OR "template" OR "model"
Co	"software" OR "system" OR "product"

Fonte: Do autor (2020).

Para responder às questões de pesquisa utilizando a *string* de busca, foram escolhidos os repositórios de artigos científicos ACM Digital Library⁷, Elsevier Engineering Village⁸, IEEE Xplore⁹, Science Direct¹⁰, Scopus¹¹ e Springer Link¹². Esses repositórios foram escolhidos por suportar (i) pesquisa avançada com utilização de palavras-chave, (ii) filtragem dos resultados por ano e área de publicação, (iii) filtragem por tipo de publicação e (iv) exportação do resultado da consulta em formato BibTex ou Endnote.

O retorno da pesquisa nesses repositórios foi armazenado em bases de dados. A busca nesses repositórios considerou documentos com conteúdo sobre PRS. Os **critérios de exclusão** utilizados na seleção dos estudos primários (artigos) desse MSL são apresentados na Tabela 3.4. Como **critério de inclusão**, foram considerados estudos primários recuperados a partir da *string* de busca não rejeitados por um ou mais critérios de exclusão.

3.3.2 Execução do MSL

Nessa fase, ocorreu a investigação nos seis repositórios de artigos científicos escolhidos, em busca de estudos relevantes para responder às questões de pesquisa. Para isso, a *string* de busca foi utilizada nesses repositórios, considerando a especificidade da máquina de busca (Tabela 3.5). Na *ACM Digital Library*, a pesquisa foi realizada utilizando o modo *Advanced Search* com filtro "*The ACM Guide to Computing Literature*". Na *Elsevier Engineering Village*, foi utilizada a função *Expert Search*, pois fornece aprimoramentos e flexibilidade ao incorporar lógica booleana avançada, com os filtros "*Language: english*". Na *IEEE Xplore*, a pesquisa foi realizada no modo *Advanced Search* com o filtro "*Content Type: Conference Publications*,

⁷ <https://dl.acm.org/>

⁸ <https://www.engineeringvillage.com>

⁹ <https://ieeexplore.ieee.org>

¹⁰ <https://www.sciencedirect.com>

¹¹ <https://www.scopus.com>

¹² <https://link.springer.com>

Journals & Magazines". Na *Science Direct*, foi utilizada a função *Advanced Search* sem filtros. Na *Scopus*, foi utilizada a função *Advanced* com o filtro "*Language: English*". Na *Springer Link*, foi utilizada a função *Search* com o filtro "*English, Computer Science e Article*". A pesquisa nos repositórios foi realizada no mês de outubro de 2020.

Tabela 3.4 - Critérios de Exclusão

Sigla	Critério
E1	O estudo está escrito em outro idioma, que não seja o inglês.
E2	A versão completa do estudo não está disponível na web.
E3	O estudo não trata do uso, adaptação/criação de PRS.
E4	O estudo é uma versão duplicada de outro estudo.
E5	O estudo é uma versão antiga de outro estudo.
E6	O estudo possui até 2 páginas.
E7	O estudo não é um estudo primário, mas um prefácio de anais de evento, capítulo de livro, tese de doutorado, entre outros.

Fonte: Do autor (2020).

Tabela 3.5 - String de Busca e Filtros Utilizados nas Bibliotecas.

Bibliotecas	String de Busca
ACM Digital Library	"query": {"pattern" OR "standard" OR "template" OR "model" OR "requirement document" OR "requirement specification"} AND ("software" OR "system" OR "product")} "filter": {Publisher: Association for Computing Machinery}
Elsevier Engineering Village	((((pattern OR standard OR template OR model) AND ("requirement document" OR "requirement specification") AND (software OR system OR product))) AND ({english} WN LA))
IEEE Xplore	(Pattern OR Template) AND ("Requirement Document" OR "Requirement Specification") AND (Software OR System OR Product) Filters Applied: Conferences Journals & Magazines
Science Direct	(pattern OR standard OR template OR model) AND ("requirement document" OR "requirement specification") AND (software OR system OR product)
Scopus	TITLE-ABS-KEY ((pattern OR standard OR template OR model) AND ("requirement document" OR "requirement specification") AND (software OR system OR product)) AND (LIMIT-TO (LANGUAGE, "English"))
Springer Link	(pattern OR standard OR template OR model) AND ("requirement document" OR "requirement specification") AND (software OR system OR product) within "English" "Computer Science" "Software Engineering" "Article"

Fonte: Do autor (2020).

Foram encontrados 2.589 documentos¹³ (Tabela 3.6), dos quais 302 documentos (11,7%) são da *ACM Digital Library*, 106 documentos (4,1%) são da *Elsevier Engineering Village*, 158 documentos (6,1%) são da *IEEE Xplorer*, 923 documentos (35,7%) são da *Science Direct*, 408 documentos (15,7%) são da *Scopus* e 692 documentos (26,7%) são da *Springer Link*. As referências com extensão de arquivo BibTex foram extraídas dos repositórios e armazenadas no software JabRef. Para obter os artigos resultantes do MSL, foram realizadas 4 etapas:

¹³ O resultado é considerado **documento**, pois nem todos são artigos, podendo ser *Inproceedings*, *Phdthesis*, *Incollection*, *Book*, *Inbook* e *Conference*, por exemplo.

- a) **Etapa 1.** Os documentos resultantes da utilização da *string* de busca nos repositórios foram armazenados em arquivos separados para serem analisados individualmente e excluídos os “não artigos” (por exemplo, foram removidos “*Inproceedings*”, “*Phdthesis*”, “*Incollection*”, “*Book*”, “*Inbook*”, “*TechReport*” e “*Conference*” para o atributo “*entryType*”). Isso foi feito para atender ao critério de exclusão E7 (Tabela 3.4). Dos 2.589 documentos, 248 documentos (9,6%) foram rejeitados (R) e 2.341 documentos (90,4%) foram aceitos (A). Dos artigos¹⁴ aceitos, 116 artigos (4,9%) são da *ACM Digital Library*, 87 artigos (3,6%) são da *Elsevier Engineering Village*, 116 artigos (4,9%) são da *IEEE Xplorer*, 923 artigos (39,3%) são da *Science Direct*, 407 artigos (17,3%) são da *Scopus* e 692 artigos (30,0%) são da *Springer Link*. Os metadados dos estudos foram rejeitados, pois, em análise com os pesquisadores, esses documentos ainda não foram avaliados ou não estão em formato de artigos;
- b) **Etapa 2.** Os 2.341 artigos resultantes da etapa anterior foram agrupados em um único arquivo (repositório) no software *JabRef*. Posteriormente, os artigos duplicados foram removidos, atendendo ao critério de exclusão E4 (Tabela 3.4), que resultou em 2.043 artigos (87,3%) aceitos e 298 artigos (12,7%) rejeitados. Dos artigos aceitos, 99 artigos (4,8%) são da *ACM Digital Library*, 50 artigos (2,4%) são da *Elsevier Engineering Village*, 82 artigos (4,0%) são da *IEEE Xplorer*, 839 artigos (41,1%) são da *Science Direct*, 345 artigos (16,9%) são da *Scopus* e 628 artigos (30,8%) são da *Springer Link*;

Tabela 3.6 - Resumo do Resultado do MSL.

Repositórios de Artigos Científicos	Busca Inicial	Etapa 1		Etapa 2		Etapa 3		Etapa 4	
		A	R	A	R	A	R	A	R
<i>ACM Digital Library</i>	302	116	186	99	17	21	78	1	20
<i>Elsevier Engineering Village</i>	106	87	19	50	37	13	37	1	12
<i>IEEE Xplorer</i>	158	116	42	82	34	9	73	0	9
<i>Science Direct</i>	923	923	0	839	84	37	802	1	36
<i>Scopus</i>	408	407	1	345	62	93	252	15	78
<i>Springer Link</i>	692	692	0	628	64	44	584	6	38
Total	2.589	2.341	248	2.043	298	217	1.826	24	193

Fonte: Do autor (2020).

- c) **Etapa 3.** O título, o resumo e as palavras-chave dos 2.043 artigos resultantes da etapa anterior foram lidos e, quando não se encaixavam na temática da pesquisa, os artigos foram rejeitados. Além disso, os critérios de exclusão foram reaplicados (Tabela 3.4). Ao final, 217 artigos (10,6%) foram aceitos e 1.826 artigos (89,4%) foram rejeitados. Houve maior incidência de estudos rejeitados, pois eles não apresentavam quaisquer informações sobre

¹⁴ Nesse momento, há apenas artigos científicos.

PRS, seguindo o critério de exclusão E3 (Tabela 3.4). Dos artigos aceitos, 21 artigos (9,7%) são da *ACM Digital Library*, 13 artigos (6,0%) são da *Elsevier Engineering Village*, 9 artigos (4,1%) são da *IEEE Xplorer*, 37 artigos (17,0%) são da *Science Direct*, 93 artigos (42,9%) são da *Scopus* e 44 artigos (20,3%) são da *Springer Link*;

- d) **Etapa 4.** Os 217 artigos selecionados na etapa anterior foram lidos por completo e os critérios de exclusão foram novamente aplicados (Tabela 3.4), resultando 24 artigos (11,0%) aceitos e 193 artigos (89,0%) rejeitados. Dos artigos aceitos, 1 artigo (4,2%) é da *ACM Digital Library*, 1 artigo (4,2%) é da *Elsevier Engineering Village*, nenhum artigo (0%) é da *IEEE Xplorer*, 1 artigo (4,2%) é da *Science Direct*, 15 artigos (62,4%) são da *Scopus* e 6 artigos (25,0%) são da *Springer Link*. Foram rejeitados os artigos que não responderam às questões de pesquisa.

3.3.3 Análise dos Resultados

Nessa fase, foi realizada a compilação das informações presentes nos artigos resultantes da fase Execução. A referência completa aos 24 artigos (estudos primários) resultantes do MSL é apresentada no APÊNDICE A.

Em resposta à questão de pesquisa QP1

QP1: Quais são os PRS existentes na literatura?

foram identificados 105 PRS nas mais diversas atividades, por exemplo, PRS para sistemas de software colaborativos ou para sistemas de software governamentais e acadêmicos. Esses PRS foram agrupados em 12 catálogos, apresentando amplo conjunto de finalidades. As categorias relatadas incluíam diversos domínios e tipos de padrões de acordo com sua aplicabilidade nas atividades da ER e configurações de sistemas de software do mundo real. Na Tabela 3.7, são apresentados os resultados encontrados, incluindo o catálogo, a quantidade de PRS nesses catálogos e os artigos onde foram encontrados.

Em P1, P7, P12, P14, P16 e P20, são apresentados mais de um catálogo de PRS. Além disso, em P1 são apresentados mais de um PRS, pois são realizadas avaliações entre PRS de Segurança e outros dois PRS para Gestão, onde os autores mostram que os PRS propostos são mais utilizados em diferentes cenários de sistemas de software. Em P7, foi apresentada uma proposta de PRS para Sistemas Embarcados e modelos genéricos de DRS. Ambos PRS foram descritos em linguagem natural, viabilizando a escrita de mais fácil compreensão e interpretação pelos desenvolvedores, segundo os autores. Em P12 e P14, foram propostos PRS para requisitos não funcionais, tais como, Usabilidade, Confiabilidade, Desempenho, Suporte

e Portabilidade. Para isso, os autores utilizaram um apoio computacional para gerar os PRS, visando à confiabilidade e à completude dos DRS.

Tabela 3.7 - PRS Encontrados no MSL.

Catálogo de PRS para	Exemplo de Padrões Identificados	Quant. de Padrões	Referências
Caso de Uso	Alarme, Comando, Confirmação da Etapa, Monitor, Solicitação, Confirmar Recebimento, Recuperar	2	P6, P23 (Total: 2 artigos)
Portabilidade	Diferentes Arquiteturas	2	P9, P14 (Total: 2 artigos)
Processo de Engenharia de Requisitos	Confiabilidade, Desempenho, Suporte, Manutenibilidade, Armazenamento	6	P12, P14 (Total: 2 artigos)
Segurança	Protege os Dados Transmítidos, Protege a Atividade de Negócios, Protege os Serviços de Negócios, Acesso não Autorizado, Protege os Dados Armazenados, Confiabilidade	5	P1, P8, P14, P20, P24 (Total: 5 artigos)
Sistemas de Informação (CRUD - <i>Create, Read, Update e Delete</i>)	Transação Processo, Execução Condição, Padrões de Domínio dos Sistemas de Informação, Sistema de Ordem de Serviço (Visão geral), Registro, <i>Login</i> Pesquisa, Manter Informação, Operações Básicas, Processamento de Transações, Informações Gerenciais, Regras de Negócio	10	P17, P18 (Total: 2 artigos)
Sistemas de Software Governamentais e Acadêmicos	<i>E-Government, e-Trademark</i> , Auditoria, Administração de Orçamento, Investimento, Gerenciamento de Performance, Apoio a Projetos, Gerenciamento de Partes Interessadas, Software de Anúncio, Desempenho	7	P3, P15, P16 (Total: 3 artigos)
Sistemas em Tempo Real	Reservas <i>on-line</i> , Serviços Bancários <i>on-line</i> , Sistemas Móveis, Sistemas Eleitorais <i>on-line</i> , Negócios <i>on-line</i> e Registro <i>on-line</i> , <i>E-commerce</i> , Peça Ajuda, Sistema Móvel	12	P5, P16 (Total: 2 artigos)
Sistemas Embarcados	Sensor de Atuador, Canal, Controlador Decompor, Examinador, Manipulador de Falhas, Moderador, Monitor de Atuador, Máscara, Interface do Usuário, <i>Watchdog</i>	4	P7, P19, P20, P23 (Total: 4 artigos)
Software de Gestão	Gestão de Custos, Gestão de Produtos, Gestão de Tráfego Aéreo, Gestão de Arquivo, Gestão de Recursos Humanos, Gestão do programa, Gestão do Programa de Pesquisa	21	P1, P22 (Total: 2 artigos)
Software de Prateleira	Armazenamento Alternativo de Dados, Autenticação, Autorização, <i>Logoff</i> Automático, Disponibilidade, <i>Backup</i> , Capacidade de Usuários Simultâneos, Capacidade de Dados, Troca de Dados, Proteção de Transmissão de Dados, Linguagem de Desenvolvimento, Documentação, Tempo de Inatividade, Alertas de Falha, Procedimentos de Instalação, Capacidade de Aprendizado da Interface, Interface de Idioma Tempo, Tipo de Interface, Interoperabilidade com Sistemas Externos, <i>Log</i> , Ajuda <i>online</i> , Plataforma, Precisão, Procedimentos de Recuperação, Proteção de Dados Armazenados, Procedimentos de Atualização, Tempo de Atividade, Capacidade de Usuários, Portabilidade	29	P11 (Total: 1 artigo)
Usabilidade	Acessibilidade, Interface, Usuário, Jogabilidade	3	P1, P12, P13 (Total: 3 artigos)

(continua)

Tabela 3.7 - PRS Encontrados no MSL (continuação).

Catálogo de PRS para	Exemplo de Padrões Identificados	Quant. de Padrões	Referências
Outros PRS	Comportamento Baseado em Modo, PRS para Decomposição de Requisitos de Segurança, PRS para Análise de Sistema em Nuvem, PRS para Análise de Lei de Padrões de Mudança, Abordagem para Especificação de DRS, Modelagem em Linguagem Natural, PRS para Qualidade dos DRS	4	P2, P7, P10, P21 (Total: 4 artigos)

Fonte: Do autor (2020).

Em P16, foi proposto PRS para elaborar DRS em linguagem natural para sistemas de software acadêmico, bancário, comercial e médico, tendo como propósito a automatização dos PRS para esses tipos de sistemas de software. Em P20, foi proposto PRS para Segurança e para sistemas embarcados, analisando uma proposta para a atuação de ambos os PRS em prol de sistemas embarcados na indústria automobilística e propondo que a segurança deve estar presente nos sistemas embarcados automotivos, a fim de garantir a segurança no processo de produção.

Conclusão. Foi constatado que, na maioria dos artigos, os catálogos de PRS informados incluíam diversos domínios que podem ser aplicados.

Em resposta à questão de pesquisa QP2

QP2: Quais são as notações utilizadas para a elaboração de PRS?

a notação mais encontrada nos artigos selecionados foi Casos de Uso. Nos artigos, foram encontradas essas notações na maioria dos DRS gerados após a utilização dos PRS. Percebeu-se que a utilização dessa notação é a mais comum entre os Engenheiros de Requisitos por ter maior adesão pelos desenvolvedores e apresentar fácil entendimento na descrição de DRS (JACOBSON; NG, 2004). Contudo, alguns autores também utilizaram ferramentas/apoios que padronizam Diagramas UML, Diagramas de Classe, Diagramas de Sequência, entre outros. Na Tabela 3.8, são mostradas as notações encontradas e suas respectivas referências.

Tabela 3.8 - Notações Utilizadas em DRS.

Notação	Referências
Casos de Uso	P3, P4, P6, P8, P9, P10, P12, P14, P16, P19, P22, P23 (Total: 12 artigos)
Diagramas UML	P4, P6, P10, P12, P14, P15, P16, P21 (Total: 8 artigos)

(continua)

Tabela 3.8 - Notações Utilizadas em DRS (continuação).

Notação	Referências
Diagramas de Sequência	P12, P16 (Total: 2 artigos)
Diagrama de Classe	P6, P16 (Total: 2 artigos)
Documentos de Requisitos de Software	P1, P2, P7, P13, P17, P18, P20, P24 (Total: 8 artigos)
Outras Notações	P5 (Total: 1 artigo)

Fonte: Do autor (2020).

Em P4, P6, P12, P14 e P16, foram encontradas mais de uma notação para os PRS. Em P4, os autores utilizaram as notações de Casos de Uso e diagramas UML, pois, por se tratar de PRS para sistemas colaborativos, a ideia foi elaborar DRS claros e objetivos com ambos os tipos de notação escolhidos. Em P6, os PRS permitem gerar especificações de requisitos de software, utilizando a notação para Casos de Uso e verificar se os DRS gerados possuem consistência para serem utilizados em sistemas reais. Em P12 e P14, os autores utilizaram os tipos de notação de Casos de Uso, Diagramas de Sequência e diagramas UML para avaliar os DRS gerados, utilizando PRS para Usabilidade, Confiabilidade, Desempenho, Segurança, Manutenibilidade, Portabilidade e Suporte. Em P16, a notação de Casos de Uso foi utilizada para elaborar DRS com os PRS para sistemas em tempo real, aproximando-se da escrita textual e utilizando frases curtas que sintetizam a funcionalidade do sistema de software especificado.

Conclusão. Foram destacadas as notações mais utilizadas, mostrando onde há mais incidências de PRS. Com isso, a maioria dos artigos utiliza a notação de Casos de Uso, pois, pelo que foi identificado, a sua aplicação facilita a compreensão dos requisitos.

Em resposta à questão de pesquisa QP3

QP3: Como os requisitos funcionais e não-funcionais são tratados nos PRS?

em 43,5% dos artigos, os requisitos funcionais são destacados em maior quantidade para PRS utilizados para elaborar DRS. Os requisitos não-funcionais foram considerados em 21,7% dos artigos, enquanto os requisitos funcionais e os requisitos não-funcionais juntos foram considerados em 34,8%. Isso mostra que, na maioria dos PRS encontrados e catalogados, os requisitos funcionais são os mais padronizados (Tabela 3.9). Nos artigos que tratam os requisitos funcionais, são utilizados PRS para CRUD, Sistemas de Gestão e Sistemas Acadêmicos, por exemplo. Nos artigos que abordam apenas os requisitos não-funcionais, nota-

se o uso de PRS para Usabilidade, Segurança e Portabilidade, por exemplo. Nos artigos que abordam os requisitos funcionais e os requisitos não-funcionais (juntos), a ideia foi validar os PRS em um sistema de software completo, ou seja, PRS foram utilizados na elaboração de DRS e, posteriormente, avaliados em DRS sem uso de PRS.

Tabela 3.9 - Tipos de Requisitos Contemplados nos Artigos.

Tipo de Requisitos	Referências
Funcional	P2, P3, P5, P6, P16, P17, P18, P20, P21, P23, P24 (Total: 11 artigos - 45,8%)
Não-Funcional	P7, P8, P9, P13, P19 (Total: 5 artigos - 20,8%)
Funcional e Não-Funcional	P1, P4, P10, P11, P12, P14, P15, P22 (Total: 8 artigos - 33,4%)

Fonte: Do autor (2020).

Em P2, P3, P5, P6, P16, P17, P18, P20, P21, P23 e P24, são apresentados PRS para requisitos funcionais. Esse tipo de requisito é encontrado na maioria dos artigos, sendo que, em P2, P16 e P21, os autores apresentaram PRS escritos em linguagem natural, pois mostraram que podem ser facilmente entendidos pelos clientes e pelos desenvolvedores. Em P2, antes dos autores apresentarem os PRS, foi aplicada uma abordagem que consiste em selecionar os requisitos funcionais de um sistema de software e encontrar as similaridades entre eles. Em seguida, foram propostos PRS de acordo com o resultado da abordagem. Em P3, P5, P6, P17, P20 e P23, foram descritos PRS para sistemas de software de diferentes utilizações; dentre eles, destacam-se PRS para Sistemas de Gestão e Sistemas Web. Em P23, são apresentados PRS para Linhas de Produto de Software, utilizando abordagem para modelar os Casos de Uso. Em P18, os autores apresentaram PRS de requisitos funcionais para CRUD, sendo realizada uma análise em um conjunto de requisitos funcionais, encontrando similaridade entre esses requisitos. Assim, os autores apresentaram como resultado PRS para CRUD, para Processamento de Transações e para Informações de Gerenciamento, sendo aplicável na maioria dos sistemas de software. Em P24, é apresentado PRS para confiabilidade em que abrangem alguns requisitos de segurança, capacidade de manutenção, interoperabilidade e flexibilidade.

Em P7, P8, P9, P13 e P19, foram apresentados PRS para requisitos não-funcionais. Em cada artigo, são descritos PRS para requisito não-funcional em diferentes cenários. Por exemplo, em P7, são apresentados PRS para requisitos não-funcionais escritos em linguagem natural; em P8, P9 e P13, são apresentados PRS para Usabilidade, Segurança, Confidencialidade,

Integridade e Portabilidade; e, em P19, é apresentado PRS para requisitos não-funcionais de Sistemas de Controle de Instalações Sem Fio para empresas com operações a distância.

Em P1, P4, P10, P11, P12, P14, P15 e P22, foram apresentados PRS para requisitos funcionais e não-funcionais, sendo que os autores elaboraram DRS de sistemas de software realizando comparações entre os requisitos de cada sistema. Por exemplo, em P1, P4 e P22, foram apresentados requisitos de sistemas de software de Gestão de Custos, de Gestão de Produto e de Gestão de Tráfego Aéreo, diretamente relacionados à Usabilidade e à Confiabilidade. Em P10 e P14, foram apresentados PRS para especificação dos requisitos funcionais e não-funcionais e uma abordagem para avaliar a qualidade dos DRS elaborados. Em P11 e P15, foram apresentados PRS para requisitos funcionais e não-funcionais de sistemas de software de prateleira e acadêmicos, mostrando que, em ambos os cenários, pode-se encontrar PRS similares que possam ser reaproveitados em diferentes contextos.

Conclusão. Verifica-se que, na maioria dos artigos, é utilizado algum PRS que contempla requisitos funcionais, requisitos não-funcionais ou ambos. Em alguns desses artigos, é destacada a importância desses requisitos serem padronizados e terem PRS em diferentes contextos.

Em resposta à questão de pesquisa QP4

QP4: Em quais contextos os PRS são utilizados/criados?

foi possível constatar a existência de apenas dois contextos: i) Sistemas Embarcados (SE), são sistemas computacionais especializados que fazem parte de uma máquina ou de um sistema maior (GAY *et al.*, 2003); e ii) Sistemas de Informação (SI), são sistemas que auxiliam os usuários a realizar suas atividades de forma automatizada (CASTRO *et al.*, 2002). Pensando nisso, foram extraídos os contextos nos quais têm mais presença de PRS (

Tabela 3.10).

Para 19 artigos, os PRS foram utilizados no contexto de SI, sendo a maioria. Muitos desses ocorrem por serem os mais presentes na maioria dos sistemas de software utilizados por usuários de forma geral e cotidiana. Em P1, P2, P3, P4, P5, P6, P8, P9, P10, P11, P12, P13, P14, P15, P16, P17, P18, P21 e P22, são apresentados sistemas utilizados para garantir que a informação seja automatizada e compartilhada com os usuários. A presença de PRS nesse contexto pode melhorar o entendimento e a escrita dos DRS.

Tabela 3.10 - Contextos.

Contexto	Referências
Sistemas de Informação	P1, P2, P3, P4, P5, P6, P8, P9, P10, P11, P12, P13, P14, P15, P16, P17, P18, P21, P22 (Total: 19 artigos - 79,2%)
Sistemas Embarcados	P7, P19, P20, P23, P24 (Total: 5 artigos - 20,8%)

Fonte: Do autor (2020).

Em P7, P19, P20, P23 e P24, há a presença de PRS para aprimorar a escrita de sistemas de software microcontrolados, como é em P19 e P20, em que os PRS são para sistemas de instalações sem fio e sistemas eletrônicos automotivos. Em P24, são apresentados PRS para SE que auxiliam na automação industrial, visando a confiabilidade desses sistemas de software.

Conclusão. Percebe-se presença menor de PRS no contexto de SE, indicando oportunidades de estudos/pesquisas futuras. Porém, para atingir o objetivo neste trabalho, é oportuna a utilização de PRS no contexto de SI, sendo o contexto mais abordado na literatura.

Em resposta à questão de pesquisa QP5

QP5: Quais ferramentas/apoios computacionais foram encontradas para dar suporte na criação de PRS?

nos estudos encontrados, pode-se notar a utilização de ferramentas/apoios computacionais para utilizar PRS na escrita dos DRS (Tabela 3.11), as quais são, na maioria, semiautomáticas¹⁵. Além disso, foram encontradas ferramentas/apoios computacionais que padronizam DRS, utilizando abordagens, métodos ou normas com PRS. A maioria dos artigos utiliza alguma ferramenta/apoio computacional. Em outros artigos, são realizados testes comparativos entre ferramentas/apoios computacionais ou é avaliada a sua utilização com alguma abordagem, método ou modelos propostos de PRS.

Em P1, P2, P4, P8, P11, P12, P13 e P20, são descritas abordagens, métodos ou normas para descrever os PRS utilizados na elaboração de DRS. Em P1, P8, P12, P13 e P20, são descritos métodos utilizados para a modelagem de PRS para Segurança, Usabilidade, Autenticação e Autorização. Em P4, são apresentados PRS para *E-Commerce*, sendo avaliado com a norma IEE830. Em P2, é apresentada uma abordagem para gerar PRS, que se baseia em *clustering*, empregando um agrupamento conceitual para classificar as funções dos requisitos funcionais. Ao agrupar esses requisitos, são gerados PRS com base nos agrupamentos

¹⁵ Realizam uma ou mais etapas dos processos de elaboração de PRS de forma automática, mas ainda necessitam da análise do Engenheiro de Software.

resultantes. Em P11, é apresentado o método PABRE (*Pattern-Based Requirements Elicitation*), gerado a partir de catálogos de PRS presentes na literatura. Esse método busca encontrar a similaridade entre PRS desse catálogo e proporcionar ao Engenheiro de Requisitos um novo PRS a ser utilizado.

Tabela 3.11 - Ferramentas/Apoios Computacionais, Abordagens, Métodos ou Normas.

Tipos	Referências
Ferramenta/Apoio Computacional	P3, P5, P6, P7, P9, P10, P14, P15, P16, P17, P18, P19, P21, P22, P23, P24 (Total: 16 artigos - 66,7%)
Abordagens, Métodos ou Normas	P1, P2, P4, P8, P11, P12, P13, P20 (Total: 8 artigos - 33,3%)

Fonte: Do autor (2020).

Em P3, P5, P6, P7, P9, P10, P14, P15, P16, P17, P18, P19, P21, P22, P23 e P24, são apresentadas e avaliadas ferramentas/apoios computacionais que auxiliam os Engenheiros de Requisitos criarem PRS. Em P3, é apresentado um *plugin* para a *Integrated Development Environment* (IDE) Eclipse, suportando o desenvolvimento de diagramas e gerando PRS a partir desses diagramas. Em P5, é apresentada uma extensão na IDE Visual Studio chamada CSRML (*Collaborative Systems Requirements Modeling Language*), que fornece PRS necessários para elaborar DRS para sistemas colaborativos aos Engenheiros de Requisitos. Em P6, é apresentada a ferramenta computacional MaramaAIC (*Automated Inconsistency Checker*), que fornece prototipagem rápida de ponta a ponta, utiliza uma abordagem em conjunto com bibliotecas de PRS que ajuda a capturar requisitos e verifica a consistência de DRS expressos em linguagem natural.

Em P7, é apresentada a ferramenta computacional GATE NPL (*Natural Language Processing*), em que existe uma linguagem de correspondência de PRS baseada em expressões, isto é, consiste em elaborar DRS com um conjunto de frases compostas por um conjunto de PRS. Em P9, é apresentada a ferramenta computacional Software-FUR (*Functional User Requirements*), desenvolvida para descrever PRS para Portabilidade de sistemas de software. Em P10, os autores apresentam a ferramenta computacional Elicito, sendo utilizada para apoiar as atividades de ER, fornecendo PRS e validando os dados gerados com base no domínio da aplicação. Em P14, é apresentada a ferramenta computacional MetaEdit+, cujo recurso possibilita gerar os PRS e seus modelos, mantendo sincronizados qualquer alteração nos gráficos, objetos, propriedades, relacionamento entre os requisitos e suas funções geradas no final do processo de geração dos DRS. Em P16, é apresentada a ferramenta computacional

LUCAMTool (*Language of Use Case to Automate Models Tool*), que gera os PRS e, automaticamente, Digramas de Classe e Diagramas de Sequência.

Em P17, é apresentada a ferramenta computacional baseada em uma técnica padronizada que orienta Engenheiros de Requisitos, por meio de PRS, a escreverem DRS com consistência, completos e corretos. Em P18, é apresentada uma ferramenta computacional para gestão de PRS para operações básicas (*e.g.*, CRUD). Em P19, é apresentada a ferramenta computacional RE-Tools para elaborar DRS, utilizando PRS para requisitos não funcionais. Em P21, é apresentada a ferramenta computacional GenLangUML (*Generating Natural Language from UML*), que permite a criação de diagramas UML e, posteriormente, gera PRS para escrever DRS em linguagem natural. Em P22, é apresentada a ferramenta computacional REDEPEND, que fornece aos Engenheiros de Requisitos funções de modelagem e análise de PRS para escrever DRS aos sistemas de software. Em P23, é apresentada a ferramenta computacional PUMConf (*Product Line Use Case Model Configurator*), que verifica automaticamente a consistência dos modelos de Linhas de Produto de Software, identifica a ordem parcial de decisões a serem tomadas, determina decisões contraditórias e gera PRS para escrever DRS de domínios de produtos específicos. Em P24, é citada a modelagem de uma ferramenta de método formal para auxiliar na estruturação do PRS de confiabilidade e apoiar os engenheiros nos processos.

Conclusão. Foi constatado que, na maioria das ferramentas/apoios, são realizadas tarefas de forma semiautomática, gerando, ao final, DRS padronizado. Isto é, na maioria dos processos de geração de PRS, é necessária a configuração inicial dos Engenheiros de Requisitos para inicializar o processo.

Em resposta à questão de pesquisa QP6

QP6: Quais são os tipos de estudos propostos para PRS?

foi possível classificar os artigos conforme a proposta de tipos de estudos apresentadas na Tabela 3.2, os artigos e a classificação, também de acordo com os tipos de estudos, são apresentados na Tabela 3.12.

Tabela 3.12 - Tipos de Estudos.

Tipos de Estudo	Referências
Estudo de Validação	P2, P5, P7, P9, P10, P11, P12, P14, P15, P19, P21, P22, P24 (Total: 13 artigos - 54,1%)
Estudo de Avaliação	P8, P16, P17, P18, P20, P23 (Total: 6 artigos - 25,0%)
Solução Original	P3, P6, P13 (Total: 3 artigos - 12,5%)
Solução Adaptada	P1 (Total: 1 artigo - 4,2%)
Estudo Filosófico	- (Total: 0 artigo - 0%)
Artigos de Opinião	- (Total: 0 artigo - 0%)
Relatos de Experiência	P4 (Total: 1 artigo - 4,2%)

Fonte: Do autor (2020).

Na maioria dos artigos, houve maior incidência em estudos de avaliação de PRS, pois os autores apresentam abordagens para criação, validação e avaliação de PRS com experimentos controlados, estudos de caso ou prova de conceito. Em P2, P5, P7, P9, P10, P11, P12, P14, P15, P19, P21, P22 e P24, são apresentadas abordagens utilizadas para criação/utilização de PRS, realizando experimentos que avaliam PRS e DRS gerados.

Em P8, P16, P17, P18, P20 e P23, há avaliação de abordagens propostas de forma prática ao gerar PRS, sendo utilizadas abordagens mais maduras para PRS gerados/criados e realizando experimentos em ambientes reais de sistemas de software. Em P3, P6 e P13, são apresentados PRS para diferentes contextos de sistemas de software e potenciais benefícios e aplicabilidade. Para isso, são apresentadas avaliações em diferentes cenários para avaliar e argumentar sobre a solução apresentada. Em P1, é apresentada uma solução para PRS, porém é uma adaptação de uma solução existente, na qual complementam a solução original, mostrando benefícios para PRS. Não houve incidências de estudos filosóficos e artigos de opinião, inferindo que não apresentam aplicabilidade sobre o tema, pois, nos artigos encontrados, os autores aplicaram seus estudos na ideia de criar/avaliar PRS. Em P4, são apresentados métodos em que a abordagem para PRS foi utilizada, baseando-se em experiências dos autores para escrever DRS em um cenário de desenvolvimento de sistemas de software.

Conclusão. Pode ser verificado nos artigos que os estudos de validação tiveram maior porcentagem com 52,2% dos artigos, estudos de avaliação com 26,1% dos artigos, solução original com 13,1% dos artigos e solução adaptada e relatos de experiência com 4,3% dos artigos. Com isso, nota-se que os autores estão buscando criar abordagens para PRS, visando aplicar novos conceitos e elementos presentes na literatura.

Em resposta à questão de pesquisa QP7

QP7: Quais são as medidas utilizadas para avaliar a qualidade dos PRS?

foram encontradas algumas formas de medir e avaliar a qualidade de PRS. Porém, em alguns artigos, foram realizadas avaliações de DRS, gerados com e sem a utilização de PRS, verificando em qual DRS os usuários aproveitaram melhor o conteúdo e em qual obtiveram melhor entendimento do que estava escrito (Tabela 3.13).

Tabela 3.13 - Formas para Avaliar PRS.

Avaliação/ Medidas	Referências
Avaliação Empírica	P1, P3, P5, P10, P13, P17, P23, P24 (Total: 8 artigos - 30,5%)
Normas ISO/IEC	P9 (Total: 1 artigo - 4,3%)
Precisão, Cobertura e <i>F-Measure</i>	P6, P7, P12 (Total: 3 artigos - 13,0%)
Validação dos DRS	P2, P4, P8, P11, P14, P15, P16, P18, P19, P20, P21, P22 (Total: 12 artigos - 52,2%)

Fonte: Do autor (2020).

Os autores buscaram realizar formas de avaliar/medir PRS. Na maioria dos artigos, a forma utilizada foi avaliar DRS gerados a partir dos PRS criados/utilizados. Em P2, P4, P8, P11, P14, P15, P16, P18, P19, P20, P21 e P22, é apresentada a forma de medir/avaliar, por meio de comparações, DRS com e sem a utilização de PRS. A ideia é Engenheiros de Requisitos experientes e/ou alunos de graduação e de pós-graduação verificarem se DRS elaborados com PRS atendem a critérios de, por exemplo: i) **aceitabilidade**, requisitos devem ser aceitáveis para as partes interessadas; ii) **ambigüamente ética**, requisitos devem ter apenas uma interpretação; iii) **integridade**, requisitos no DRS devem estar completos; iv) **verificabilidade**, DRS devem ter um critério de aceitação associado para verificá-los depois de implementar os requisitos; v) **compreensível**, medida de coeficiente de correlação que varia de 0 a 1; vi) **performance**, DRS escritos com PRS aumentou a produtividade dos sistemas de software desenvolvidos; e vii) **qualidade**, verificar se os DRS atendem às normas de qualidade na

especificação de requisitos. Dessa forma, caso os DRS elaborados com PRS atendessem a esses critérios, os autores considerariam que PRS foram eficientes e eficazes para aquele contexto específico (e.g., PRS para Segurança, CRUD e Usabilidade).

Em P1, P3, P5, P10, P13, P17, P23 e P24, os autores realizaram uma avaliação empírica dos DRS gerados com PRS. Essa avaliação consistiu em especialistas observarem DRS gerados e os avaliarem com base em suas experiências. Eles avaliaram PRS para Usabilidade, Segurança e *Login*. Em P6, P7 e P12, os autores utilizaram medidas para avaliar PRS, são elas: i) **Precisão**, verificando se os DRS atendem aos diferentes cenários de sistemas de software; ii) **Cobertura**, verificando se os DRS elaborados com PRS cobrem os requisitos; e iii) **F-Measure**, verificando, na média ponderada entre cobertura e precisão, DRS elaborados com PRS. Em P9, é realizada a avaliação da qualidade de PRS utilizando a norma ISO/IEC 9126. Essa avaliação é realizada após a concepção de PRS para verificar sua qualidade, após os usuários utilizarem. Em P24, os autores utilizam a avaliação empírica com grande amostragem de participantes, que possuem grau de conhecimento no PRS de confiabilidade e utilizam o *Goal Question Metrics* (GQM) para definir os objetivos da avaliação.

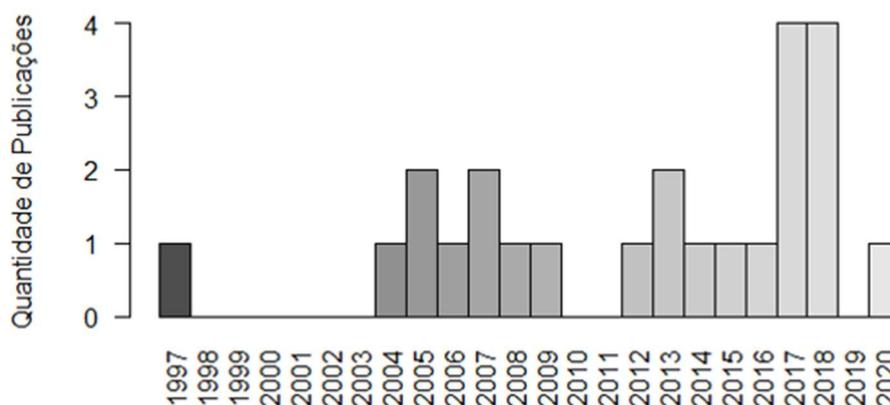
Conclusão. Foram encontradas medidas utilizadas para avaliar PRS e outras que fornecem apenas suporte. Na maioria dos artigos, foram realizadas apenas comparações de DRS elaborados com e sem utilizar PRS, mostrando como DRS escritos com PRS podem ser bem aproveitados para diferentes contextos do que DRS escritos sem PRS.

Em resposta à questão de pesquisa QP8

QP8: Qual é a quantidade de publicações por ano nos estudos encontrados?

o conjunto de artigos selecionados representam quantidade constante de obras ao longo dos anos, com exceção dos anos de 1998 a 2003, 2010 a 2011 e 2019 a 2020, que não tiveram artigos publicados (Figura 3.1). Nos demais anos, houve a publicação de, pelo menos, um artigo.

Figura 3.1 - Publicações por Ano.



Fonte: Do autor (2020).

Conclusão. É possível verificar que a quantidade de publicações sobre PRS tem quantidade constantes ao longo dos anos. Na maioria dos artigos, as publicações vêm crescendo a partir de 2017, com cerca de 3 publicações por ano. Porém, em 2019, não houve publicações, interrompendo esse crescimento. Em 2020, há uma publicação sobre PRS. Pode-se perceber que as publicações não são contínuas, porém as publicações existentes são de grande contribuição para os engenheiros de software.

3.4 Engenharia de Requisitos Orientada a Aspectos

A EROA suporta a separação das propriedades de corte transversal dos estágios iniciais do desenvolvimento e a identificação e classificação dos interesses, sendo mapeamentos e influenciando diretamente nos estágios posteriores ao desenvolvimento. Isso torna possível identificar conflitos e estabelecer possíveis compensações no início do ciclo de desenvolvimento e promove a rastreabilidade de requisitos e restrições de escopo amplo ao longo do desenvolvimento, manutenção e evolução do sistema. A modularização e a rastreabilidade aprimoradas obtidas por meio da separação precoce dos interesses podem desempenhar papel central na construção de sistemas resilientes a mudanças imprevistas, atendendo às necessidades de adaptabilidade de domínios voláteis, como bancos, telecomunicações e comércio eletrônico (SAMPAIO *et al.*, 2007; CHITCHYAN *et al.*, 2005; GRUNDY, 1999).

O objetivo é promover melhorias quanto à separação dos interesses nas fases iniciais do desenvolvimento de sistemas de software, pois, quanto mais tarde for realizada a identificação de interesses, maior o esforço necessário para sua modularização (SAMPAIO *et al.*, 2007). Com quantidade considerável de pesquisas nos últimos anos, foi realizado um MSL para coleta de trabalhos que utilizam as atividades da EROA para o contexto de interesses no

desenvolvimento de sistemas de software. A partir disso, espera-se obter conhecimento das pesquisas existentes até o momento.

3.4.1 Planejamento

Nesta seção, é apresentado o planejamento do MSL realizado neste trabalho, de acordo com o modelo proposto na literatura (KEELE, 2007; KITCHENHAM, 2004). Esse MSL consiste em uma extensão de um MSL anterior sobre abordagens para EROA (PARREIRA JÚNIOR; PENTEADO, 2014), no qual foram investigados estudos primários (artigos) sobre EROA publicados até 2012 nos repositórios de artigos científicos *Scopus*¹⁶, *ACM Digital Library*¹⁷ e *IEEE Xplorer*¹⁸. Com o intuito de estender este trabalho, por meio da condução de um novo MSL, foram realizadas buscas por estudos publicados a partir de 2011. Optou-se por ter interseção de dois anos (2011 e 2012) com o MSL anterior, pois os repositórios podem apresentar *delay* de atualização (armazenamento) de artigos científicos. Sendo assim, se o MSL realizado neste trabalho buscasse por estudos publicados a partir de 2012, ele poderia desconsiderar trabalhos não analisados no MSL anterior. O objetivo desse MSL, assim como o do anterior, foi identificar, classificar e catalogar abordagens para EROA. Com base nesse objetivo, foram utilizadas as quatro questões de pesquisa (QP) do MSL anterior (Tabela 3.14).

Em seguida, a mesma *string* de busca do MSL anterior para recuperação de documentos foi utilizada. Essa *string* foi organizada para atender o *framework* PICo, que possui três elementos (Tabela 3.15): i) **População (P)** - o objeto de estudo é ER; ii) **Interesse (I)** - o interesse são abordagens (métodos, técnicas e ferramentas) em geral; e iii) **Contexto (Co)** - o contexto é EROA.

Tabela 3.14 - Questões de Pesquisa (QP) e Justificativas.

Sigla	Questão	Justificativa
QP1	Quais são as abordagens (técnicas, ferramentas e métodos) para EROA disponíveis na literatura e quais atividades elas contemplam?	Identificar abordagens para EROA que mais se alinham com a proposta deste mestrado, ou seja, estudar o impacto da utilização de PRS no processo de identificação e classificação de interesses. É importante salientar que, neste trabalho, o termo abordagem abrange os conceitos de “técnicas”, “ferramentas” e “métodos”.
QP2	Quais são os tipos de estudos propostos para abordagens para EROA?	Distinguir tipos de estudos, por exemplo, trabalhos que (i) propõem novas abordagens, daqueles que estendem as existentes e (ii) relatam experiências e opiniões de uso, daqueles baseados em evidências, obtidas a partir de estudos experimentais. Tudo isso auxilia na busca por uma abordagem para EROA mais adequada para utilização neste mestrado.

(continua)

¹⁶ <https://www.scopus.com>

¹⁷ <https://dl.acm.org/>

¹⁸ <https://ieeexplore.ieee.org>

Tabela 3.14 - Questões de Pesquisa (QP) e Justificativas (continuação)

Sigla	Questão	Justificativa
QP3	Quais são os “paradigmas” ¹⁹ utilizados pelas abordagens para EROA para identificação e classificação de interesses?	Identificar o paradigma mais utilizado pelas abordagens para o resultado da integração desse paradigma com PRS favorecer outras abordagens para EROA, além daquela escolhida como alvo de estudo neste mestrado.
QP4	Em quais eventos (e.g., conferências, workshops e periódicos) os artigos sobre PRS foram publicados?	Subsidiar (futuros) pesquisadores para nortear suas pesquisas sobre abordagens para EROA.

Fonte: PARREIRA JÚNIOR; PENTEADO (2014).

Tabela 3.15 - Estrutura para Elaboração da *String* de Busca Utilizando *Framework* PICO.

PICO	Termos
P	"requirements engineering"
I	"approach" OR "approaches" OR "tool" OR "tools" OR "method" OR "methods" OR "technique" OR "techniques"
Co	"aspect-oriented" OR "aspect orientation"

Fonte: Do autor (2020).

Como dito anteriormente, este trabalho baseia-se em um MSL anterior (PARREIRA JÚNIOR; PENTEADO, 2014). Portanto, o conjunto de palavras-chave e *string* de busca foram mantidas. É importante ressaltar que, mesmo que o MSL anterior não tenha utilizado o *framework* PICO, os pesquisadores chegaram a uma *string* de busca coerente com a proposta desse *framework*. Para cada elemento do *framework* PICO, quando aplicável, são elencadas palavras-chave e respectivos sinônimos, combinando-as por utilizar os operadores lógicos OR e AND. Assim, a *string* de busca foi:

```
(( "approach" OR "approaches" OR "technique" OR "techniques" OR
  "tool" OR "tools" OR "method" OR "methods" ) AND ( "aspect-
  orientation" OR "aspect-oriented" ) AND ( "requirements
  engineering" ))
```

Os repositórios de artigos científicos utilizados neste MSL foram os mesmos do MSL anterior (*ACM Digital Library*, *IEEE Xplorer* e *Scopus*). O retorno da pesquisa nesses repositórios foi armazenado em bases de dados. A busca nesses repositórios considerou documentos com conteúdo sobre EROA. Os **critérios de exclusão** utilizados na seleção dos estudos primários (artigos) desse MSL são apresentados na Tabela 3.16. Como **critério de inclusão**, foram considerados estudos primários recuperados a partir da *string* de busca não rejeitados por um ou mais critérios de exclusão.

¹⁹ “Paradigma” representa a maneira como os interesses são identificados e classificados pela abordagem para EROA. Um exemplo de paradigma é PLN (Processamento de Linguagem Natural), que consiste no uso de recursos de processamento linguístico para identificação de potenciais interesses.

Tabela 3.16 - Critérios de Exclusão.

Sigla	Critério
E1	O estudo está escrito em outro idioma, que não seja o inglês.
E2	A versão completa do estudo não está disponível na web.
E3	O estudo não trata do uso, adaptação e/ou criação de abordagens para EROA.
E4	O estudo é uma versão duplicada de outro estudo.
E5	O estudo é uma versão antiga de outro estudo.
E6	O estudo possui até 2 páginas.
E7	O estudo não é um estudo primário, mas um prefácio de anais de evento, um capítulo de livro, uma tese de doutorado, entre outros.

Fonte: Do autor (2020).

3.4.2 Execução do MSL

Nessa fase, ocorreu investigação nos três repositórios de artigos científicos, em busca de estudos relevantes para responder às questões de pesquisa. Para isso, a *string* de busca foi utilizada nesses repositórios, considerando a especificidade da máquina de busca (Tabela 3.17). Na *ACM Digital Library*, a pesquisa foi realizada utilizando o modo *Advanced Search* com filtro “*The ACM Guide to Computing Literature*”, “*Publication Date: (01/01/2011 TO 12/31/2020)*”. Na *IEEE Xplore*, a pesquisa foi realizada no modo *Command Search* e foi utilizado o filtro “*Year: 2011 - 2020*”. Na *Scopus*, foi utilizado o modo *Advanced Search*, com os filtros “*Language: English*” e “*Pubyear: 2011 - 2020*”. A pesquisa nos repositórios foi realizada no mês de outubro de 2020.

Foram encontrados 1.517 documentos²⁰ (Tabela 3.18), dos quais 285 documentos (18,8%) são da *ACM Digital Library*, 14 documentos (0,9%) são da *IEEE Xplore* e 1.218 documentos (80,3%) são da *Scopus*. As referências com extensão de arquivo BibTeX foram extraídas dos repositórios e armazenadas no software JabRef. Para obter os artigos resultantes do MSL, foram realizadas 4 etapas:

- a) **Etapa 1.** Os documentos resultantes da utilização da *string* de busca nos repositórios foram armazenados em arquivos separados para serem analisados individualmente e excluídos os “não artigos” (por exemplo, foram removidos “*Inproceedings*”, “*Phdthesis*”, “*Incollection*”, “*Book*”, “*Inbook*”, “*TechReport*” e “*Conference*” para o atributo “*entryType*”). Isso foi feito para atender ao critério de exclusão E7 (Tabela 3.16). Dos 1.517 documentos, 819 documentos (54,0%) foram aceitos (A) e 698 documentos (46,0%) foram rejeitados (R). Dos artigos²¹ aceitos, 190 artigos (23,2%) são da *ACM Digital Library*, nenhum artigo (0%) é da *IEEE Xplore* e 629 artigos (76,8%) são da *Scopus*. Os metadados

²⁰ O resultado é considerado **documento**, pois nem todos são artigos, podendo ser *Inproceedings*, *Phdthesis*, *Incollection*, *Book*, *Inbook* e *Conference*, por exemplo.

²¹ Nesse momento, há apenas artigos científicos.

dos estudos foram rejeitados, pois, em análise com os pesquisadores, esses documentos ainda não foram avaliados ou não estão em formato de artigos. Os artigos oriundos do MSL anterior não foram incluídos, pois pressupõe-se que tenham passado por filtros no MSL anterior. Como esse MSL é uma extensão, a inserção dos artigos do MSL anterior foi realizada na Etapa 4;

Tabela 3.17 - Strings de Busca e Filtros Utilizados nos Repositórios.

Repositórios	String de Busca
ACM Digital Library	[[All: "approach"] OR [All: "approaches"] OR [All: "technique"] OR [All: "techniques"] OR [All: "tool"] OR [All: "tools"] OR [All: "method"] OR [All: "methods"]] AND [[All: "aspect-orientation"] OR [All: "aspect-oriented"]] AND [All: "requirements engineering"] AND [Publication Date: (01/01/2011 TO 12/31/2020)]
IEEE Xplore	((("approach" OR "approaches" OR "technique" OR "techniques" OR "tool" OR "tools" OR "method" OR "methods") AND ("aspect-orientation" OR "aspect-oriented") AND ("requirements engineering"))) Filters Applied: 2011 - 2020
Scopus	((("approach" OR "approaches" OR "technique" OR "techniques" OR "tool" OR "tools" OR "method" OR "methods") AND ("aspect-orientation" OR "aspect-oriented") AND ("requirements engineering")) AND (LIMIT-TO(PUBYEAR, 2020) OR LIMIT-TO(PUBYEAR, 2019) OR LIMIT-TO(PUBYEAR, 2018) OR LIMIT-TO(PUBYEAR, 2017) OR LIMIT-TO(PUBYEAR, 2016) OR LIMIT-TO(PUBYEAR, 2015) OR LIMIT-TO(PUBYEAR, 2014) OR LIMIT-TO(PUBYEAR, 2013) OR LIMIT-TO(PUBYEAR, 2012) OR LIMIT-TO(PUBYEAR, 2011)) AND (LIMIT-TO(LANGUAGE, "English"))

Fonte: Do autor (2020).

- b) **Etapa 2.** Os 819 artigos resultantes da etapa anterior foram agrupados em um único arquivo no software JabRef. Posteriormente, os artigos duplicados foram removidos, atendendo ao critério de exclusão E4 (Tabela 3.16), que resultou em 720 artigos (87,9%) aceitos e 99 artigos (12,1%) rejeitados. Dos artigos aceitos, 182 artigos (25,3%) são da *ACM Digital Library* e 538 artigos (74,7%) são da *Scopus*;
- c) **Etapa 3.** O título, o resumo e as palavras-chave dos 720 artigos resultantes da etapa anterior foram lidos e, quando eles não se encaixavam na temática da pesquisa, os artigos foram rejeitados. Além disso, os critérios de exclusão foram reaplicados (Tabela 3.16). Ao final, 58 artigos (8,1%) foram aceitos e 662 artigos (91,9%) foram rejeitados. Houve maior incidência de estudos rejeitados, pois eles não apresentavam quaisquer informações sobre abordagens para EROA, seguindo o critério de exclusão E3 (Tabela 3.16). Dos artigos aceitos, 16 artigos (27,6%) são da *ACM Digital Library* e 42 artigos (72,4%) são da *Scopus*;
- d) **Etapa 4.** Os 58 artigos selecionados na etapa anterior foram lidos por completo e os critérios de exclusão foram novamente aplicados (Tabela 3.16), resultando em 11 artigos (19,0%) aceitos e 47 artigos (81,0%) rejeitados. Dos artigos aceitos, 2 artigos (18,1%) são da *ACM Digital Library* e 9 artigos (81,9%) são da *Scopus*. Com a inserção dos 47 artigos advindos do MSL anterior, que juntamente com os procedimentos adotados pelos três pesquisadores

no início da coleta das informações, realizaram o seguinte procedimento: o Pesquisador A, o Pesquisador B e o Pesquisador C avaliaram os artigos presentes no MSL realizado em 2012 sobre EROA (PARREIRA JÚNIOR; PENTEADO, 2014) e no MSL estendido para removerem os artigos duplicados no MSL estendido. Com isso, os artigos que pertenciam aos MSL (artigos duplicados) foram removidos. Ao todo, foram analisados para responder às questões de pesquisa 58 artigos.

Tabela 3.18 - Resumo do Resultado do MSL.

Repositórios de Artigos Científicos	Busca Inicial	Etapa 1		Etapa 2		Etapa 3		Etapa 4	
		A	R	A	R	A	R	A	R
ACM Digital Library	285	190	95	182	8	16	166	2	14
IEEE Xplorer	14	0	14	0	0	0	0	0	0
Scopus	1.218	629	589	538	91	42	496	9	33
MSL anterior	0	0	0	0	0	0	0	47	0
Total	1.517	819	698	720	99	58	662	58	47

Fonte: Do autor (2020).

3.4.3 Análise dos Resultados

Nessa fase, foi realizada a compilação das informações presentes nos artigos resultantes da fase Execução. A referência completa aos 11 artigos (estudos primários) resultantes do MSL atual (P1 a P11) e a referência completa aos 47 artigos analisados no MSL anterior (P12 a P58) são apresentadas no APÊNDICE B.

Em resposta à questão de pesquisa QP1

QP1: Quais são as abordagens (técnicas, ferramentas e métodos) para EROA disponíveis na literatura e quais atividades elas contemplam?

foram identificadas 45 abordagens para EROA (Tabela 3.19). Nessa tabela, são apresentados o identificador único para a abordagem, o nome da abordagem e a referência dos artigos que as tratam. Ressalta-se que, nos casos em que não há um nome específico para a abordagem, é apresentado o título do artigo no qual essa abordagem foi apresentada. Além disso, são apresentadas informações sobre as principais atividades da EROA (PARREIRA JÚNIOR, 2015; SAMPAIO *et al.*, 2007):

- a) **Identificação e Classificação de Interesses (ICI).** Consiste na descoberta dos interesses a partir dos requisitos, classificando-os em interesses base ou interesses;
- b) **Representação de Interesses (RI).** Trata-se de formas gráficas/textuais para representar os interesses encontrados, caracterizando como os interesses se entrelaçam com requisitos de outros interesses e em quais pontos isso ocorre;

- c) **Composição de Interesses (CI).** É o modo de especificar os relacionamentos entre interesses e interesses base, agrupando-os para atingir o objetivo do sistema de software;
- d) **Detecção e Análise de Conflitos (DAC).** Consiste na investigação das influências dos diferentes interesses entre si, com o intuito de identificar possíveis conflitos existentes.

Tabela 3.19 - Abordagens para EROA.

#	Abordagem para EROA	Referências	Atividades			
			ICI	RI	CI	DAC
A1	An Approach for Crosscutting Concern Identification at Requirements Level using NLP*	P12, P19, P26	X			
A2	ACE - Aspect Clustering Engine	P13	X			
A3	AWC - Aspect Weaving Connector	P14		X	X	
A4	RDL - Requirements Description Language	P15, P47			X	
A5	ASSD - Aspects Specification for the Space Domain	P16	X	X		
A6	A Semi-Automatic Strategy to Identify Crosscutting Concerns in PL-AOVgraph Requirement Models*	P17	X			
A7	EA-Miner	P17, P28	X	X	X	X
A8	NFR/AUC	P18, P51	X	X	X	X
A9	DERAF	P20		X	X	
A10	An Evolutionary Model of Requirements Correctness with Early Aspects*	P21	X	X	X	
A11	AORE/ XML	P8, P22	X	X	X	
A12	Theme	P6, P23, P50	X	X	X	
A13	AspOrAs	P25, P53	X	X	X	X
A14	EA-Analyzer	P4, P27				X
A15	AORE with Arcade	P5, P29, P37	X	X	X	X
A16	PROBE	P30				X
A17	MAST - Modeling Aspectual Scenarios with Theme	P32	X	X	X	
A18	Integrating Problem Frames with Aspects*	P33	X	X	X	X
A19	Interaction Analysis in Aspect-Oriented Models*	P34				X
A20	Isolating and Relating Concerns in Requirements Using Latent Semantic Analysis*	P35	X			
A21	ADORA	P36		X		
A22	Multi-ComBO	P38			X	X
A23	Multi-Dimensional Separation of Concerns in Requirements Engineering*	P39	X	X	X	X
A24	CIG - Concern Interaction Graph	P40, P48				X
A25	On the Discovery of Candidate Aspects in Software Requirements*	P41	X			
A26	Promoting the Software Evolution in AOSD with Early Aspects: Architecture-Oriented Model-Based Pointcuts*	P42		X	X	
A27	RCT - Requirements Composition Table	P43	X	X	X	
A28	AOZCL	P45		X		
A29	Scenario Modeling with Aspects*	P46	X	X	X	
A30	VisualAORE	P49		X		
A31	Aspectual i* Model	P54	X	X		
A32	AO-ADL	P55		X		
A33	AoUCM-to-RAM	P56, P58		X	X	
A34	Using Tagging to Identify and Organize Concerns during Pre-Requirements Analysis*	P57	X			
A35	VGraph - From Goals to Aspects	P31		X	X	X
A36	AORA - Aspect-Oriented Requirements Analysis	P52	X	X	X	
A37	AoURN - Aspect-oriented User Requirements Notation	P44	X	X	X	
A38	RAM - Reusable Aspect Models	P24		X	X	
A39	Multi-Criteria Decision Making*	P1		X		X
A40	AORE4PF	P2	X	X		
A41	Trade-Off na forma do RCM	P11	X			X
A42	MultiCoS	P9	X	X		

(continua)

Tabela 3.20 - Abordagens para EROA (continuação).

#	Abordagem para EROA	Referências	Atividades			
			ICI	RI	CI	DAC
A43	GEA - Goal-Driven Early Aspect	P7	X			
A44	Dataflow Modeling*	P3	X	X	X	X
A45	ObasCId (Ontologically-Based Concern Identification)	P10	X	X		X
Quantidade de Abordagens para Atividades da EROA			28	31	23	16

Legenda: * - Título do Artigo em que a Abordagem não tem um nome específico; ICI - Identificação e Classificação de Interesses; RI - Representação de Interesses; CI - Composição de Interesses; DAC - Detecção e Análise de Conflitos

Fonte: Do autor (2020).

Analisando os artigos, muitas abordagens incluem as atividades ICI, RI e CI e faltam abordagens para a atividade DAC. Outros pontos interessantes são (i) apenas 15,6% das abordagens analisadas (A7, A8, A13, A15, A18, A23 e A44) incluir todas as atividades relacionadas à EROA, (ii) 22,2% das abordagens incluem, pelo menos, três atividades para EROA e (iii) 62,2% das abordagens incluem uma ou duas atividades para EROA. Isso indica que a realização de estudos em uma atividade específica da EROA ou um pequeno subconjunto de atividades pode ser uma estratégia interessante de pesquisa.

Pode-se notar que, em 11 abordagens analisadas (A1, A4, A7, A8, A11, A12, A13, A14, A15, A24 e A33), houve maior quantidade de artigos em que os autores optaram por realizar estudos de caso e/ou experimentos. Foram utilizadas abordagens para EROA para apresentar novas descobertas ou provar que a abordagem proposta é eficiente e eficaz para EROA. Essa incidência deve-se ao fato das abordagens serem as principais a serem utilizadas na EROA.

Conclusão. Muitas abordagens estão relacionadas à atividade de identificação e classificação de interesses. Artigos apontam essa atividade como importante na EROA e a maioria dos Engenheiros de Software experientes têm a *expertise* na identificação dos interesses sem a utilização de alguma abordagem. Entre as abordagens encontradas, algumas apresentaram algum tipo de apoio computacional para automatização de suas atividades. Além disso, foi constatado que, na maioria dos artigos, os autores utilizaram uma abordagem para uma ou todas as atividades para EROA e há poucas abordagens que atendem a todas as atividades.

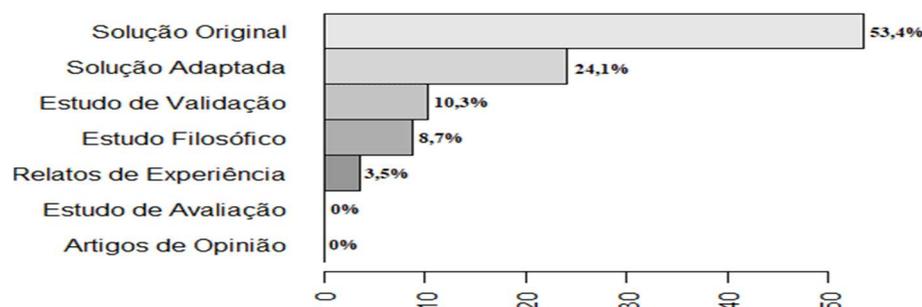
Em resposta à questão de pesquisa QP2

QP2: Quais são os tipos de estudos propostos para abordagens para EROA?

a classificação dos artigos resultantes da MSL é apresentada na Figura 3.2. Os tipos de estudos “Solução Original” e “Solução Adaptada” foram identificados em 77,5% dos artigos, sendo poucos os “Estudos de Validação”, “Estudos Filosóficos” e “Relatos de Experiência”, que somados, são cerca de 22,5% dos artigos. Nos tipos “Estudos de Avaliação” e “Artigos de

Opinião”, não houve artigo. Isso é relevante, pois muitas abordagens são utilizadas/adaptadas sem terem sido submetidas aos estudos de avaliação.

Figura 3.2 - Classificação dos Artigos.



Fonte: Do autor (2020).

Em 31 artigos (53,4%), a maioria das abordagens encontradas apresenta novas perspectivas às atividades para EROA (Identificação e Classificação de Interesses, Representação e Composição de Interesses e Detecção e Análise de Conflitos). Isso se deve pois os autores buscam melhorar as abordagens presentes na literatura para as atividades para EROA. Porém, essas abordagens são propostas sem conhecer a precisão, a cobertura e a *F-Measure* real das abordagens existentes nas atividades para EROA. Por exemplo, em P10, são apresentadas a abordagem *ObasCId (Ontologically-Based Concern Identification)* para aprimorar as atividades para EROA e uma ferramenta computacional que implementa a abordagem. Em P1, o autor apresenta a abordagem *Multi-Criteria Decision Making*, que consiste em critérios para o Engenheiro de Software identificar e classificar interesses.

Nesses artigos, os autores apresentaram estudos experimentais que avaliam as abordagens propostas. Em P12, P19 e P26, os autores propuseram uma solução adaptada para os estudos de caso apresentados por eles, pois, ao utilizarem uma abordagem conhecida na literatura, eles minimizaram algumas limitações da solução original. Em P6, P23 e P50, são apresentados estudos de validação das abordagens propostas pelos autores, realizando experimentos controlados com Engenheiros de Software.

Em P5, P29 e P37, são apresentados estudos filosóficos em que os autores apresentaram novas formas para as abordagens presentes na literatura, propondo aprimorá-las de maneira mais eficiente e eficaz. Em P1, P3 e P11, são apresentados relatos de experiência por expressarem depoimentos dos engenheiros de software mais experientes, mostrando como as abordagens podem ser utilizadas nas atividades para EROA.

Conclusão. Foi possível auxiliar na utilização dessas abordagens e suas ferramentas/apoios computacionais para realizar estudos experimentais, que validem a abordagem utilizada neste trabalho e a abordagem proposta na literatura, melhorando DRS elaborados com PRS e as atividades para EROA. Além disso, ao utilizar a abordagem *ObasCId* nos estudos experimentais para alcançar o objetivo deste trabalho, contribuiu-se para a melhoria da efetividade da identificação e classificação de interesses.

Em resposta à questão de pesquisa QP3

QP3: Quais são os “paradigmas” utilizados pelas abordagens para EROA para identificação e classificação de interesses?

foram identificados 5 paradigmas para identificação e classificação de interesses. Na Tabela 3.20, são apresentados o nome e a descrição desses paradigmas e as abordagens que os utilizam. Em P29, é descrita a atividade ICI (essa atividade é um gargalo para o processo de EROA), sendo importante conhecer os paradigmas utilizados nessa atividade para auxiliar profissionais e pesquisadores a obterem melhores estratégias para realização dessa atividade.

O paradigma mais usado em diferentes abordagens é “Análise manual do documento de requisitos pelos Engenheiros de Software com o auxílio de diretrizes”. Apesar de não ser adequado para aplicação em sistemas de software de grande porte, esse paradigma tem benefícios promissores. Um deles é a redução da dependência quanto à experiência dos usuários durante a aplicação da abordagem. Essa é uma indicação de que o paradigma pode trazer benefícios significativos para a atividade ICI. Outro ponto importante é em relação às poucas abordagens utilizarem, simultaneamente, mais do que dois paradigmas para a atividade ICI (por exemplo, A45). Esse fato pode ser um campo de pesquisa a ser explorado, pois o uso de técnicas combinadas pode levar a melhor precisão e cobertura às abordagens para EROA.

Tabela 3.20 - Paradigmas de Identificação e Classificação de Interesses.

Paradigma de Identificação e Classificação de Interesses	Descrição
Análise manual do documento de requisitos por Engenheiros de Software	Nesse tipo de paradigma, o Engenheiro de Software realiza uma inspeção manual no documento de requisitos, tentando descobrir os interesses de acordo com sua experiência com o software. Como limitações, tem-se: i) os resultados obtidos com este paradigma são fortemente dependentes da experiência de quem a aplica; e ii) este paradigma é propenso a erros, difícil de replicar sua aplicação e tem alto custo de execução quando documentos de requisitos de grandes sistemas de software são usados.
Abordagens que utilizam esse paradigma: A10, A13, A15, A17, A30.	

(continua)

Tabela 3.20 - Paradigmas de Identificação e Classificação de Interesses (continuação).

Paradigma de Identificação e Classificação de Interesses	Descrição
Análise manual do documento de requisitos por Engenheiros de Software com o auxílio de diretrizes	É semelhante ao paradigma de análise manual, mas difere-se pelo fato de os usuários deste paradigma possuírem orientações que podem auxiliá-los durante o processo de identificação de interesses. Tipos de diretrizes bastante comuns são os catálogos de requisitos não funcionais, como os propostos por (CHUNG; DO PRADO LEITE, 2009). Esse tipo de paradigma pode minimizar a dependência da experiência do usuário que a aplica. No entanto, ainda é dispendioso ser executada em um software grande. Além disso, é necessária certa experiência do usuário para entender e seguir as diretrizes.
Abordagens que utilizam este paradigma: A5, A7, A8, A11, A23, A27, A29, A31, A36 e A37, A39, A40, A41 e A45.	
Modelos probabilísticos e clusters	Esta abordagem é baseada em modelos estatísticos e em técnicas de agrupamento, como o uso de <i>tags</i> para encontrar candidatos a interesses. Como esse é um paradigma baseado em análise estatística, geralmente não traz bons resultados quando o DRS é pequeno, ou seja, quando a amostra é pequena.
Abordagens que utilizam esse paradigma: A2, A20, A25 e A34.	
NLP - <i>Natural Language Processing</i>	Baseia-se em técnicas de NLP para encontrar palavras-chave do texto do documento de requisitos relacionadas a algum tipo de interesse. De acordo com os estudos analisados, esse paradigma não era adequado para identificação de interesses implícitos, ou seja, interesses não explicitamente descritos no texto do documento de requisitos.
Abordagens que utilizam esse paradigma: A1, A7, A23 e A43, A45.	
Visualização de Software	Tal abordagem é baseada em técnicas de visualização para ajudar o usuário a identificar os interesses com o software. Um tipo de visualização bem conhecida denominada " <i>Action Views</i> " (SIOBHAN; ELISA, 2005). Uma limitação desse paradigma é que, para a construção das visualizações, geralmente, o usuário deve realizar uma inspeção manual do documento de requisitos, isto é, sofrer os mesmos problemas técnicos com base na análise manual do documento de requisitos citados acima.
Abordagens que utilizam esse paradigma: A6, A18, A12, A20, A25, A27, A34, A42 e A44.	

Fonte: Do autor (2020).

Conclusão. Foi constatado que os paradigmas encontrados são adaptações conhecidas de diferentes áreas, vinculadas ou não à Engenharia de Software. Especialmente a representação de interesses por meio de arquivos XML (*Extensible Markup Language*) e por meio dos tradicionais Diagramas UML. Por exemplo, a proposta *Reusable Aspect Models* (MUSSBACHER *et al.* 2011) é uma adaptação de Diagramas UML para o contexto de EROA, adaptando Diagramas de Casos de Uso, de Classes, de Sequência e de Estados. Além disso, nota-se que os paradigmas com maior quantidade de abordagens são aqueles que necessitam de maior *expertise* do ES, tal como a abordagem *Análise manual do documento de requisitos por Engenheiros de Software com o auxílio de diretrizes*, mesmo utilizando diretrizes que o auxiliem nas atividades para EROA.

Em resposta à questão de pesquisa QP4

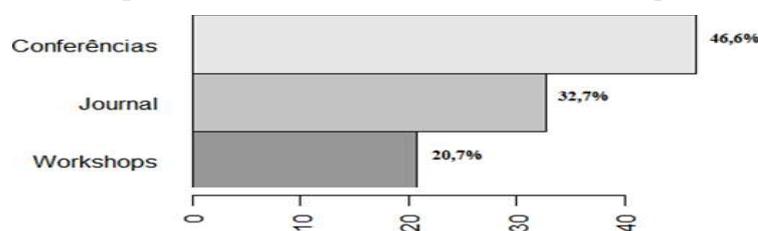
QP4 - Em quais eventos (e.g., conferências, workshops e periódicos) os artigos sobre PRS foram publicados?

notou-se que a maioria das publicações são em anais de congresso (Figura 3.3) das mais variadas linhas de pesquisa, por exemplo, "Projeto de Arquitetura" e "Arquitetura de Requisitos Orientado a Aspectos". Verificou-se ainda que, desde o ano de 2002, as publicações na área de

EROA vêm mostrando que o tema é de grande relevância para a área acadêmica, pois é possível encontrar publicações recentes sobre EROA.

Foi constatado que a maioria das publicações são realizadas no *Workshop de Engenharia de Requisitos Orientada a Aspectos e Design de Arquitetura*, por ser um evento dedicado à publicação de trabalhos relacionados à EROA. Outro evento com quantidade relevante de publicações relacionadas a EROA é *International Requirements Engineering*, uma reconhecida conferência em ER. Foram publicados 27 artigos (46,6%) em Conferências, com destaque para *International Requirements Engineering Conference*. Foram publicados 17 artigos (32,7%) em *Journals*, sendo que a maioria foi na área de Ciência da Computação e/ou Engenharia de Software, com destaque para o *Journal Science of Computer Programming*. Dos restantes, 12 artigos (20,7%) foram publicados em *Workshops*, com destaque para *ICSE Workshop on Aspect-Oriented Requirements Engineering and Architecture Design*.

Figura 3.3 - Tipos de Veículos de Publicação dos Artigos Analisados.



Fonte: Do autor (2020).

Conclusão. É possível verificar que os veículos de publicação aderentes ao tema de abordagens, técnicas e paradigmas para EROA são aceitos em diferentes locais. Na maioria dos artigos, é notável a diferença de publicações em conferências internacionais, mostrando que, ao finalizar as pesquisas deste trabalho e com os estudos experimentais, é possível submeter seus resultados nesses veículos de publicação.

3.5 Considerações Finais

Neste capítulo, foram discutidos resultados de dois MSL. O primeiro MLS foi sobre PRS na ER. Foram encontrados 104 PRS para padronizar requisitos de software, organizados em 12 catálogos de PRS. Parte dos artigos analisados utilizaram notações de casos de uso em PRS. Os PRS são utilizados em requisitos funcionais. Alguns artigos mostraram que PRS são utilizados para elaborar DRS de Sistemas de Informação. Há artigos que utilizaram alguma ferramenta/apoio computacional para escrever DRS utilizando PRS. A maioria dos artigos foi

estudo de validação. A forma de medir a qualidade de PRS, em alguns artigos, é por meio de avaliações com grupos de Engenheiros de Software. Há maior incidência de publicações no ano de 2017, com 5 publicações.

Com a descoberta de PRS, pode-se verificar que na literatura existem diferentes PRS para vários contextos em que eles podem ser utilizados. Para a utilização neste trabalho, foram selecionados PRS de CRUD (*Create, Read, Update, Delete*), de Segurança, de Desempenho, de Usabilidade e de Portabilidade.

A escolha desses PRS foi em decorrência da existência de diversas identificações e classificações de interesses relacionados aos tipos de requisitos contemplados pelos PRS escolhidos. Os PRS de Segurança foram descritos nos trabalhos P1, P8, P14 e P20 e possuem objetivos variados, tais como, proteger os dados transmitidos, a atividade de negócios, os serviços de negócios, os dados contra acessos não autorizados e os dados armazenados. Os PRS de Portabilidade são descritos nos trabalhos P9, P11 e P14 e estão inseridos no contexto onde é uma das principais preocupações quando as configurações afetam a correção de sistemas de software e os ambientes operacionais de destino são altamente configuráveis (ABRAN *et al.*, 2013). Os PRS de CRUD são descritos nos trabalhos P17 e P18 e possuem enfoque em Sistemas de Informação. Os PRS de Usabilidade são descritos nos trabalhos P1, P12 e P13 e possuem o enfoque na interface do usuário com o sistema de software, tendo em vista garantir que a interface seja fácil de usar. Os PRS de Desempenho são descritos nos trabalhos P3, P15 e P16 e são importantes aos sistemas de software. Para sistemas em tempo real, são mais que necessários, podendo ser visível aos usuários seu desempenho em determinadas tarefas que demandam mais processamento.

O segundo MSL foi sobre abordagens (técnicas, ferramentas e métodos) e paradigmas para EROA. Foram encontradas 45 abordagens para atividades da EROA a partir de requisitos de software, sendo que 31 abordagens apresentaram soluções originais para as atividades da EROA, ou seja, são novas abordagens para aperfeiçoar as atividades. Foram identificados 5 paradigmas para a atividade de identificação e classificação de interesses (“Análise manual do documento de requisitos por Engenheiros de Software”, “Análise manual do documento de requisitos por Engenheiros de Software com o auxílio de diretrizes”, “Modelos probabilísticos e *clusters*”, “NLP - *Natural Language Processing*” e “Visualização de Software”). A maioria dos artigos é publicada em conferências, com destaque para *International Requirements Engineering Conference*. A abordagem *ObasCld* (PARREIRA JÚNIOR; PENTEADO, 2018) foi utilizada neste trabalho, uma vez que (i) é uma das

abordagens mais recentes sobre EROA (conforme MSL apresentado neste capítulo), (ii) contém resultados experimentais e (iii) possui apoio computacional (ObasCId-Tool) para automatização de algumas atividades.

Os resultados dos MSL podem ser utilizados por pesquisadores, para exploração dos temas, e por profissionais, para identificar abordagens e PRS adequados às suas necessidades. Por exemplo, a partir dos resultados obtidos no MSL sobre EROA, é possível desenvolver um conjunto de critérios de comparação para as abordagens, baseado em características comuns e na variabilidade das abordagens analisadas. Os resultados podem servir como um guia aos usuários escolherem a abordagem que melhor atenda às suas necessidades e facilitar a conduta de pesquisa em EROA. Outra direção que pode ser seguida é criar uma abordagem ou desenvolver uma ferramenta/apoio computacional para combinar diferentes abordagens de forma a melhorar a cobertura e a precisão nas atividades para mostrar os interesses. Os resultados apresentam diferentes abordagens, cujo foco são paradigmas para EROA. No MSL sobre PRS, os resultados obtidos fornecem subsídios aos Engenheiros de Software a criar PRS ou utilizar os existentes na literatura, visto que o MSL mostrou que existem diversos padrões diferentes cenários em que os sistemas de software são desenvolvidos.

4 PADRÕES DE REQUISITOS DE SOFTWARE

4.1 Considerações Iniciais

Tendo como objetivo verificar se a utilização de PRS para escrever DRS contribuirá para a melhoria da efetividade da identificação e classificação de interesses em sistemas de software, são apresentados os PRS de CRUD, de Segurança, de Desempenho, de Usabilidade e de Portabilidade neste capítulo. Nesses PRS, são mantidas as nomenclaturas do catálogo de PRS apresentado no Capítulo 3 (Tabela 3.7).

Esses PRS foram utilizados, pois são os que têm maior incidência de utilização nos artigos encontrados referente ao MSL sobre PRS. Na escrita de requisitos funcionais, geralmente, o texto possui uma parte fixa e outra variável. A parte fixa representa a operação a ser realizada e a parte variável contém os detalhes específicos dessa operação, tais como, a entidade de dados atendida, os atributos necessários e as restrições específicas do negócio. Os PRS de sistemas de informação utilizados neste trabalho contemplam essa estrutura.

Este capítulo está organizado da seguinte forma. Na Seção 4.2, é apresentada a estrutura (*template*) dos PRS a ser utilizada. Na Seção 4.3, são apresentados os PRS de CRUD, de Segurança, de Desempenho, de Usabilidade e de Portabilidade e suas especificidades.

4.2 Estrutura do PRS

Em um dos trabalhos existentes na literatura (BARCELOS; PENTEADO, 2017), pode-se encontrar um *template* que contempla o PRS de CRUD. Em outros trabalhos (AYSOLMAZ *et al.*, 2018; MIRANDA *et al.*, 2017; ABRAN *et al.*, 2013; POST *et al.*, 2012; RENAULT *et al.*, 2009; JURISTO *et al.*, 2007; HAGGE; LAPPE, 2005), foram encontrados *templates* que contemplam informações similares. Tendo essas informações em vista, uma estrutura genérica para os PRS é apresentada na Tabela 4.1. Os elementos definidos nessa estrutura são:

- a) **Nome:** especifica o nome único para cada PRS;
- b) **Domínio:** especifica o contexto do PRS;
- c) **Propósito:** descreve o objetivo do PRS;
- d) **Problema:** descreve a situação a ser resolvida com o PRS;
- e) **Consequência:** descreve por qual razão utilizar o PRS;
- f) **Tipo:** estabelece o tipo de requisito a ser atendido pelo PRS;
- g) **Solução:** apresenta o *template* utilizado para a escrita do requisito padronizado;

- h) **PRS Relacionados:** complementa o entendimento do PRS e de seu relacionamento com outros PRS.

Tabela 4.1 - *Template* de Apresentação do PRS.

Elemento do PRS	Descrição
Nome	Especifica o nome do padrão que deve ser único e refletir a sua aplicabilidade.
Domínio	Corresponde ao domínio de aplicação do padrão.
Propósito	Descreve o objetivo da aplicação do padrão.
Problema	Descreve a situação em que o padrão pode ser aplicado.
Consequência	Descreve as consequências de utilizar o padrão.
Tipo	Especifica o tipo de requisito atendido pelo padrão (Funcional, Não-funcional ou Regra de Negócio).
Solução	Apresenta um <i>template</i> para a escrita das partes fixa e variável do requisito que o padrão deve representar. A parte fixa apresenta um texto padrão para a escrita do requisito. A notação "<...>" é usada para descrever a parte variável, denominada de parâmetro, e deve ser substituída pelos dados pertinentes ao requisito a ser elaborado.
PRS Relacionados	Especifica os padrões relacionados complementares ao padrão em questão. Esse elemento contribui para o Engenheiro de Requisitos na indicação de outros padrões que podem ser usados quando ocorrer o uso do padrão em questão.

Fonte: Do autor (2020).

4.3 Seleção dos PRS

Para atender aos diferentes domínios de sistemas de software, foram selecionados 5 PRS, sendo PRS de CRUD, contendo um PRS para cada uma das quatro operações básicas, as quais estão presentes na maioria dos sistemas de software em que realizam operações de manipulação da informação, e PRS de Segurança, de Desempenho, de Usabilidade e de Portabilidade que correspondem aos requisitos não-funcionais, possibilitando o reúso do PRS na íntegra em diferentes domínios, sofrendo o mínimo de alterações.

Os PRS selecionados direcionam os Engenheiros de Software na padronização dos requisitos funcionais e dos requisitos não-funcionais para a confecção de DRS. Profissionais com experiência em ER ou que possuem DRS de projetos finalizados podem utilizar os PRS apresentados neste trabalho e reusá-los em DRS de outros domínios de sistemas de software.

4.3.1 *Template* para PRS de CRUD

Os PRS de CRUD são utilizados para descrever manipulações básicas de dados, ou seja, criação (*Create*), leitura (*Read*), atualização (*Update*) e remoção (*Delete*) de dados. Estruturas iniciais (AHMAD *et al.*, 2018; BARCELOS; PENTEADO, 2017) para o PRS de CRUD foram encontradas ao realizar o MSL sobre PRS (Capítulo 3). Para a operação *Create*, o *template* proposto para **PRS de CRUD - Incluir Informação** é apresentado na Tabela 4.2.

Tabela 4.2 - *Template* para PRS de CRUD - Incluir Informação.

Elemento do PRS	Descrição
Nome	PRS de CRUD - Incluir Informação
Domínio	Sistemas de Informação
Propósito	Armazenar informações de uma entidade de dados.
Problema	Estabelecer os dados da entidade que devem ser armazenados.
Consequência	A escrita de um requisito que descreve a necessidade de um usuário de armazenar informação de uma entidade de dados.
Tipo	Funcional
Solução	<p>Template: Permitir a inclusão de informações de <entidade>, contendo os seguintes atributos: <atributos>, <condição>.</p> <p>Parâmetro: <entidade> nome da entidade que contém informações do domínio (1). <atributos> nome dos atributos da entidade que devem ser armazenados (1..*).* <condição> descrição da condição para incluir a informação (0..*).</p> <p>Exemplo de Parâmetros: <entidade>: cliente <atributos>: código, nome, endereço, telefone.... <condição>: deve concordar com os termos de uso da aplicação</p> <p>Exemplo da Descrição do Requisito: - Permitir a inclusão de informações de cliente, contendo os seguintes atributos: código, nome, endereço, bairro e cidade, deve concordar com os termos de uso da aplicação.</p>
PRS Relacionados	PRS de CRUD - Recuperar Informação PRS de CRUD - Alterar Informação PRS de CRUD - Excluir Informação

Fonte: Do autor (2020).

Ao apresentar o *template* para PRS de CRUD - Incluir Informação, pode-se notar aspectos importantes para sua utilização, tais como, seu propósito (incluir os dados), qual tipo de requisito atendido (requisitos funcionais), *template* que o Engenheiro de Requisitos utilizará para descrever os requisitos de inserção de dados e os PRS relacionados (PRS de CRUD - Recuperar Informação, PRS de CRUD - Alterar Informação e PRS de CRUD - Excluir Informação). O *template* proposto para as operações *Read*, *Update* e *Delete* é apresentado na Tabela 4.3 (PRS de CRUD - Recuperar Informação), na Tabela 4.4 (PRS de CRUD - Alterar Informação) e na Tabela 4.5 (PRS de CRUD - Excluir Informação).

Ao analisar diferentes DRS de sistemas de software, percebeu-se que as operações de CRUD, apesar de serem distintas, podem ser encontradas em um único requisito para sistemas simples. Às vezes, alguns requisitos podem ser especificados com o interesse Persistência, que consiste em armazenar (*Create*) dados/informações em um meio persistente para, posteriormente, serem atualizadas (*Update*), recuperadas (*Read*) ou removidas (*Delete*).

Tabela 4.3 - *Template* para PRS de CRUD - Recuperar Informação.

Elemento do PRS	Descrição
Nome	PRS de CRUD - Recuperar Informação
Domínio	Sistemas de Informação
Propósito	Recuperar informações de uma entidade de dados.
Problema	Recuperar uma informação.
Consequência	A escrita de um requisito que descreve a necessidade de um usuário de recuperar informação de uma entidade de dados por meio da descrição de parâmetros.
Tipo	Funcional
Solução	<p>Template: Permitir a recuperação de informação de <entidade> pelo(s) atributo(s) <atributos>, <condição>.</p> <p>Parâmetro: <entidade> nome da entidade que contém informações do domínio (1). <atributos> nome dos atributos da entidade que podem receber parâmetro (1..*). <condição> descrição da condição para a recuperação da informação (0..*).</p> <p>Exemplo de Parâmetros: <entidade>: cliente, produto, vendedor <atributos>: código, nome, cpf <condição>: sempre que o cliente for selecionado, durante o processamento da operação.</p> <p>Exemplo da Descrição do Requisito: - Permitir a recuperação de informação de cliente pelo(s) atributo(s) nome e cpf. - Permitir a recuperação de informação de reserva pelo(s) atributo(s) número da reserva, durante o processamento da operação de venda.</p>
PRS Relacionados	PRS de CRUD - Incluir Informação

Fonte: Do autor (2020).

Tabela 4.4 - *Template* para PRS de CRUD - Alterar Informação.

Elemento do PRS	Descrição
Nome	PRS de CRUD - Alterar Informação
Domínio	Sistemas de Informação
Propósito	Alterar informações de uma entidade de dados.
Problema	Estabelecer quais dados da entidade podem ser alterados.
Consequência	A escrita de um requisito que descreve a necessidade de um usuário de alterar determinados dados de uma entidade de dados.
Tipo	Funcional
Solução	<p>Template: Permitir a alteração de <entidade> no(s) seguinte(s) atributo(s) <atributos>, <condição>.</p> <p>Parâmetro: <entidade> nome da entidade que contém informações do domínio (1). <atributos> nome dos atributos da entidade que podem ser alterados (1..*). <condição> descrição da condição para a alteração da informação (0..*).</p> <p>Exemplo de Parâmetros: <atributos>: código, nome, endereço <entidade>: cliente, produto, vendedor <condição>: situação ABERTA e usuário logado.</p> <p>Exemplo da Descrição do Requisito: - Permitir a alteração de cliente no(s) seguinte(s) atributo(s) endereço, com o cliente ativo. - Permitir a alteração de cliente no(s) seguinte(s) atributo(s) endereço e bairro, somente por usuários logados.</p>
PRS Relacionados	PRS de CRUD - Incluir Informação

Fonte: Do autor (2020).

Tabela 4.5 - *Template* para PRS de CRUD - Excluir Informação.

Elemento do PRS	Descrição
Nome	PRS de CRUD - Excluir Informação
Domínio	Sistemas de Informação
Propósito	Excluir informações de uma entidade de dados.
Problema	Excluir a informação desejada.
Consequência	A escrita de um requisito que descreve a necessidade de um usuário de excluir informação de uma entidade de dados.
Tipo	Funcional
Solução	Template: Permitir a exclusão de um registro de <entidade>, <condição>.
	Parâmetro: <entidade> nome da entidade que contém informações do domínio (1). <condição> descrição da condição para a exclusão do registro (1..*).
	Exemplo de Parâmetros: <entidade>: cliente, produto, vendedor <condição>: desde que não possua nenhuma transação vinculado ao mesmo.
	Exemplo da Descrição do Requisito: - Permitir a exclusão de um registro de cliente. - Permitir a exclusão de um registro de cliente, desde que não possua nenhuma transação vinculada a ele.
PRS Relacionados	PRS de CRUD - Incluir Informação

Fonte: Do autor (2020).

4.3.2 *Template* para PRS de Segurança

Nas organizações em que há transações de dados, têm-se a preocupação com a segurança deles em sistemas de software (HAGGE; LAPPE, 2005). Diversos autores descrevem PRS de Segurança (Capítulo 3). Neste trabalho, é proposto um *template* para PRS de Segurança para autenticação de usuários. Contudo, o nome Segurança será utilizado, pois faz referência ao catálogo de PRS (Capítulo 3) relacionado ao interesse Segurança. Esse PRS é contemplado nos estudos experimentais, visto que é um dos mais utilizados em sistemas de informação para descrever um tipo de requisito não-funcional. As informações para preenchimento das descrições desse *template* foram sintetizadas de alguns trabalhos existentes na literatura (ZHANG *et al.*, 2020; ZAFAR *et al.*, 2014; POST *et al.*, 2012; NAVARRO *et al.*, 2006; HAGGE; LAPPE, 2005) e encontrados no MSL sobre PRS. Seguindo a estrutura apresentada, o *template* proposto para **PRS de Segurança** é apresentado na Tabela 4.6.

Tabela 4.6 - *Template* para PRS de Segurança.

Elemento do PRS	Descrição
Nome	PRS de Segurança
Domínio	Sistemas de Informação
Propósito	Garantir a segurança das informações em um sistema de software
Problema	Segurança dos dados
Consequência	A escrita de um requisito que descreve a necessidade de um usuário somente executar alguma funcionalidade se estiver autenticado, logado para recuperar informação de uma entidade de dados por meio da descrição de parâmetros.
Tipo	Não-Funcional

(continua)

Tabela 4.6 - *Template* para PRS de Segurança (continuação).

Elemento do PRS	Descrição
Solução	Template: Permitir acesso a <entidade> na funcionalidade de <funcionalidade>, apenas quando estiver autenticado no sistema <condição>.
	Parâmetro: <entidade> nome da entidade que contém informações do domínio (1) <funcionalidade> nome da funcionalidade em que o usuário terá permissão de acesso (1) <condição> descrição da condição para a segurança (autenticação) (0..*)
	Exemplo de Parâmetros: <entidade>: cliente, produto, vendedor <funcionalidade>: Cadastrar, Excluir, PerfilDeUsuario, Alteração <condição>: sempre que o cliente for selecionado, durante o processamento da operação.
	Exemplo da Descrição do Requisito: Permitir acesso a EnvioDeDados na funcionalidade de armazenar informações, apenas quando estiver autenticado no sistema ou quando existir autorização para realizar as tarefas no sistema e se o funcionário tem permissão para enviar dados pela internet.
PRS Relacionados	Não se aplica.

Fonte: Do autor (2020).

4.3.3 *Template* para PRS de Desempenho

A busca por desempenho está presente em quase todas as áreas da ciência. Um exemplo típico de seu uso, fora da área da Ciência da Computação, é encontrado na indústria automobilística, que busca melhor o desempenho dos veículos na velocidade máxima, na retomada de velocidade, na aceleração, na aceleração lateral, no espaço de frenagem, no nível de ruído, entre outros (SMITH; WILLIAMS, 2002). Neste trabalho, o PRS aborda somente o contexto de *tempo de resposta* (FEITELSON, 2015). Para contribuir com a identificação e a escrita de requisitos relacionados ao interesse Desempenho, foi utilizada o *template* proposto anteriormente para PRS, juntamente com as informações descritas nos trabalhos encontrados no Capítulo 3 (AYSOLMAZ *et al.*, 2018; MIRANDA *et al.*, 2017; VALLES-BARAJAS, 2007; NAVARRO *et al.*, 2006; DÍAZ *et al.*, 2004). O *template* proposto para **PRS de Desempenho** é apresentado na Tabela 4.7.

Tabela 4.7 - *Template* para PRS de Desempenho.

Elemento do PRS	Descrição
Nome	PRS de Desempenho
Domínio	Sistemas de Informação, Sistemas em tempo real, sistemas embarcados
Propósito	Garantir o desempenho de um sistema de software ao ser utilizado por muitos usuários
Problema	Sistema de software lento
Consequência	A escrita de um requisito que descreve a necessidade de um ou "n" usuários de estarem utilizando o sistema de software e que o tem de resposta de cada requisição seja o mesmo ou mais rápido.
Tipo	Não-Funcional

(continua)

Tabela 4.7 - *Template* para PRS de Desempenho (continuação).

Elemento do PRS	Descrição
Solução	Template: Permitir acesso à funcionalidade <funcionalidade> do sistema de software e o tempo de resposta não deve exceder <condição>.
	Parâmetro: <funcionalidade> nome da funcionalidade em que o sistema de software terá que apresentar maior desempenho (1). <condição> descrição da condição para a recuperação da informação (0..*).
	Exemplo de Parâmetros: <funcionalidade>: CadastroDeQueixas, CadastroDeJogadores, RankingDeJogadores <condição>: valor da duração para cada requisição dos usuários.
	Exemplo da Descrição do Requisito: - Permitir acesso à funcionalidade <i>CadastradosDeQueixas</i> do sistema de software e o tempo de resposta não deve exceder 3 segundos.
PRS Relacionados	Não se aplica.

Fonte: Do autor (2020).

4.3.4 *Template* para PRS de Usabilidade

A usabilidade é uma característica de qualidade de produto (ISO/IEC 25000, 2014), sendo a capacidade de uso e a aceitabilidade de um produto, tal qual um sistema de software para uma classe específica de usuários que executam tarefas específicas sob determinadas condições, por exemplo, reconhecimento de sua adequação, aprendizibilidade, operabilidade, proteção contra erros do usuário, estética da interface do usuário e acessibilidade. O *Template* proposto para **PRS de Usabilidade** (Tabela 4.8) é uma descrição de como o sistema de software deve ser de fácil uso aos usuários, tendo em vista que essa é uma das contribuições da usabilidade envolvendo outros fatores, por exemplo, eficácia e satisfação. O termo “Usabilidade” é utilizado para respeitar os catálogos encontrados no Capítulo 3 (VALLES-BARAJAS, 2007; JURISTO *et al.*, 2007; HAGGE; LAPPE, 2005).

4.3.5 *Template* para PRS de Portabilidade

A portabilidade é uma característica de qualidade de produto (ISO/IEC 25000, 2014), sendo definida como a capacidade de eficiência e eficácia com que um produto ou sistema de software pode ser transferido de um hardware, software, sistema operacional ou ambiente de uso para outro mantendo seu pleno funcionamento sem exigir grande retrabalho, mantendo, por exemplo, a adaptabilidade, a capacidade de ser instalado e a capacidade de ser substituído. O *Template* proposto para **PRS de Portabilidade** (Tabela 4.9) é uma descrição de como o sistema de software pode ser portátil a diferentes arquiteturas de software e/ou hardware. A utilização do termo “Portabilidade” foi mantida, respeitando os catálogos encontrados no Capítulo 3 (ABRAN *et al.*, 2013; RENAULT *et al.*, 2009; NAVARRO *et al.*, 2006).

Tabela 4.8 - *Template* para PRS de Usabilidade.

Elemento do PRS	Descrição
Nome	PRS de Usabilidade
Domínio	Sistemas de Informação, Sistemas em tempo real, sistemas embarcados
Propósito	Garantir que o sistema de software seja de fácil aprendizado e uso
Problema	Sistema de software de difícil utilização
Consequência	A escrita de um requisito que descreve a necessidade da usabilidade, para que o sistema de software tenha uma capacidade de uso que proporcione satisfação aos usuários
Tipo	Não-Funcional
Solução	Template: Permitir que o <ator> realize suas tarefas, utilizando a funcionalidade <funcionalidade> do sistema de software, com a condição <condição>.
	Parâmetro: <ator> nome do ator que contém informações do domínio (1). <funcionalidade> nome da funcionalidade em que o sistema de software terá que apresentar a capacidade de uso (1). <condição> descrição da condição para a melhorar a usabilidade (0..*).
	Exemplo de Parâmetros: <ator>: cliente, usuário, vendedor, administrador <funcionalidade>: armazenamento de dados de Fornecedores, Atualização, Relatórios Gerenciais <condição>: de no máximo 30 minutos de treinamento.
	Exemplo da Descrição do Requisito: Permitir que o Usuario realize suas tarefas, utilizando a funcionalidade de Cadastro de Fornecedores do sistema de software, com a condição de no máximo 30 minutos de treinamento.
PRS Relacionados	Não se aplica.

Fonte: Do autor (2020).

Tabela 4.9 - *Template* para PRS de Portabilidade.

Elemento do PRS	Descrição
Nome	PRS de Portabilidade
Domínio	Sistemas de Informação, Sistemas em tempo real, Software de Prateleira
Propósito	Garantir que os sistemas de software sejam capazes de serem executados em diferentes arquiteturas
Problema	Sistema de software que não executa em arquiteturas diferentes das quais foi desenvolvido
Consequência	A escrita de um requisito que descreve a necessidade de portabilidade, corresponde ao sistema de software seja portátil para diferentes, possibilitando a execução em diferentes versões de software e hardware
Tipo	Não-Funcional
Solução	Template: Permitir que o <software> seja executado em diferentes hardwares, softwares, sistemas operacionais ou ambientes de uso, com a condição <condição> mantendo a capacidade de eficiência e eficácia.
	Parâmetro: <software> nome do sistema de software que contém informações do domínio (1). <condição> descrição da condição para a portabilidade do sistema de software (0..*).
	Exemplo de Parâmetros: <software>: software de estoque, software nutricional, software de ponto eletrônico <condição>: de execução em dispositivos com sistema operacional Android ou iOS.
	Exemplo da Descrição do Requisito: - Permitir que o software nutricional seja executado em diferentes hardwares, softwares, sistemas operacionais ou ambientes de uso, com a condição de execução em dispositivos com sistema operacional Android ou iOS mantendo a capacidade de eficiência e eficácia.

(continua)

Tabela 4.9 - *Template* para PRS de Portabilidade (continuação).

Elemento do PRS	Descrição
PRS Relacionados	Não se aplica.

Fonte: Do autor (2020).

4.4 Considerações Finais

Os *templates* para PRS apresentados neste capítulo são a consolidação dos estudos realizados no MSL sobre PRS (Capítulo 3). Neste capítulo, foram propostos cinco PRS, sendo eles PRS de CRUD (contemplando o interesse Persistência), de Segurança, de Usabilidade, de Desempenho e de Portabilidade, que atendem aos respectivos interesses com o mesmo nome. Assim, um PRS atende os requisitos funcionais e quatro PRS atendem os requisitos não-funcionais. O objetivo é utilizá-los para escrever DRS e avaliar se contribuirão para a melhoria da efetividade, da identificação e da classificação de interesses em sistemas de software nas atividades para EROA.

5 UMA ABORDAGEM PARA A PADRONIZAÇÃO DE DOCUMENTOS DE REQUISITOS DE SOFTWARE

5.1 Considerações Iniciais

Um DRS deve proporcionar aos Engenheiros de Software entendimento mais amplo a respeito da funcionalidade do sistema de software, das restrições impostas a ele e do ambiente sobre o qual deve operar (SOMMERVILLE, 2019; SAMPAIO *et al.*, 2007). Neste capítulo, é apresentada uma abordagem para padronizar DRS, utilizando os PRS apresentados no Capítulo 4. Como saída dessa abordagem, o Engenheiro de Software terá à disposição um DRS padronizado, podendo utilizá-lo nas atividades da EROA. Com isso, espera-se melhoria nas atividades para EROA, em termos de resultados com maior precisão e maior cobertura.

Este capítulo está organizado da seguinte forma. Na Seção 5.2, é apresentada uma visão geral da abordagem proposta, destacando suas principais fases, artefatos consumidos e gerados/atualizados. Na Seção 5.3, são detalhadas as atividades de cada fase da abordagem proposta, ilustrando-as por meio de exemplos de uso.

5.2 Visão Geral da Abordagem

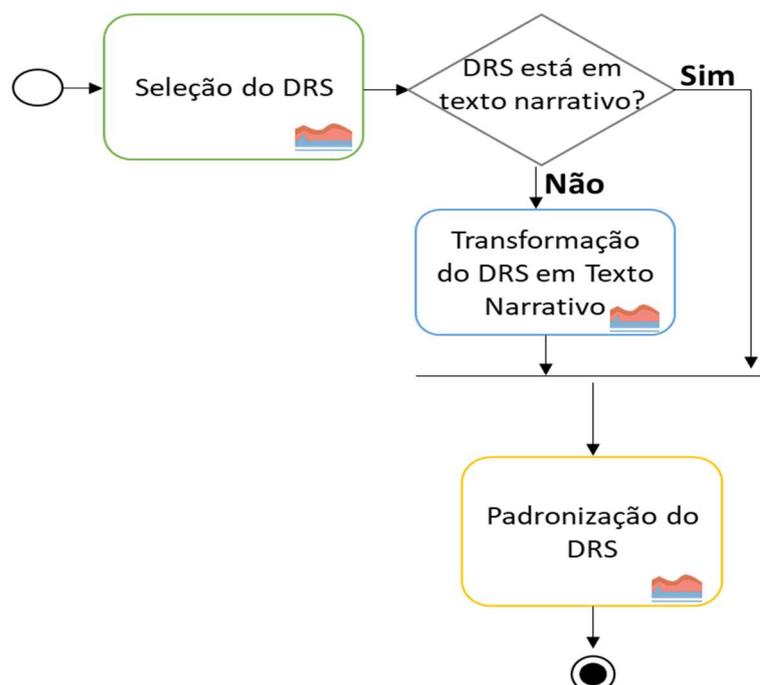
Para apresentar a abordagem proposta neste trabalho de forma gráfica, foi utilizado um conjunto de elementos de modelagem do meta-modelo para engenharia de processos de software SPEM (*Software Process Engineering Metamodel*) (SPEM, 2008). A abordagem é composta das seguintes fases (Figura 5.1):

- a) **Seleção do DRS.** Obter o DRS sobre o qual ocorrerá a identificação e classificação de interesses;
- b) **Transformação do DRS em Texto Narrativo.** Transformar o DRS em texto narrativo;
- c) **Padronização do DRS.** Utilizar os PRS nos requisitos presentes no DRS.

A lista contendo os artefatos consumidos e gerados/atualizados em cada fase da abordagem é apresentada na Tabela 5.1. Esses artefatos são evidenciados nas seções específicas de cada fase da abordagem. Na fase **Seleção do DRS**, é realizada a busca por documentos de requisitos em algum repositório externo ou da própria organização. O artefato de entrada obrigatório é o repositório de DRS e o artefato de saída é um DRS. Para iniciar a fase **Transformação do DRS em Texto Narrativo**, Engenheiros de Software devem utilizar o DRS escolhido na fase anterior, mas é necessário saber se o DRS está ou não escrito em texto narrativo. Por exemplo, DRS que possuem Casos de Uso ou cenários (SCHNEIDER;

WINTERS, 2001; DOUG; KENDALL, 1999) precisam passar por essa transformação. Caso o DRS esteja escrito em texto narrativo, inicia-se a fase **Padronização do DRS**. Caso contrário, é realizada a fase **Transformação do DRS em Texto Narrativo** em que o artefato de saída é o DRS em texto narrativo a ser utilizado pela abordagem proposta. Na fase **Padronização do DRS**, os artefatos de entrada são o DRS escrito em texto narrativo e os PRS escolhidos pelo Engenheiro de Software. Como artefato de saída, tem-se o DRS padronizado, utilizando os PRS escolhidos.

Figura 5.1 - Representação Gráfica da Abordagem.



Fonte: Do autor (2020).

Tabela 5.1 - Artefatos Consumidos e Gerados/Atualizados.

Fase	Artefatos Consumidos	Artefatos Gerados/Atualizados
Seleção do DRS	Repositórios de DRS	DRS
Transformação do DRS em Texto Narrativo	DRS	DRS escrito em texto narrativo
Padronização do DRS	DRS escrito em texto narrativo PRS	DRS padronizado

Fonte: Do autor (2020).

5.3 Descrição das Atividades da Abordagem Proposta

A descrição detalhada das fases da abordagem proposta é apresentada nesta seção, destacando suas atividades, os artefatos gerados/consumidos e o fluxo em que essas atividades ocorrem.

5.3.1 Fase: Seleção do DRS

Nessa fase, o objetivo é gerenciar (selecionar/elaborar/validar) o DRS a ser utilizado na abordagem proposta nesse trabalho que possui duas atividades. Em ambas, o artefato de saída é um DRS a ser utilizado nas próximas fases da abordagem:

- a) **Selecionar DRS.** Consiste em realizar uma seleção do DRS existente. Nesse momento, o Engenheiro de Software não precisa se preocupar com a forma estrutural do DRS, mas com o conteúdo a ser utilizado na abordagem;
- b) **Elaborar/Atualizar DRS.** Consiste na elaboração/atualização do DRS.

5.3.2 Fase: Transformação do DRS em Texto Narrativo

Nessa fase, o artefato de entrada é o DRS não escrito em texto narrativo. Assim, é realizada adequação em sua estrutura, utilizando o modelo de DRS escrito em texto narrativo (PARREIRA JÚNIOR; PENTEADO, 2016). Tal modelo é utilizado na maioria das abordagens para EROA, tais como, *MDSoc* (MOREIRA *et al.*, 2005), *EA-Miner* (CHITCHYAN *et al.*, 2006; SAMPAIO *et al.*, 2005), *EROA/XML* (SOEIRO *et al.*, 2006), *Theme* (SIOBHAN; ELISA, 2005; BANIASSAD; CLARKE, 2004), bem como em abordagens que fazem uso de ontologias no contexto da ER (VERMA; KASS, 2008). Esse modelo de DRS contém uma listagem de requisitos escritos em texto narrativo, cujas principais informações são o código do requisito, seu tipo (funcional e não-funcional) e sua descrição em texto narrativo (Tabela 5.2).

Tabela 5.2 - Modelo de DRS adotado na Abordagem Proposta.

Código	Tipo	Descrição do Requisito
RF-01	RF	Descrição do requisito funcional 1.
RNF-02	RNF	Descrição do requisito não-funcional 2.
...
<N>	<Tipo>	Descrição do requisito <Tipo><N>.

Legenda: Requisito Funcional (RF); Requisito Não-Funcional (RNF)

Fonte: Do autor (2020).

Na Tabela 5.3, estão ilustrados alguns trechos do DRS do sistema de software *Health Watcher* (HEALTH WATCHER, 2020). Esse DRS foi selecionado por ser de fácil entendimento e os autores terem *expertise* para avaliar o DRS resultante no final das fases da abordagem proposta. São apresentados dois requisitos não-funcionais e um requisito funcional. Um exemplo de trecho de DRS escrito em forma de casos de uso é apresentado na Tabela 5.4, representando o sistema de software *Health Watcher*. Após realizar a fase **Transformação DRS em Texto Narrativo** utilizando o exemplo anterior (requisito: *[RF01] Login* - Tabela 5.4), é apresentado o artefato gerado na Tabela 5.5.

Tabela 5.3 - Trecho do DRS do Sistema de Software Health Watcher (HEALTH WATCHER, 2020).

Código	Tipo	Descrição do Requisito
RF-01	RF	Tem como propósito permitir o acesso do funcionário a operações restritas no sistema Health Watcher.
...
RNF- 01	RNF	O sistema deve utilizar algum protocolo de segurança para envio de dados pela Internet.
RNF - 01.1	RNF	Para ter acesso aos recursos de registro das queixas, o usuário deve estar habilitado pelo controle de acesso ao sistema.

Legenda: Requisito Funcional (RF); Requisito Não-Funcional (RNF)

Fonte: Do autor (2020).

Tabela 5.4 - Trecho de DRS do Sistema de Software Health Watcher.

[RF01] Login
Este caso de uso tem como propósito permitir o acesso do funcionário a operações restritas no sistema Health-Watcher.
Prioridade
Essencial
Entradas e pré-condições
Nenhuma
Saídas e pós condições
Senha validada pelo sistema
Fluxo de eventos principal
1. O funcionário entra com o login e senha; 2. O sistema verifica a senha digitada.
Fluxo alternativo
No passo 2 caso a senha ou funcionário não seja válido, mostrar uma mensagem de erro.

Fonte: Do autor (2020).

Tabela 5.5 - Requisito do Sistema de Software Health Watcher após a Transformação para Texto Narrativo.

Código	Tipo	Descrição do Requisito
RF-01	RF	Tem como propósito permitir o acesso do funcionário a operações restritas no sistema de software Health Watcher, caso o usuário esteja logado no sistema (login e senha).

Legenda: Requisito Funcional (RF); Requisito Não-Funcional (RNF)

Fonte: Do autor (2020).

Após a transformação, é realizada a atividade **Validar o DRS em Texto Narrativo**, cujo objetivo é garantir que o DRS transformado não “fugiu” do contexto no qual foi escrito. Para isso, é necessário que essa atividade seja executada pelo Engenheiro de Software que tem *expertise* com o sistema de software em questão.

5.3.3 Fase: Padronização do DRS

Nessa fase, o objetivo é gerar um DRS padronizado, por exemplo, utilizando os PRS detalhados no Capítulo 4. Os artefatos de entrada são o DRS gerado na fase **Transformação do DRS em Texto Narrativo** (ou o DRS original, caso ele esteja escrito em texto narrativo) e os PRS propostos pela abordagem, por exemplo, o requisito escrito em texto narrativo:

RF-01: Tem como propósito permitir o acesso do funcionário a operações restritas no sistema de software Health Watcher, caso o usuário esteja logado no sistema (login e senha).

sendo que, com a utilização do *template* PRS de Segurança da abordagem proposta (Tabela 4.6), o requisito é padronizado como:

RF-01: Permitir acesso <entidade> na funcionalidade de apenas quando existir autorização para realizar as tarefas no sistema de software <condição>.

resultando em:

RF-01: Permitir acesso **Funcionario** nas funcionalidades de apenas quando existir autorização para realizar as tarefas no sistema de software **caso o usuário esteja logado no sistema (login e senha)**.

sendo a *tag* <entidade> a entidade pela qual o Engenheiro de Software quer permitir acesso e a *tag* <atributos> os atributos que a entidade deverá verificar para permitir o acesso.

5.4 Considerações Finais

Neste capítulo, foi apresentada uma descrição da abordagem proposta para a padronização de DRS para EROA, incluindo informações sobre a seleção do DRS, bem como sobre a transformação do DRS em texto narrativo e padronização do DRS. Além disso, exemplos foram utilizados para ilustrar e facilitar a compreensão das etapas dessa abordagem. O diferencial da abordagem proposta é o auxílio aos Engenheiros de Software durante as etapas de seleção, de transformação e de padronização do DRS para ser utilizado nas atividades para EROA.

6 OBASCID-TOOL+: APOIO COMPUTACIONAL À EROA UTILIZANDO PADRÕES DE REQUISITOS DE SOFTWARE

6.1 Considerações Iniciais

Um apoio computacional foi desenvolvido neste trabalho, denominado *ObasCId-Tool+*, cujo objetivo é nortear a construção de DRS, utilizando PRS e, conseqüentemente, identificar e classificar interesses a partir desses DRS.

Este capítulo está organizado da seguinte forma. Na Seção 6.2, é exibida a visão geral do apoio computacional desenvolvido - *ObasCId-Tool+*. Na Seção 6.3, é apresentado o módulo de gerenciamento de DRS, utilizando PRS.

6.2 Visão Geral do *ObasCId-Tool+*

O apoio computacional *ObasCId-Tool+* é uma extensão/evolução do *ObasCId-Tool* (PARREIRA JÚNIOR, 2015), composto por cinco módulos (PARREIRA JÚNIOR; PENTEADO, 2018):

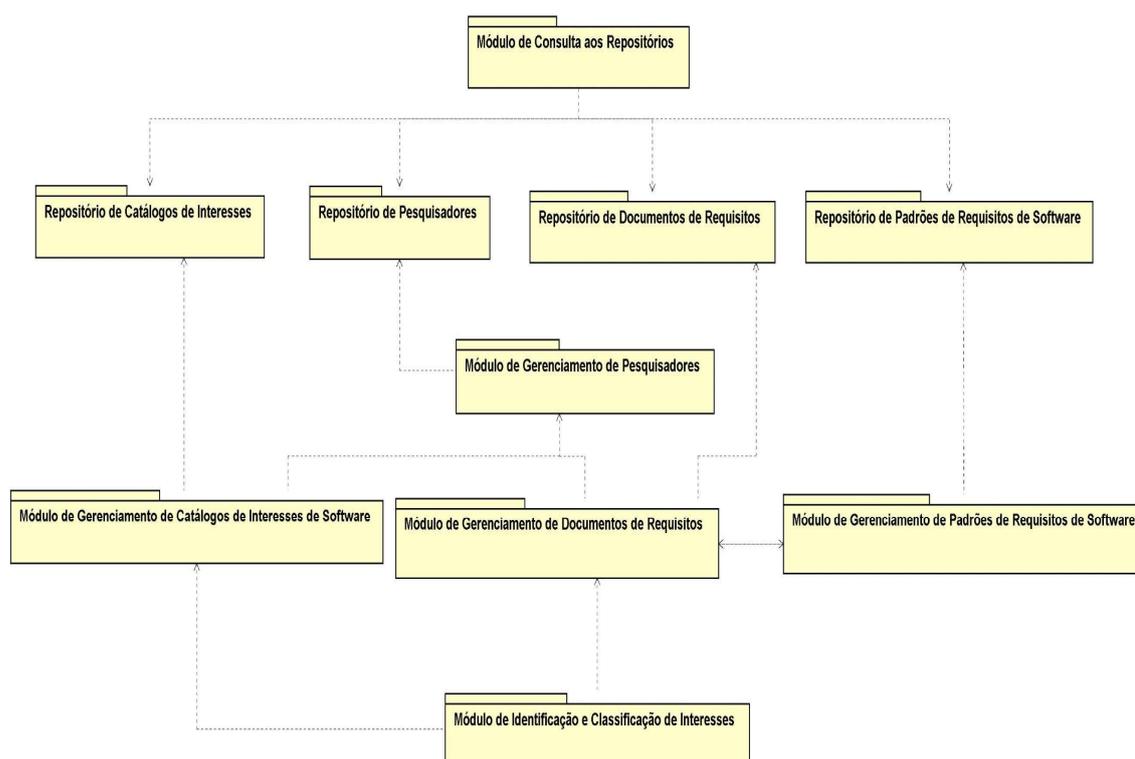
- a) **Consulta aos Repositórios**, utilizado por usuários interessados no apoio computacional *ObasCId-Tool*, esteja ou não cadastrado, para descoberta de documentos de requisitos e/ou catálogos de interesses já cadastrados na ferramenta computacional;
- b) **Gerenciamento de Pesquisadores**, cuja responsabilidade é incluir, atualizar e autenticar pesquisadores usuários do apoio computacional;
- c) **Gerenciamento de Catálogos de Interesses de Software**, cuja responsabilidade é elaborar e gerenciar catálogos de interesses de sistemas de software;
- d) **Gerenciamento de Documentos de Requisitos**, cuja responsabilidade é permitir pesquisadores incluir/atualizar DRS;
- e) **Identificação e Classificação de Interesses**, cuja responsabilidade é identificar e classificar interesses no DRS.

Na Figura 6.1, é apresentada a arquitetura da *ObasCId-Tool+* em um Diagrama de Pacotes da UML²², destacando seus principais componentes e dependências existentes entre eles. Nessa figura, as caixas retangulares especificam os módulos, os repositórios de dados e as setas pontilhadas indicam as dependências entre esses módulos, de forma que o módulo ligado

²² A *Unified Modeling Language* (UML) da OMG ajuda a especificar, visualizar e documentar modelos de sistemas de software, incluindo sua estrutura e design, de uma maneira que atenda a todos esses requisitos.

à origem da seta depende do módulo ligado ao destino da seta. *ObasCId-Tool+* é composto por seis módulos: i) Módulo de Consulta aos Repositórios; ii) Módulo de Gerenciamento de Pesquisadores; iii) Módulo de Gerenciamento de Catálogos de Interesses de Software; iv) Módulo de Gerenciamento de Documentos de Requisitos; v) Módulo de Identificação e Classificação de Interesses; e vi) Módulo de Gerenciamento de Padrões de Requisitos de Software.

Figura 6.1 - Arquitetura de *ObasCId-Tool+*.



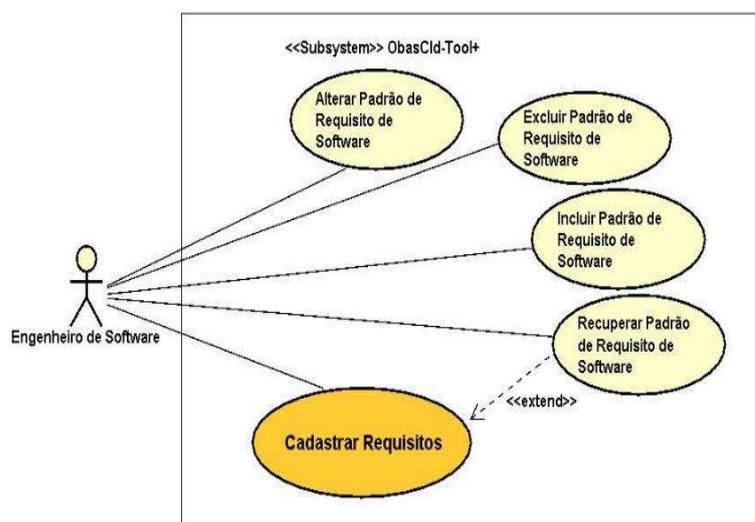
Fonte: Do autor (2020).

Neste trabalho, o foco central de *ObasCId-Tool+* é a extensão do **Módulo de Gerenciamento de Documentos de Requisitos**, originando no **Módulo de Gerenciamento de Padrões de Requisitos de Software**, conforme arquitetura da *ObasCId-Tool+* apresentado na Figura 6.1. Esse módulo foi desenvolvido para armazenar e utilizar PRS para padronizar os DRS cadastrados no **Módulo de Gerenciamento de Documentos de Requisitos**. Para utilizar os módulos iii), iv), v) e vi), um pesquisador deve estar cadastrado e autenticado. Deste modo, tais módulos dependem, direta ou indiretamente, do **Módulo de Gerenciamento de Pesquisadores**. Em seguida, o pesquisador pode criar e manter catálogos de interesses de software, *templates* para PRS e documentos de requisitos, utilizando o **Módulo de Gerenciamento de Catálogos de Interesses de Software**, o **Módulo de Gerenciamento de Documentos de Requisitos** e o **Módulo de Gerenciamento de Padrões de Requisitos de**

Software”. Na Figura 6.2, é apresentado o Diagrama de Caso de Uso referente ao **Módulo de Gerenciamento de Padrões de Requisitos de Software** que possui os seguintes casos de uso:

- a) **Incluir Padrão de Requisito de Software.** Permite a inclusão de informações *template* (Capítulo 4) dos PRS, tais como, nome, domínio, propósito, problema, consequência, tipo, solução e PRS Relacionados;
- b) **Recuperar Padrão de Requisito de Software.** Permite que o Engenheiro de Software possa utilizar o *template* dos PRS ao realizar ao cadastrar requisitos;
- c) **Alterar Padrão de Requisito de Software.** Permite que sejam alteradas as informações do *template* dos PRS;
- d) **Excluir Padrão de Requisitos de Software.** Permite que o *template* do PRS seja excluído.

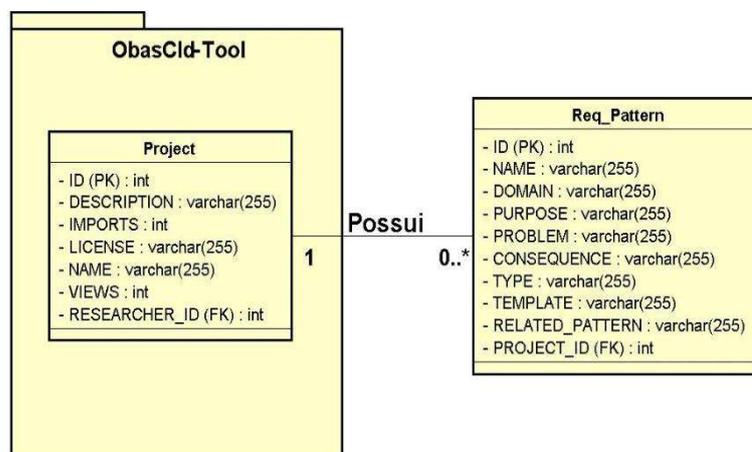
Figura 6.2 - Diagrama de Casos de Uso.



Fonte: Do autor (2020).

O caso de uso **Cadastrar Requisitos** está em laranja por estar presente na ObasCId-Tool, porém houve alteração para recuperar e utilizar os PRS dos requisitos funcionais e não-funcionais cadastrados. Com a implementação desses casos de uso, foi necessário construir uma tabela no banco de dados da ObasCId-Tool+ (Figura 6.3). Na tabela **Req_Pattern**, são armazenadas as informações inseridas pelo Engenheiro de Software ao incluir PRS. Essas informações são as mesmas cadastradas de acordo com o *template* apresentado no Capítulo 4. A *foreign key* “PROJECT_ID” faz o relacionamento com a tabela **Project** existente na ObasCId-Tool, sendo responsável pelo relacionamento do PRS cadastrado com o projeto em que foi utilizado no requisito funcional ou no requisito não-funcional.

Figura 6.3 - Tabela do Req_Pattern.



Fonte: Do autor (2020).

Para a desenvolvimento/evolução de da ObasCId-Tool+, foram utilizadas: i) a especificação *Java Server Faces* (JSF); ii) o *framework* para desenvolvimento de aplicações web *Mojarra*, que implementa a especificação JSF (utilizado para desenvolvimento do *backend* e do *frontend* da ObasCId-Tool+); e iii) o *framework CSS Bootstrap* para desenvolvimento de interfaces web responsivas. Ao iniciar a utilização da ObasCId-Tool+, é necessário o usuário ser cadastrado, fornecendo informações para os campos “Nome do Pesquisador”, “E-mail”, “Instituição”, “Cidade”, “Estado”, “País”, “Senha” e “Confirmar senha” (Figura 6.4). Posteriormente, o usuário procede com o *login*, utilizando o “E-mail” e a “Senha” cadastrados previamente (Figura 6.5).

Uma função importante é apresentada na tela “Meus Catálogos de Interesses” (Figura 6.6). Por meio dela, o usuário cadastra o catálogo a ser utilizado na descoberta dos interesses do sistema de software. Cabe ressaltar que isso é uma característica da abordagem ObasCId-Tool+, não dependendo se os requisitos estão ou não padronizados (PARREIRA JÚNIOR; PENTEADO, 2018). O usuário poderá criar o seu próprio catálogo de interesses ao selecionar o botão “Novo catálogo de interesses”. Nesse momento, é necessário inserir “Nome do catálogo de interesses” e “Tipo de licença do catálogo de interesses”. A última informação fornecida é para verificar se o usuário deseja que o catálogo seja “Privado” (somente quem cadastrou pode utilizar) ou “Público” (todos que tenham acesso à ObasCId-Tool+ podem utilizar). Além disso, o usuário pode inserir uma descrição do catálogo no campo “Descrição do catálogo de interesses” (Figura 6.7).

Figura 6.4 - Tela Incluir Pesquisador.

 Novo pesquisador

Nome do pesquisador *

Email *

Instituição

Cidade

Estado

País

Senha *

Confirmar senha *

* Campos de preenchimento obrigatório.

Fonte: Do autor (2020).

Figura 6.5 - Tela Login do Usuário.

 Login

Email *

Senha *

* Campos de preenchimento obrigatório.

Fonte: Do autor (2020).

Figura 6.6 - Tela de Manutenção de Catálogos de Interesses.

ObasCId-Tool  Meus Catálogos de Interesses  Meus Documentos de Requisitos  Repositórios Públicos  Michael ▾

Meus Catálogos de Interesses

-    Segurança
-    Logging
-    CRUD
-    Catálogo Health Watcher

Fonte: Do autor (2020).

Figura 6.7 - Tela de Inclusão de Catálogo de Interesses.

🔗 Novo catálogo de interesses

Nome do catálogo de interesses *

Descrição do catálogo de interesses

Tipo de licença do catálogo de interesses *

Privada ▼
 Pública
 Privada

* Campos de preenchimento obrigatório. ✓ Salvar Sair

Fonte: Do autor (2020).

Após a criação do catálogo, o usuário cadastra os interesses pertinentes a ele (Figura 6.8). Para cada interesse, o usuário informa “Nome do interesse”, “Tipo de interesse” (“Funcional” ou “Não-Funcional”) e “Prioridade do interesse” (“Não Especificada”, “Baixa”, “Média” ou “Alta”).

Figura 6.8 - Tela de Inclusão de Interesses.

🔗 Novo interesse

Nome do Interesse *

Descrição do Interesse

Tipo de Interesse *

Não funcional ▼
 Não especificada
 Baixa
 Média
 Alta

Prioridade do Interesse *

Alta ▼
 Não especificada
 Baixa
 Média
 Alta

Fonte: Do autor (2020).

Além disso, são cadastradas palavras-chave que possam realizar a sua identificação. Por exemplo, para o interesse Segurança, foram adicionadas as palavras-chave, “segurança”, “controle de acesso”, “autorização”, “autenticação”, “autenticado”, “logado” e “proteção” (Figura 6.9). Por fim, o usuário insere um DRS (Figura 6.10), fornecendo “Nome do documento de requisitos” e “Tipo de licença do documento de requisitos” (“Pública” ou “Privada”).

Até o momento, não foram utilizados os *templates* propostos. Os dados obrigatórios a serem informados para cada requisito são “Nome do requisito”, “Descrição do requisito” e “Tipo de requisito” (“Não-funcional” ou “Funcional”) (Figura 6.11).

Figura 6.9 - Tela de Manutenção de Palavras-Chave para Interesses.

Fonte: Do autor (2020).

Figura 6.10 - Tela de Inclusão de Documentos de Requisitos de Software (DRS).

Fonte: Do autor (2020).

Figura 6.11 - Tela de Inclusão de Requisitos.

Fonte: Do autor (2020).

6.3 Módulo Gerenciamento de Documentos de Requisitos

Conforme mencionado anteriormente, o módulo **Gerenciamento de Documentos de Requisitos** foi estendido, originando um botão na tela “Meus Documentos de Requisitos”, em que, ao ser selecionado, é aberta uma tela *modal* para o Engenheiro de Software cadastrar um novo PRS (Figura 6.12). Sendo assim, ao cadastrar um novo requisito, os PRS cadastrados anteriormente ou existentes na ferramenta serão listados na caixa de seleção “Utilizar padrão de requisito” (Figura 6.11). Caso o Engenheiro de Software escolha o padrão “Criação”, o seguinte texto será exibido no elemento gráfico que contém a descrição do requisito:

“Permitir a inclusão de informações de <entidade>, contendo os seguintes atributos: <atributos>, <condição>”

sendo a *tag* <entidade> a entidade a ser incluída na base de dados do sistema, a *tag* <atributos> os atributos da entidade e a *tag* <condição> alguma condição para incluir a informação. Por exemplo, ao incluir dados de usuário, é solicitado que ele autorize, via termo, o uso do sistema de software (Figura 6.13).

Figura 6.12 - Tela de Cadastro de novos PRS.

Fonte: Do autor (2020).

6.4 Módulo Classificação e Identificação de Interesses

Para identificar e classificar interesses na ObasCIId-Tool+, é necessário criar uma “Unidade de identificação”, informando “Nome da unidade de identificação” e vinculando um catálogo de interesses cadastrado pelo Engenheiro de Software (Figura 6.14). Essa unidade é responsável por executar a *engine* de identificação e classificação de interesses, gerando dados

a serem apresentados na forma de relatórios. ObasCId-Tool+ gera um relatório quanto ao resultado do processo de identificação e classificação dos interesses (Figura 6.15). Nesse relatório, pode-se notar que um alerta de possíveis erros e sugestões ao Engenheiro de Software são apresentados. No exemplo da Figura 6.15, é apresentado que dos cinco interesses candidatos a serem encontrados no DRS, ObasCId-Tool+ encontrou todos. Além disso, pode-se notar que esses interesses afetam 6 dos 25 requisitos cadastrados, o que equivale a 24% do total.

Figura 6.13 - Tela de Inclusão de Requisitos com PRS.

Fonte: Do autor (2020).

Figura 6.14 - Tela de Inclusão de Unidade de Identificação.

Fonte: Do autor (2020).

Figura 6.15 - Tela do Resultado: DRS sem PRS.

ObasCId-Tool ✦ Interesses identificados ★ Interesse principal ☰ Matrizes ▾ 🏠 Unidade de identificação Sem Padrão ▾

Meus Documentos de Requisitos / Unidade de Identificação / Resultados da Execução

Ops, algo de errado aconteceu!

- Ocorrência do Tipo I: não foi possível identificar o interesse principal do(s) seguinte(s) requisito(s): RNF01 – Usabilidade, RNF01.1 – Usabilidade (help), RNF02 – Confiabilidade, RNF02.1 – Confiabilidade (inacessível), RNF03 – Desempenho, RNF03.1 – Desempenho (tempo de resposta), RNF04 – Padrões, RNF05 – Distribuição, RNF06 – Manipulação de erros e exceções, RNF06.1 – Manipulação de erros e exceções (erros gerais), RNF06.2 – Manipulação de erros e exceções (outros erros), RNF07 – Interface com o usuário, RNF08 – Meio de armazenamento.

HealthWatcher

Quantidade de interesses selecionados: 5 (Criação, Segurança, Remoção, Leitura, Atualização)

Quantidade de interesses identificados: 5; 100,00% (Criação, Segurança, Remoção, Leitura, Atualização)

Quantidade de requisitos: 25

Quantidade de requisitos afetados: 6; 24,00%

🔍 Filtrar requisitos 🧼 Limpar filtro

Fonte: Do autor (2020).

6.5 Considerações Finais

ObasCId-Tool+ é destinado para identificar e classificar interesses com base em palavras-chave e no relacionamento de dependência existentes entre requisitos de software e esses interesses. O módulo **Gerenciamento de Documento de Requisitos** foi estendido, com base na abordagem apresentada nesse trabalho, juntamente com o PRS.

No Capítulo 7, são apresentados estudos experimentais realizados para avaliar a abordagem proposta e a extensão do apoio computacional ObasCId-Tool+, analisando a cobertura, a precisão e a *F-Measure*, quanto à sua efetividade e à sua eficiência no contexto da EROA.

7 ESTUDOS EXPERIMENTAIS

7.1 Considerações Iniciais

No Capítulo 5, foi apresentada uma abordagem para a padronização de DRS. Essa abordagem propõe um conjunto de atividades para guiar os pesquisadores na padronização dos requisitos utilizando PRS. No Capítulo 6, foi apresentada a *ObasCIId-Tool+* que automatiza a tarefa de cadastro e uso dos PRS em um sistema de software. Por ser baseada na *ObasCIId-Tool* (PARREIRA JÚNIOR, 2015), a *ObasCIId-Tool+* herda as funções de sua antecessora, por exemplo, a identificação e de classificação de interesses de software.

Cabe ressaltar, novamente, que a hipótese de pesquisa desta dissertação é **a utilização de PRS na escrita de DRS contribuir para a melhoria da efetividade de identificação e classificação de interesses em sistemas de software**. Com o intuito de verificar essa hipótese, alguns estudos experimentais foram realizados, de acordo com a abordagem apresentada por WOHLIN *et al.* (2012). Assim, cada experimento deve ser composto por três fases: i) **Planejamento**, na qual são especificados o contexto, as hipóteses, as variáveis, os participantes (se for o caso de um experimento com usuários), os instrumentos e o modelo do experimento; ii) **Execução**, na qual ocorre a preparação e a execução do experimento; e iii) **Análise dos Resultados**, em que os dados são organizados e analisados por meio de técnicas estatísticas.

Nesse trabalho, inicialmente, foi realizado um estudo experimental piloto (WOHLIN *et al.*, 2012). O foco principal de um experimento piloto não é encontrar uma resposta definitiva, mas criar um corpo de conhecimento sobre a aplicação da abordagem e/ou tecnologia (WOHLIN *et al.*, 2012, MAFRA *et al.*, 2006). Com isso, ganha-se conhecimento se o processo/abordagem em desenvolvimento é viável, se produz resultados consistentes e quais são suas limitações.

Este capítulo está organizado da seguinte forma. Na Seção 7.2, é apresentado o sistema de software *Health Watcher* e seus requisitos funcionais e não-funcionais utilizados no estudo experimental piloto. Na Seção 7.3, há uma descrição das medidas utilizadas nos estudos experimentais realizados nesta dissertação. Na Seção 7.4, é descrito o estudo experimental piloto. Na Seção 7.5, são apresentados os Estudos Experimentais e os sistemas de software utilizados. Na Seção 7.6, é apresentado o Estudo Experimental I, referente ao sistema de software *EasyHalf*. Na Seção 7.7, é apresentado o Estudo Experimental II, referente ao sistema de software *Fake News App*. Na Seção 7.8, é apresentado o Estudo Experimental III, referente ao sistema de software *ObasCIId-Tool*. Na Seção 7.9, é apresentado o Estudo

Experimental IV, referente ao sistema de software Sistema Gerenciador de Atividades Curriculares (SISGAC). Na Seção 7.10, é apresentado o Estudo Experimental V, referente ao sistema de software Veredas Sol e Lares. Para cada estudo experimental, são apresentados o planejamento, a execução, os resultados obtidos de acordo com o roteiro de experimentação para Engenharia de Software (WOHLIN *et al.*, 2012). Na Seção 7.11, é apresentada uma sumarização dos resultados dos estudos experimentais. Na Seção 7.12, são apresentados os testes de hipóteses abordados nos estudos experimentais. Na Seção 7.13, são apresentadas as ameaças à validade dos estudos experimentais realizados neste capítulo.

7.2 Sistema de Software Health Watcher

O sistema de software Health Watcher é um sistema de informação acessível pela Web, desenvolvido para melhorar a qualidade dos serviços oferecidos pela Secretaria de Saúde de uma prefeitura (MASSONI *et al.*, 2006). Esse sistema possibilita ao público registrar denúncias relacionadas à saúde, permitindo que a Secretaria de Saúde possa investigá-las e realizar as ações necessárias. Além disso, esse sistema é usado para notificar os cidadãos quanto às informações que sejam do interesse da população. O sistema de software Health Watcher tem sido utilizado como referência para o desenvolvimento de pesquisas em EROA por causa da heterogeneidade dos interesses encontrados nele (CHAVEZ *et al.*, 2009).

7.2.1 Requisitos Funcionais e Não-Funcionais do Sistema de Software Health Watcher

Por simplicidade, apenas três requisitos do sistema de software Health Watcher são apresentados nesta seção (esses requisitos foram obtidos a partir do trabalho de HEALTH WATCHER (2020)). O restante dos requisitos pode ser encontrado no APÊNDICE C.

[RF02] Especificar Queixa
<p>Este caso de uso tem como propósito o registro de queixas, que podem ser:</p> <p>e) Queixa Animal - DVA</p> <ul style="list-style-type: none"> - Casos de apreensão de animais; - Controle de vetores e animais sinantrópicos (roedores, escorpiões, morcegos, etc.); - Doenças associadas ao pernilongo (dengue, filariose); - Maus tratos com animais. <p>f) Queixa Alimentar - DVISA</p> <ul style="list-style-type: none"> - Casos de suspeita por ingestão de alimentos estragados. <p>g) Queixa diversa - DVISA</p> <ul style="list-style-type: none"> - Casos relacionados a diversos motivos, que não têm ligações com as queixas citadas anteriormente (restaurante suspeito quanto à higiene, fossas a céu aberto, carros pipas de procedimento suspeito, etc.);

[RF02] Especificar Queixa (continuação)

Os três tipos de queixa têm as seguintes informações em comum:

- h) Dados da queixa: descrição (obrigatório) e observações (opcional);
- i) Dados do reclamante: nome, rua, complemento, bairro, cidade, estado, CEP, número do telefone e e-mail. Todas essas informações são opcionais;
- j) Situação da queixa (obrigatório), que pode ser: ABERTA, SUSPensa ou FECHADA. No registro da queixa, a sua situação deve ser ABERTA;
- k) O sistema deve registrar a data de registro da queixa.

Além dessas, cada queixa tem suas informações específicas. São elas:

- l) Queixa Animal - DVA
 - Tipo de animal (obrigatório), quantidade de animais (obrigatório), data do incômodo (obrigatório);
 - Rua, complemento, bairro, cidade, estado, CEP e número do telefone do local de ocorrência. Todas essas informações são opcionais.
- m) Queixa Alimentar - DVISA
 - Nome da vítima (obrigatório);
 - Rua, complemento, bairro, cidade (ou a mais próxima), estado, CEP e número do telefone da vítima. Todas opcionais;
 - Quantidade de comensais (pessoas que comeram a comida), quantidade de doentes, número de pessoas internadas e número de óbitos. Todos obrigatórios;
 - Local em que os pacientes foram atendidos, refeição suspeita. Todas opcionais.
- n) Queixa diversa - DVISA
 - Idade (obrigatório), escolaridade (opcional) e ocupação (opcional);
 - Rua, complemento, bairro, cidade, estado, CEP e número do telefone do local mais próximo da ocorrência da queixa. Todas essas informações são opcionais.

Prioridade: Essencial.

Entradas e pré-condições:

- o) Nenhuma.

Saídas e pós-condições:

- p) A queixa deve estar registrada no sistema.

Fluxo de eventos principal

1. O cidadão informa o tipo de queixa;
2. O sistema registra o tipo, a data/ hora do atendimento;
3. O sistema apresenta a tela específica para cada tipo de queixa;
4. O cidadão informa os dados;
5. O sistema registra a queixa (com a situação ABERTA), retornando o número da ocorrência para que o cidadão anote e possa acompanhar o andamento.

[RF13] Cadastrar Novo Funcionário

Este caso de uso tem como propósito permitir o cadastramento de novos funcionários no sistema.

Prioridade: Essencial

Entradas e pré-condições:

- q) Funcionário logado no sistema.

Saídas e pós-condições:

- r) Novo funcionário cadastrado no sistema.

Fluxo de eventos principal

1. O funcionário entra com as seguintes informações do novo funcionário:
 - Nome;
 - Login.

[RF13] Cadastrar Novo Funcionário (cont.)
<ol style="list-style-type: none"> 2. Senha (com campo especial para confirmação). 3. O funcionário confirma a inserção. 4. O sistema cadastra os dados do novo funcionário. <p>Fluxo alternativo</p> <ol style="list-style-type: none"> 1. No passo 2, caso algum dos dados não tenha sido informado, mostrar uma mensagem de erro.

[RF14] Atualizar Funcionário
<p>Este caso de uso tem como propósito permitir a atualização de dados do funcionário no sistema.</p> <p>Prioridade: Essencial.</p> <p>Entradas e precondições:</p> <p>s) Funcionário logado no sistema.</p> <p>Saídas e pós-condições:</p> <p>t) Informações do funcionário atualizadas no sistema.</p> <p>Fluxo de eventos principal</p> <ol style="list-style-type: none"> 1. O funcionário ativa a opção atualizar funcionário; 2. O funcionário entra com as suas informações para atualização: <ul style="list-style-type: none"> - Nome; - Nova senha (com campo extra para confirmação); - Senha atual; 3. O funcionário confirma a atualização; 4. O sistema atualiza os dados do funcionário. <p>Fluxo alternativo</p> <ol style="list-style-type: none"> 1. No passo 3, caso algum dos dados não tenha sido informado o nome ou a senha atual, mostrar uma mensagem de erro.

Para os requisitos do Health Watcher serem utilizados no experimento piloto, foi utilizada a Abordagem para Padronização de DRS (Capítulo 5). Isso foi necessário para transformar os requisitos em texto narrativo, possibilitando sua utilização na ObasCId-Tool+. Na Tabela 7.1, são apresentados os três requisitos funcionais em texto narrativo (os demais requisitos, transformados em Texto Narrativo, podem ser verificados no APÊNDICE D). A partir da transformação da especificação dos requisitos em texto narrativo, eles foram padronizados, utilizando os *templates* para PRS de CRUD, de Segurança, de Desempenho, de Usabilidade e de Portabilidade. Na Tabela 7.2, é apresentada a padronização dos três requisitos usados como exemplo nesta seção e, no APÊNDICE E, é possível verificar todos os requisitos utilizados no experimento piloto com dos DRS padronizados.

7.3 Medidas Cobertura, Precisão e *F-Measure*

Para realização dos estudos experimentais (não apenas do experimento piloto), foram utilizadas as medidas Cobertura, Precisão e *F-Measure*, cuja fórmulas são apresentadas na Figura 7.1.

Tabela 7.1 - Requisitos Funcionais do Sistema de Software Health Watcher em Texto Narrativo.

Código	Descrição do Requisito
RF02 - Especificar Queixa	<p>Tem como propósito o registro de queixas. As queixas podem ser: (i) Queixa Animal: casos de apreensão de animais, controle de vetores e animais sinantrópicos (roedores, escorpiões, morcegos, etc.), doenças associadas ao pernilongo (dengue, filariose), maus tratos com animais; (ii) Queixa Alimentar: casos de suspeita por ingestão de alimentos estragados; e (iii) Queixa Diversa: casos relacionados a diversos motivos, que não têm ligações com as queixas citadas anteriormente (restaurante suspeito quanto à higiene, fossas a céu aberto, carros pipas de procedimento suspeito, etc.). Os três tipos de queixa têm as seguintes informações em comum: (i) Dados da queixa: descrição (obrigatório) e observações (opcional); (ii) Dados do reclamante: nome, rua, complemento, bairro, cidade, estado, CEP, número do telefone e e-mail. Todas estas informações são opcionais; e (iii) Situação da queixa (obrigatório), que pode ser: ABERTA, SUSPENSA ou FECHADA. No registro da queixa a sua situação deve ser ABERTA.</p> <p>O sistema deve registrar a data de registro da queixa. Além desta informação, cada queixa tem suas informações específicas. São elas: (i) Queixa Animal: tipo de animal (obrigatório), quantidade de animais (obrigatório) e data do incômodo (obrigatório); rua, complemento, bairro, cidade, estado, CEP e número do telefone do local de ocorrência (todas estas informações são opcionais); (ii) Queixa Alimentar: nome da vítima (obrigatório); rua, complemento, bairro, cidade, estado, CEP e número do telefone da vítima (todas opcionais); quantidade de comensais (pessoas que comeram a comida), quantidade de doentes, número de pessoas internadas e número de óbitos (todos obrigatórios); local em que os pacientes foram atendidos e refeição suspeita (todas opcionais); e (iii) Queixa Diversa: idade (obrigatório), escolaridade (opcional) e ocupação (opcional); rua, complemento, bairro, cidade, estado, CEP e número do telefone.</p>
RF13 - Cadastrar Novo Funcionário	<p>Tem como propósito realizar o cadastro de um novo funcionário no sistema. O funcionário deverá estar logado no sistema, assim entrar com as seguintes informações do novo funcionário devem ser inseridas, tais como Nome, Login e Senha (com campo especial para confirmação). Após, o funcionário confirma a inserção e o sistema cadastra os dados do novo funcionário. Caso algum dos dados não tenha sido informado, mostrar uma mensagem de erro. Ao final o novo funcionário cadastrado no sistema.</p>
RF14 - Atualizar Funcionário	<p>Tem como propósito permitir a atualização de dados do funcionário no sistema. O funcionário deverá estar logado no sistema e ativa a opção atualizar funcionário, entra com as informações para atualização, assim como Nome, Nova senha (com campo extra para confirmação) e Senha atual. O funcionário confirma a atualização e o sistema atualiza os dados do funcionário. Caso algum dos dados não tenha sido informado o nome ou a senha atual, mostrar uma mensagem de erro.</p>

Fonte: Do autor (2020).

No cálculo da cobertura, tem-se no numerador o **total de requisitos corretamente afetados por determinado interesse, de acordo com a abordagem utilizada**, e no denominador o **total de requisitos corretamente afetados por determinado interesse, de acordo com o oráculo**. O oráculo fornece um valor de referência, contendo os interesses existentes no DRS do sistema de software, uma vez que eles foram identificados e catalogados por especialistas em EROA. No caso do Health Watcher, um oráculo pode ser encontrado em HEALTH WATCHER (2020).

Por exemplo, considerando um DRS com cinco requisitos (R_1 , R_2 , R_3 , R_4 e R_5), supondo que a abordagem utilizada classificou corretamente quatro requisitos como afetados pelo interesse I_1 (R_1 , R_2 , R_4 e R_5) e que o oráculo afirma que há três requisitos afetados por esse interesse (R_2 , R_4 e R_5). Assim, tem-se que a cobertura dessa abordagem foi de 100%, pois todos os requisitos indicados pelo oráculo foram identificados pela abordagem. Porém, a abordagem classificou um requisito a mais (R_1), o qual não foi classificado pelo oráculo. Sendo assim, esse requisito se trata de um falso positivo, o que afeta negativamente a precisão da abordagem.

Tabela 7.2 - Requisitos Funcionais do Sistema de Software Health Watcher com PRS.

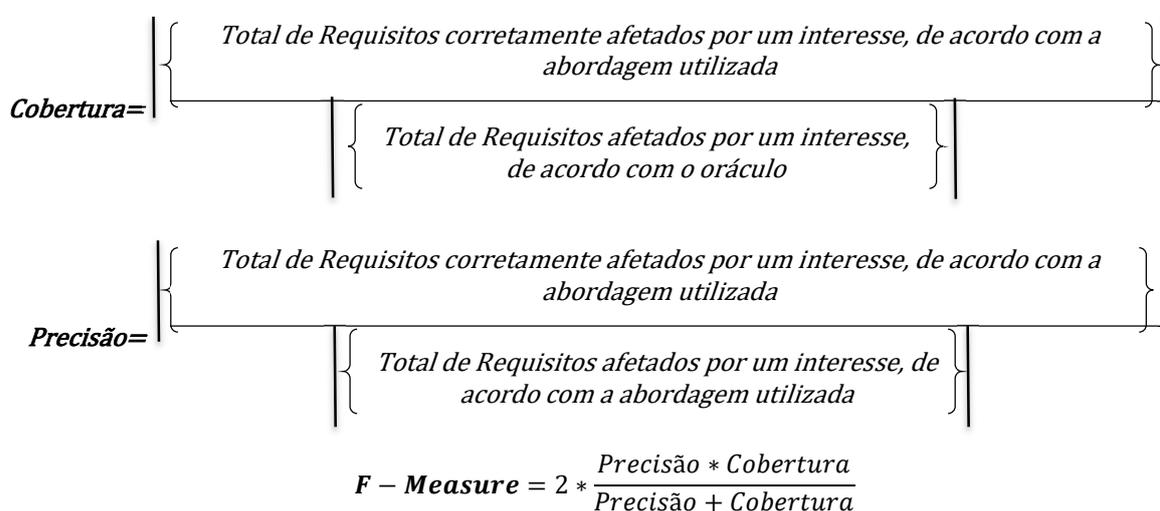
Código	Descrição do Requisito	PRS Utilizado
RF02 - Especificar Queixa	Permitir a inclusão de informações de Queixas, contendo os seguintes atributos: queixa animal, apreensão de animais, controle de vetores e animais sinantrópicos (roedores, escorpiões, morcegos, etc.), doenças associadas ao pernilongo (dengue, filariose), maus tratos com animais. De QueixaAlimentar contendo os seguintes atributos: casos de suspeita por ingestão de alimentos estragados e a inclusão de queixa diversa: casos relacionados a diversos motivos, motivos estes que não têm ligações com as queixas citadas anteriormente (restaurante suspeito quanto à higiene, fossas a céu aberto, carros pipas de procedimento suspeito, etc). De TiposDeQueixa, contendo os seguintes atributos: descrição (obrigatório) e observações (opcional), dados do reclamante, nome, rua, complemento, bairro, cidade, estado, CEP, número do telefone e e-mail e Situação da queixa (obrigatório), que pode ser: ABERTA, SUSPENSA ou FECHADA. De OutraInformacoesDeQueixas, contendo os seguintes atributos: data de registro da queixa, Queixa Animal: tipo de animal (obrigatório), quantidade de animais (obrigatório) e data do incômodo (obrigatório), rua, complemento, bairro, cidade, estado, CEP e número do telefone do local de ocorrência (todas estas informações são opcionais), Queixa Alimentar: nome da vítima (obrigatório); rua, complemento, bairro, cidade, estado, CEP e número do telefone da vítima (todas opcionais); quantidade de comensais (pessoas que comeram a comida), quantidade de doentes, número de pessoas internadas e número de óbitos (todos obrigatórios); local em que os pacientes foram atendidos e refeição suspeita (todas opcionais); Queixa Diversa: idade (obrigatório), escolaridade (opcional) e ocupação (opcional); rua, complemento, bairro, cidade, estado, CEP e número do telefone, a queixa deve ser cadastrada e com a situação ABERTA.	PRS de CRUD - Incluir Informação
RF13 - Cadastrar Novo Funcionário	Permitir a inclusão de informações de Funcionario, contendo os seguintes atributos: Nome, Login e Senha (com campo especial para confirmação), funcionário deve estar acessando o sistema.	PRS de CRUD - Incluir Informação
RF14 - Atualizar Funcionário	Permitir a alteração de Funcionario no(s) seguinte(s) atributo(s) Nome, Nova senha (com campo extra para confirmação) e Senha atual, caso algum dos dados não tenha sido informado, o nome ou a senha atual, mostrar uma mensagem de erro, e caso sejam informados, o funcionário confirma a atualização, sistema atualiza os dados do funcionário.	PRS de CRUD - Alterar Informação

Fonte: Do autor (2020).

No cálculo da precisão, tem-se no numerador o **total de requisitos corretamente afetados por determinado interesse, de acordo com a abordagem utilizada**, e no denominador o **total de requisitos afetados por determinado interesse, de acordo com a**

abordagem utilizada. Utilizando o exemplo anterior, tem-se que a precisão é 75%, pois dos quatro requisitos classificados pela abordagem (R₁, R₂, R₄ e R₅), três requisitos estavam corretos, segundo o oráculo (R₂, R₄ e R₅), e um requisito era um falso positivo.

Figura 7.1 - Cobertura, Precisão F-Measure.



Fonte: Do autor (2020).

A medida *F-Measure* é definida como a média harmônica ponderada da precisão e da cobertura. Essa medida se faz necessária quando é desejável buscar um equilíbrio entre Precisão e Cobertura. Utilizando os valores apresentados anteriormente, o valor resultante da medida *F-Measure* é 86%, caracterizando que 14% dos requisitos considerados na análise, em média, não foram identificados interesses. Seguindo o exemplo mencionado, tem-se:

$$F - Measure = 2 * \frac{0.75 * 1.0}{0.75 + 1.0} = 2 * 0,43 = 0,86$$

7.4 Estudo Experimental Piloto

O objetivo desse estudo experimental piloto foi prover dados que, quando analisados e comparados, possibilitassem aos pesquisadores identificar se a utilização dos *templates* propostos de PRS propiciou aumento na cobertura e na precisão do processo de identificação e classificação de interesses do sistema de software Health Watcher, a partir do seu DRS, subsidiando a execução dos demais estudos experimentais.

7.4.1 Planejamento do Estudo Experimental Piloto

O objetivo é apresentado na Tabela 7.3. O planejamento deste e dos outros estudos experimentais foi realizado de acordo com o modelo proposto por WOHLIN *et al.*, (2012), que envolve as fases:

- a) **Seleção do Contexto.** O estudo experimental piloto foi realizado utilizando a ObasCId-Tool+. O DRS utilizado neste experimento piloto foi do sistema de software Health Watcher. Esse sistema foi escolhido por apresentar um DRS propício à identificação e classificação de interesses, uma vez que vários interesses estão presentes nesse software, tais como, Persistência, Usabilidade, Segurança, Persistência e Desempenho. Além disso, os interesses do sistema de software Health Watcher foram identificados e catalogados por especialistas em EROA (HEALTH WATCHER, 2020), servindo como oráculo para verificação dos resultados obtidos;

Tabela 7.3 - Definição do objetivo do Estudo Experimental Piloto.

Analisar	o uso de PRS no processo identificação e classificação de interesses de software nos requisitos do sistema de software Health Watcher
com o propósito de	Avaliar
a partir do ponto de vista dos	engenheiros de software e analista de requisitos
com respeito à	cobertura, precisão e <i>F-Measure</i>

Fonte: Do autor (2020).

- b) **Formulação de hipóteses.** Seis hipóteses foram elaboradas para este e os demais experimentos realizados e apresentados nesta dissertação, sendo duas hipóteses relacionadas à cobertura, duas hipóteses relacionadas à precisão e duas hipóteses relacionadas à *F-Measure*. Na Tabela 7.4, são apresentadas essas hipóteses associadas a um identificador da hipótese, composto por H_0 para as hipóteses nulas e H_1 para as hipóteses alternativas e uma descrição das hipóteses;

Tabela 7.4 - Hipóteses para os Estudos Experimentais.

#	Hipóteses
$H_{0_cobertura}$	Não há diferença quanto à cobertura global média proporcionada pelo uso dos PRS com relação ao não uso de PRS no processo de identificação e classificação de interesses de software.
$H_{1_cobertura}$	Há diferença quanto à cobertura global média proporcionada pelo uso dos PRS com relação ao uso de PRS no processo de identificação e classificação de interesses de software.
$H_{0_precisão}$	Não há diferença quanto à precisão global média proporcionado pelo uso dos PRS com relação ao não uso de PRS no processo de identificação e classificação de interesses de software.
$H_{1_precisão}$	Há diferença quanto à precisão global média proporcionado uso dos PRS com relação ao uso de PRS no processo de identificação e classificação de interesse de software.
$H_{0_F-measure}$	Não há diferença quanto à <i>F-Measure</i> proporcionado pelo uso dos PRS com relação ao não uso de PRS no processo de identificação e classificação de interesses de software.
$H_{1_F-measure}$	Há diferença quanto à <i>F-Measure</i> proporcionado pelo uso dos PRS com relação ao uso de PRS no processo de identificação e classificação de interesses de software.

Fonte: Do autor (2020).

- c) **Seleção de variáveis.** Variáveis independentes são manipuladas e controladas durante o estudo. Nesse estudo, a variável independente consiste na utilização ou não dos *templates* para PRS no processo de identificação e classificação de interesses. As variáveis dependentes estão sob análise e cujas variações, com base nas mudanças feitas nas variáveis

independentes, devem ser observadas. Nesse experimento piloto e nos demais realizados neste trabalho, a cobertura, a precisão e a *F-Measure* são considerados como variáveis dependentes;

- d) **Projeto e execução do experimento piloto realizado.** Foram utilizados os requisitos funcionais e não-funcionais do sistema de software *Health Watcher* (Tabela 7.5) (três requisitos - RF02, RF13 e RF14 - foram descritos anteriormente). Nessa tabela, são apresentados o código referente ao requisito funcional (RF) ou requisito não-funcional (RNF), o nome do requisito no DRS do sistema de software *Health Watcher* e o interesse compreendido em cada requisito, de acordo com o oráculo. Tais requisitos foram cadastrados na *ObasCId-Tool+* de duas maneiras: i) somente o texto narrativo de cada requisito foi cadastrado, sem a usar os *templates* propostos de PRS; e ii) os mesmos requisitos foram reescritos usando PRS cadastrados. Como foi visto no Capítulo 6, a *ObasCId-Tool+* usa um catálogo para identificar e classificar interesses. Um desses catálogos foi o proposto por PARREIRA JÚNIOR; PENTEADO (2014), que possui oito interesses do tipo não funcional, relacionados a 32 palavras-chave (Tabela 7.6). Esse catálogo foi utilizado na execução dos estudos experimentais deste trabalho, pois contempla a maior parte dos interesses existentes nos sistemas de software analisados. Conforme mencionado anteriormente, o oráculo proposto em HEALTH WATCHER (2020) foi utilizado para avaliar o resultado da identificação e classificação dos interesses nesse experimento piloto. Na Tabela 7.7, é apresentada a relação dos interesses e os requisitos afetados por eles, de acordo com o oráculo utilizado. Além disso, apresenta-se o resultado da identificação e classificação de interesses, com e sem o uso dos *templates* para PRS. Cabe ressaltar que a abordagem para identificação e classificação de interesses e o catálogo de interesses são os mesmos para ambos os casos, diferenciando apenas se os requisitos estão ou não padronizados. Os requisitos sublinhados e em negrito na Tabela 7.7 representam falsos positivos, conforme será discutido mais à frente.

Tabela 7.5 - Requisitos Funcionais e Requisitos Não-Funcionais do Sistema de Software *Health Watcher* e seus Interesses.

Código	Nome	Interesses
RF01	Consultar Informações	Persistência
RF02	Especificar Queixa	Persistência
RF10	Login	Logging
RF11	Cadastrar Tabelas	Persistência
RF12	Atualizar Queixa	Persistência
RF13	Cadastrar Novo Funcionário	Persistência
RF14	Atualizar Funcionário	Persistência

(continua)

Tabela 7.5 - Requisitos Funcionais e Requisitos Não-Funcionais do Sistema de Software Health Watcher e seus Interesses (continuação).

Código	Nome	Interesses
RF15	Atualizar Unidade de Saúde	Persistência
RF16	Mudar Funcionário Logado	Alterar Funcionário Registrado
RNF01	Usabilidade	Usabilidade
RNF02	Confiabilidade	Disponibilidade
RNF03	Desempenho	Desempenho
RNF04	Segurança	Segurança
RNF05	Padrões	Normas
RNF06	Portabilidade	Portabilidade
RNF07	Distribuição	Distribuição
RNF08	Interface com o usuário	Interface de Usuário
RNF09	Meio de armazenamento	Armazenamento
RNF10	Manipulação de erros e exceções	Manipulação de Erros

Fonte: Do autor (2020).

Tabela 7.6 - Catálogo de Interesses Desenvolvido a partir de Dados Históricos do Sistema de Software Health Watcher.

Interesse Não-Funcional	Palavras-chave	Interesse Não-Funcional	Palavras-chave
Persistência	Recuperação	Segurança	segurança
	Cadastrar		controle de acesso
	Persistir		autorização
	Salvar		autenticação
	Incluir		autenticado
	Inclusão		logado
	Registrar	Padronização	Padronização
	Remover		Padrões
	Exclusão		Normas
	Alteração	Usabilidade	Help
	Atualização		Ajuda
	Consultar		Usabilidade
	Disponível		fácil utilização
Disponibilidade	disponibilidade	Desempenho	tempo de resposta
	simultaneamente		Portabilidade
Concorrência	Simultâneo	portável	

Fonte: Do autor (2020).

Tabela 7.7 - Execução do Estudo Experimental Piloto.

Interesse	Oráculo	Sem PRS	Com PRS
Persistência	RF01, RF02, RF11, RF12, RF13, RF14, RF15	RF02, RF11, RF12, RF14, RF15	RF01, RF02, RF11, RF12, RF13, RF14, RF15
Usabilidade	RNF01	RNF01	RNF01
Desempenho	RNF03	RNF03	RNF03
Segurança	RNF04, RF16	RNF04, RF16, RF11, RF12, RF13, RF14, RF15	RNF04, RF16
Portabilidade	RNF06	-----	RNF06

Fonte: Do autor (2020).

7.4.2 Análise dos Resultados do Estudo Experimental Piloto

Na Tabela 7.8, são apresentados os resultados obtidos de acordo com as medidas descritas na Seção 7.3. É possível notar que, com a utilização de PRS, quase todas medidas

alcançaram o valor máximo, isto é, 100%. Isso mostra que a hipótese de pesquisa é promissora e novos experimentos podem ser realizados para aumentar a confiabilidade dos resultados.

Tabela 7.8 - Resultados do Estudo Experimental Piloto para o Sistema de Software Health Watcher.

Interesse	DRS sem PRS			DRS com PRS		
	Cobertura	Precisão	<i>F-Measure</i>	Cobertura	Precisão	<i>F-Measure</i>
Persistência	71,4%	100%	76,92%	100%	100%	100%
Segurança	100%	28,6%	44,5%	100%	100%	100%
Usabilidade	100%	100%	100%	100%	100%	100%
Desempenho	100%	100%	100%	100%	100%	100%
Portabilidade	0%	0%	0%	100%	100%	100%

Fonte: Do autor (2020).

Cabe ressaltar alguns pontos a respeito dos resultados obtidos. A cobertura do interesse Persistência (CRUD), sem a utilização de *templates* para PRS, foi 71,4%. Isso ocorreu porque alguns requisitos não foram identificados corretamente por ObasCId-Tool+, (e.g., o requisito “RF01 - Consultar Informações”). Apesar de ser um requisito relacionado ao cadastro e recuperação de informações, em sua descrição não foram utilizadas palavras-chave comumente usadas em requisitos desse tipo. A partir do uso do *template* para PRS de CRUD, foi adequadamente identificado por ObasCId-Tool+. Como consequência, *F-Measure* para esse interesse sofreu impacto por não cobrir os requisitos totalmente, resultando em 76,92%.

Para a medida cobertura do interesse Segurança, aconteceu o inverso, sendo identificado e coberto em sua totalidade. Quanto à precisão, nota-se que houve diferença significativa se comparar com os interesses, por causa da presença de falsos positivos na identificação e classificação dos interesses. Esses falsos positivos podem ser vistos sublinhados e em negrito na Tabela 7.7. No caso do cenário sem a utilização de *templates* para PRS, os seguintes requisitos foram encontrados para o interesse Segurança, mas não estão relacionados a esse interesse, segundo o oráculo (RF11, RF12, RF13, RF14 e RF15). Isso aconteceu por causa da palavra-chave “logado”, presente nos requisitos funcionais.

Para os interesses Usabilidade e Desempenho, as medidas cobertura, precisão e *F-Measure* foram iguais, com e sem a utilização de *templates* para PRS. Isso pode ter ocorrido, pois, nos requisitos do sistema de software Health Watcher, existia apenas um requisito relacionado a cada um desses interesses. Para o interesse Portabilidade, quando DRS não foi escrito com *template* para PRS, ObasCId-Tool+ não encontrou o requisito correspondente; com isso, o resultado das medidas foi 0%.

Ao analisar a execução do estudo experimental piloto com *templates* para PRS, nota-se que, para o interesse Persistência, houve melhora, encontrando todos os requisitos correspondentes ao interesse. Consequentemente, o valor das medidas precisão e *F-Measure*

apresentaram valores 100%. Para o interesse Segurança, a cobertura foi 100%. Para a de precisão, não foram encontrados falsos positivos (Tabela 7.7). Isso mostra que houve melhora em relação à não utilização de *templates* para PRS. Ao analisar os interesses Usabilidade, de Desempenho e de Portabilidade, o uso de *templates* para PRS foi essencial para as medidas cobertura, precisão e *F-Measure* obterem resultados em 100%.

Uma vez constatada a possível validade da hipótese de pesquisa deste trabalho, vários outros estudos foram conduzidos para confirmar os resultados obtidos com o estudo piloto. A partir da próxima seção, são apresentados cinco estudos experimentais com DRS de diferentes domínios, como sistemas de informação e ferramentas educacionais.

7.5 Estudos Experimentais

Nos estudos experimentais foram utilizados 5 (cinco) sistemas de software escolhidos por conveniência para serem utilizados neste trabalho e de diferentes contextos de domínios, tais como sistemas de software gerencias, sistemas de software web e sistemas para dispositivos móveis. Na Tabela 7.9, são apresentados os sistemas de software utilizados nos estudos experimentais, os interesses presentes nos requisitos de acordo com cada oráculo, os PRS utilizados para padronizar os requisitos e as respectivas quantidades de requisitos presentes em cada um dos sistemas de software.

7.6 Estudo Experimental I - EASYHALF

O sistema de software EasyHalf é uma aplicação web para aprimorar o processo de obtenção do desconto de meia-entrada por parte dos estudantes, garantindo agilidade e confiabilidade a esse processo (TERRA *et al.*, 2020). Com esse sistema, pode-se verificar, em tempo real, a situação da matrícula de um estudante de uma instituição de ensino superior, no momento da solicitação do desconto. Para isso, o estudante deve apresentar ao representante do estabelecimento seu número de matrícula e um documento de identificação com foto (RG ou CNH).

7.6.1 Requisitos Funcionais e Não-Funcionais

Para esse estudo experimental, foram utilizados os requisitos do sistema de software EasyHalf (Tabela 7.10). Nessa tabela, são apresentados o código referente ao requisito funcional ou não-funcional, o nome do requisito contido no DRS do sistema de software EasyHalf e o interesse em cada requisito (esses interesses foram identificados com a ajuda

dos desenvolvedores do software). Os requisitos originais podem ser encontrados no APÊNDICE F e os requisitos padronizados podem ser encontrados no APÊNDICE G.

Tabela 7.9 - Sistemas de Software utilizados nos Estudos Experimentais.

Sistemas de Software	Interesses	PRS Utilizados	Quant. de Requisitos
I - EasyHalf	Persistência, Segurança	PRS de CRUD, PRS de Segurança	12
II - Fake News App	Persistência, Segurança, Gerenciar Notícia, Usabilidade, Portabilidade	PRS de CRUD, PRS de Segurança, PRS de Usabilidade, PRS de Portabilidade	13
III - ObasCId-Tool	Persistência, Responsividade, Segurança, Usabilidade	PRS de CRUD, PRS de Segurança, PRS de Usabilidade	30
IV - SisGAC	Persistência, Segurança, Login	PRS de CRUD, PRS de Segurança	20
V - Veredas Sol e Lares	Persistência, Segurança, Gerenciar comandos, Cálculos, Gerenciar alarmes e eventos, Histórico, Relatórios, Gestão de comunicação e coleta de dados, Alarmes e Controles	PRS de CRUD, PRS de Segurança; PRS de Usabilidade, PRS de Portabilidade, PRS de Desempenho	60

Fonte: Do autor (2020).

Tabela 7.10 - Requisitos Funcionais e Não-Funcionais e seus Interesses.

Código	Nome	Interesses
RF01	Cadastrar Estudantes	Persistência
RF02	Alterar Conta de Estudante	Persistência
RF03	Cadastrar Representantes de Estabelecimentos	Persistência
RF04	Cadastrar Instituição de Ensino	Persistência
RF05	Consultar Informações de Estudantes	Persistência
RF06	Consultar Situação dos Representantes de Estabelecimentos	Persistência
RF07	Alterar Informações dos Representantes de Estabelecimentos	Persistência
RF08	Alterar Informações dos Estudantes	Persistência
RF09	Alterar Autorização dos Estabelecimentos	Persistência
RF10	Alterar Informações da Instituição de Ensino	Persistência
RF11	Listar Estabelecimentos	Persistência
RNF01	Autenticação de Usuários	Segurança

Fonte: Do autor (2020).

7.6.2 Planejamento do Estudo Experimental I

O planejamento do Estudo Experimental I segue a mesma estrutura do estudo experimental piloto apresentado anteriormente, tendo como objetivo a avaliação da efetividade e da eficiência do uso de PRS para identificação dos interesses no sistema de software EasyHalf. As hipóteses e o catálogo de interesses utilizados no Estudo Experimental I são os mesmos do experimento piloto (Tabela 7.4 e Tabela 7.6, respectivamente).

Na Tabela 7.11, são apresentados os interesses contidos no sistema de software EasyHalf e os requisitos relacionados a eles, de acordo com (i) o oráculo (coluna 2), ObasCId-Tool+, executada no DRS original (coluna 3), e ObasCId-Tool+, executada no DRS usando *templates* para PRS (coluna 4).

Tabela 7.11 - Execução do Estudo Experimental I.

Interesse	Oráculo	Sem PRS	Com PRS
Persistência	RF01, RF02, RF03, RF04, RF05, RF06, RF07, RF08, RF09, RF10, RF11	RF05, RF06, RF09	RF01, RF02, RF03, RF04, RF05, RF06, RF07, RF08, RF09, RF10, RF11
Segurança	RNF01	RNF01, RF05, RF06, RF07, RF08, RF09, RF10, RF11	RNF01

Fonte: Do autor (2020).

7.6.3 Análise dos Resultados do Estudo Experimental I

Na Tabela 7.12, é apresentado o resultado com os cálculos das medidas descritas na Seção 7.3. Analisando os resultados obtidos na execução do Estudo Experimental I, pode-se notar que, sem o uso de *templates* para PRS, ObasCId-Tool+ não identificou corretamente o interesse *Persistência* nos requisitos funcionais RF01, RF02, RF03, RF04, RF07, RF08, RF10 e RF11. Além disso, ObasCId-Tool+ gerou falsos positivos. Foi encontrado o interesse *Segurança* nos requisitos RF05, RF06, RF07, RF08, RF09, RF10 e RF11 (sublinhados e em negrito na Tabela 7.11). Outro fator relevante é, nos requisitos RF05, RF06, RF07, RF08, RF09, RF10 e RF11, sem *templates* para PRS ter encontrado o interesse *Segurança*. Essa exceção pode ser explicada, pois, nesses requisitos, a palavra “autenticado” estava presente, mostrando que a utilização de *templates* para PRS teve impacto positivo na cobertura e na precisão no processo de identificação de interesses e, em especial, no resultado de *F-Measure* em 100%.

7.7 Estudo Experimental II - FAKE NEWS APP

O sistema de software Fake News App (PARREIRA JÚNIOR, 2020) é um aplicativo móvel desenvolvido para combater a propagação de notícias falsas, chamadas *fake news*. Com esse aplicativo, o usuário pode (i) encontrar as principais notícias falsas divulgadas por meio da Internet, juntamente com os motivos para serem consideradas falsas, (ii) “favoritar” notícias de seu interesse para facilitar acessá-las posteriormente e (iii) compartilhar notícias com outros usuários, por meio do aplicativo WhatsApp.

Tabela 7.12 - Resultados do Estudo Experimental I.

Interesse	DRS sem PRS			DRS com PRS		
	Cobertura	Precisão	<i>F-Measure</i>	Cobertura	Precisão	<i>F-Measure</i>
Persistência	27,3%	100%	42,9%	100%	100%	100%
Segurança	100%	12,5%	22,2%	100%	100%	100%

Fonte: Do autor (2020).

7.7.1 Requisitos Funcionais e Não-Funcionais

Foram utilizados os requisitos funcionais e não-funcionais do software Fake News App (Tabela 7.13). Nessa tabela, são apresentados o código referente ao requisito funcional ou não-funcional, o nome do requisito contido no DRS do sistema de software Fake News App e o interesse em cada requisito (os interesses foram identificados com a ajuda dos desenvolvedores do software). Os requisitos, em forma de casos de uso, podem ser encontrados no APÊNDICE H. Da mesma forma que aconteceu com os requisitos do sistema de software Health Watcher, os requisitos do sistema de software Fake News App foram transformados em texto narrativo e validados pelos seus desenvolvedores. Esses requisitos, em texto narrativo, podem ser verificados no APÊNDICE I e os requisitos padronizados podem ser encontrados no APÊNDICE J.

Tabela 7.13 - Requisitos Funcionais e Não-Funcionais e seus Interesses.

Código	Nome	Interesses
RF01	Criar conta	Persistência
RF02	Autenticar usuário	Segurança
RF03	Sair	Segurança
RF04	Listar/ buscar notícias	Persistência
RF05	Listar notícias favoritas	Persistência
RF06	Gerenciar notícias favoritas	Persistência
RF07	Gerenciar like da notícia	Persistência
RF08	Exibir notícia	Persistência
RF09	Compartilhar notícia	Gerenciar notícia
RF10	Ler mais sobre a notícia	Gerenciar notícia
RNF01	Facilidade de uso	Usabilidade
RNF02	Portabilidade	Portabilidade
RNF03	Segurança	Segurança

Fonte: Do autor (2020).

7.7.2 Planejamento do Estudo Experimental II

Na Tabela 7.14, são apresentados os interesses contidos no sistema de software Fake News App, os requisitos relacionados a eles, segundo o oráculo e ObasCId-Tool+, com e sem a utilização de *templates* para PRS.

7.7.3 Análise dos Resultados do Estudo Experimental II

Na Tabela 7.15, é apresentado o resultado com os cálculos das medidas descritas na Seção 7.3. Ao analisar os resultados alcançados na execução do Estudo Experimental II (Tabela 7.14), pode-se verificar que a não utilização de *templates* para PRS levou a valores inferiores para as medidas analisadas. Contudo, em nenhuma das situações (com e sem o uso de *templates* para PRS), o interesse Usabilidade foi identificado. Percebe-se que isso ocorreu por causa do requisito não-funcional RNF01 não conter qualquer palavra-chave que identifique o interesse. Em decorrência de não ter encontrado esse interesse, mesmo com a utilização de *templates* para PRS, conclui-se que o catálogo de interesses utilizado no experimento precisa ser atualizado, com a adição de novos sinônimos das palavras-chave existentes referentes ao interesse Usabilidade.

Tabela 7.14 - Execução do Estudo Experimental II.

Interesse	Oráculo	Sem PRS	Com PRS
Persistência	RF01, RF04, RF05, RF06, RF07, RF08	RF06, RF07	RF01, RF04, RF05, RF06, RF07, RF08
Usabilidade	RNF01	-----	-----
Portabilidade	RNF02	-----	RNF02
Segurança	RF02, RF03, RNF03	RF03, RNF03, RF04, RF05, RF06, RF07, RF08, RF09, RF10	RF02, RF03, RNF03, RF09, RF10

Fonte: Do autor (2020).

Tabela 7.15 - Resultados do Estudo Experimental II.

Interesse	DRS sem PRS			DRS com PRS		
	Cobertura	Precisão	F-Measure	Cobertura	Precisão	F-Measure
Persistência	33,3%	100%	50%	100%	100%	100%
Usabilidade	0%	0%	0%	0%	0%	0%
Portabilidade	0%	0%	0%	100%	100%	100%
Segurança	66,7%	22,2%	33,3%	100%	60%	75%

Fonte: Do autor (2020).

Para o interesse Persistência, sem o uso de *templates* para PRS, pode-se observar que a medida cobertura obteve resultado 33,3%; por outro lado, o uso de *templates* para PRS levou esse valor ao total de 100%. Para a medida precisão, o valor foi o mesmo (100%) para ambos os cenários. Conforme ocorreu no Estudo Experimental I, foram encontrados falsos positivos quanto ao interesse Segurança, com e sem a utilização de *templates* para PRS. Porém, com a utilização de *templates* para PRS foi encontrado somente os requisitos RF09 e RF10, pois ambos os requisitos não têm um PRS para melhorar a identificação dos interesses principais presentes. Para o interesse de Portabilidade, apesar de somente existir um requisito pertencente a esse interesse, o uso de *templates* para PRS permitiu encontrá-lo.

7.8 Estudo Experimental III - OBASCID-TOOL

Um apoio computacional denominado *ObasCId-Tool* para automatização de algumas das atividades da abordagem *ObasCId* foi desenvolvido para minimizar parte dos problemas que o uso de uma abordagem manual apresenta.

7.8.1 Requisitos Funcionais e Não-Funcionais

Os requisitos de *ObasCId-Tool* são descritos de acordo com o modelo proposto para a abordagem *ObasCId* (PARREIRA JÚNIOR, 2015). Foram utilizados os requisitos funcionais e não-funcionais do sistema de software *ObasCId-Tool* (Tabela 7.16) que apresenta o código referente ao requisito funcional ou não-funcional, o nome do requisito contido no DRS do sistema de software *ObasCId-Tool* e o interesse em cada requisito (esses interesses foram identificados e catalogados no trabalho de PARREIRA JÚNIOR (2015)). Os requisitos originais da *ObasCId-Tool* podem ser encontrados no APÊNDICE K e os requisitos padronizados podem ser encontrados no APÊNDICE L.

Tabela 7.16 - Requisitos Funcionais e Não-Funcionais e seus Interesses.

Código	Nome	Interesses	Código	Nome	Interesses	Código	Nome	Interesses
RF01	RF-01	Persistência	RF11	RF-11	Persistência	RF21	RF-21	Persistência
RF02	RF-02	Persistência	RF12	RF-12	Persistência	RF22	RF-22	Persistência
RF03	RF-03	Persistência	RF13	RF-13	Persistência	RF23	RF-23	Persistência
RF04	RF-04	Persistência	RF14	RF-14	Persistência	RF24	RF-24	Persistência
RF05	RF-05	Persistência	RF15	RF-15	Persistência	RF25	RF-25	Persistência
RF06	RF-06	Persistência	RF16	RF-16	Persistência	RF26	RF-26	Persistência
RF07	RF-07	Persistência	RF17	RF-17	Persistência	RF27	RF-27	Persistência
RF08	RF-08	Persistência	RF18	RF-18	Persistência	RNF01	RNF-1	Responsividade
RF09	RF-09	Persistência	RF19	RF-19	Persistência	RNF02	RNF-2	Segurança
RF10	RF-10	Persistência	RF20	RF-20	Persistência	RNF03	RNF-3	Usabilidade

Fonte: Do autor (2020).

7.8.2 Planejamento do Estudo Experimental III

Na Tabela 7.17, são apresentados os requisitos identificados pelo *ObasCId-Tool+*, com e sem a utilização de *templates* para PRS.

Tabela 7.17 - Execução do Estudo Experimental III.

Interesse	Oráculo	Sem PRS	Com PRS
Persistência	RF01, RF02, RF03, RF04, RF05, RF06, RF07, RF08, RF09, RF10, RF11, RF12, RF13, RF14, RF15, RF16, RF17, RF18, RF19, RF20, RF21, RF22, RF23, RF24, RF25, RF26, RF27	RF01, RF02, RF03, RF04, RF05, RF06, RF07, RF08, RF09, RF11, RF12, RF13, RF14	RF01, RF02, RF03, RF04, RF05, RF06, RF07, RF08, RF09, RF10, RF11, RF12, RF13, RF14, RF15, RF16, RF17, RF18, RF19, RF20, RF21, RF22, RF23, RF24, RF25, RF26, RF27

(continua)

Tabela 7.17 - Execução do Estudo Experimental III (continuação).

Interesse	Oráculo	Sem PRS	Com PRS
Segurança	RNF02	RNF02	RNF02
Usabilidade	RNF03	RNF03	RNF03

Fonte: Do autor (2020).

7.8.3 Análise dos Resultados do Estudo Experimental III

Na Tabela 7.18, são apresentados os resultados das medidas de cobertura, precisão e *F-Measure*. Ao analisar os resultados do Estudo Experimento III, é possível verificar que são bem próximos dos anteriores. Para a cobertura do interesse *Persistência*, o valor foi 48,1%, pois sem a utilização de *templates* para PRS, *ObasCId-Tool+* não encontrou os requisitos RF10, RF15, RF16, RF17, RF18, RF19, RF20, RF21, RF22, RF23, RF24, RF25, RF26 e RF27. Para os interesses *Segurança* e *Usabilidade*, o valor das medidas foi, em sua totalidade, 100%, encontrando todos os requisitos pertencentes aos interesses apresentados. Na execução do Estudo Experimental III com *templates* para PRS, os resultados obtidos foram superiores, pois em relação ao interesse *Persistência*, *ObasCId-Tool+* cobriu 100% a identificação dos interesses nos requisitos. Para os interesses *Segurança* e *Usabilidade*, os valores formam iguais aos obtidos sem PRS.

Tabela 7.18 - Resultados do Estudo Experimental III.

Interesse	DRS sem PRS			DRS com PRS		
	Cobertura	Precisão	<i>F-Measure</i>	Cobertura	Precisão	<i>F-Measure</i>
Persistência	48,1%	100%	64,9%	100%	100%	100%
Segurança	100%	100%	100%	100%	100%	100%
Usabilidade	100%	100%	100%	100%	100%	100%

Fonte: Do autor (2020).

7.9 Estudo Experimental IV - Sistema Gerenciador de Atividades Curriculares (SisGAC)

O sistema de *software* Sistema Gerenciador de Atividades Curriculares (SisGAC) (BATISTA, 2020) visa facilitar o gerenciamento das entregas, bem como o acompanhamento do progresso dos discentes em um cenário de aplicação da metodologia de aprendizagem baseada em projetos com recursos de gamificação. Para isso, desenvolveu-se uma ferramenta computacional denominada SisGAC. Assim sendo, no sistema de software SisGAC, os professores conseguem criar conjuntos de atividades, adicionar atividades para esses conjuntos e vincular esses conjuntos de atividades em suas disciplinas. Após isso, eles podem disponibilizar aos alunos, de modo que eles possam realizar as atividades dessas disciplinas e

enviar aos professores. Uma vez enviadas as atividades, o professor pode proceder com a avaliação dessas atividades, pontuando-as e atribuindo troféus, caso necessário.

7.9.1 Requisitos Funcionais e Não-Funcionais

Foram utilizados os requisitos funcionais do sistema de software *SisGAC* (Tabela 7.9). Nessa tabela, são apresentados o código referente ao requisito funcional ou não-funcional, o nome do requisito contido no DRS do sistema de software *SisGAC* e o interesse em cada requisito (os interesses foram identificados com a ajuda dos desenvolvedores do sistema). Os requisitos detalhados podem ser encontrados no APÊNDICE M e os requisitos padronizados no APÊNDICE N.

Tabela 7.19 - Requisitos Funcionais e Não-Funcionais e seus Interesses.

Código	Nome	Interesses	Código	Nome	Interesses	Código	Nome	Interesses
RF01	1	Persistência	RF10	10	Persistência	RF17	17	Persistência
RF03	3	Persistência	RF11	11	Persistência	RF18	18	Persistência
RF05	5	Persistência	RF12	12	Persistência	RF19	19	Persistência
RF06	6	Persistência	RF13	13	Persistência	RF20	20	Persistência
RF07	7	Persistência	RF14	14	Persistência	RNF01	2	Segurança
RF08	8	Persistência	RF15	15	Persistência	RNF02	4	Login
RF09	9	Persistência	RF16	16	Persistência			

Fonte: Do autor (2020).

7.9.2 Planejamento do Estudo Experimental IV

O oráculo desse Estudo Experimental IV, bem como os resultados com a execução de *ObasCId-Tool*, estão na Tabela 7.20.

Tabela 7.20 - Execução do Estudo Experimental IV.

Interesse	Oráculo	Sem PRS	Com PRS
Persistência	RF01, RF03, RF05, RF06, RF07, RF08, RF09, RF10, RF11, RF12, RF13, RF14, RF15, RF16, RF17, RF18, RF19, RF20	RF03, RF07, RF09	RF01, RF03, RF05, RF06, RF07, RF08, RF09, RF10, RF11, RF12, RF13, RF14, RF15, RF16, RF17, RF18, RF19, RF20
Segurança	RNF01	RNF01	RNF01

Fonte: Do autor (2020).

7.9.3 Análise dos Resultados do Estudo Experimental IV

Na Tabela 7.21, é apresentado o resultado com os cálculos das medidas descritas na Seção 7.3. Ao avaliar os resultados obtidos na execução do Estudo Experimental IV, percebe-se que a execução sem a utilização de *templates* para PRS apresentou baixa cobertura para o interesse *Persistência*, cobrindo apenas 17,6% dos requisitos (requisitos RF03, RF07 e RF09). Para o interesse *Segurança*, os resultados para as medidas cobertura, precisão e *F-*

Measure foi 100% com e sem uso de *templates* para PRS. Verificando os dados obtidos com a utilização de *templates* para PRS para o interesse *Persistência*, nota-se que, apesar de não cobrir todos os requisitos, houve melhora na cobertura, passando de 17,6% para 100%.

Tabela 7.21 - Resultados do Estudo Experimental IV.

Interesse	DRS sem PRS			DRS com PRS		
	Cobertura	Precisão	F-Measure	Cobertura	Precisão	F-Measure
Persistência	17,6%	100%	29,9%	100%	100%	100%
Segurança	100%	100%	100%	100%	100%	100%

Fonte: Do autor (2020).

7.10 Estudo Experimental V - VEREDAS SOL E LARES

O sistema de software fotovoltaico flutuante *Veredas Sol e Lares* é um sistema real, em desenvolvimento por uma empresa especializada em desenvolvimento de software de monitoramento e controle. Esse sistema consiste em coletar dados dos componentes do sistema, desde os módulos fotovoltaicos até a saída de energia para a concessionária de energia elétrica.

7.10.1 Requisitos Funcionais e Não-Funcionais

Nesse Estudo Experimental, foram utilizados os requisitos do sistema de software *Veredas Sol e Lares* (Tabela 7.22). Na tabela, são apresentados o código referente ao requisito funcional ou não-funcional, o nome do requisito contido no DRS do sistema de software *Veredas Sol e Lares* e o interesse em cada requisito (os interesses foram identificados com a ajuda dos desenvolvedores do sistema de software). Os requisitos detalhados utilizados no Estudo Experimental V podem ser encontrados no APÊNDICE O e os requisitos padronizados podem ser encontrados no APÊNDICE P.

Tabela 7.22 - Requisitos Funcionais e Não-Funcionais e seus Interesses.

Código	Nome	Interesses
RF01	Manter usuário	Persistência
RF02	Manter perfis de acesso	Persistência
RF03	Realizar login	Segurança
RF04	Realizar logout	Segurança
RF05	Relembrar senha	Segurança
RF06	Alterar senha	Persistência
RF07	Registrar ações	Persistência
RF08	Desativar e reativar item	Persistência
RF10	Manter UFV	Persistência
RF11	Manter planta	Persistência
RF14	Gerenciar comandos	Gerenciar comandos
RF15	Exibir diagrama unifilar	Persistência
RF16	Manter Contrato	Persistência
RF17	Manter parâmetros gerais	Persistência

(continua)

Tabela 7.22 - Requisitos Funcionais e Não-Funcionais e seus Interesses (continuação).

Código	Nome	Interesses
RF20	Visualizar painel principal	Persistência
RF21	Visualizar esquemático de processo da usina	Persistência
RF22	Gerenciar manutenções	Persistência
RF23	Acompanhar manutenções	Persistência
RF24	Gerar curva I/V	Persistência
RF30	Realizar balanço de energia após o faturamento	Persistência
RF31	Calcular eficiência por inversor	Cálculos
RF32	Calcular energia estimada	Cálculos
RF33	Calcular energia gerada	Cálculos
RF34	Calcular desempenho	Cálculos
RF40	Gerenciar alarmes e eventos	Gerenciar alarmes e eventos
RF41	Categorizar alarme	Persistência
RF42	Alertar operadores	Gerenciar alarmes e eventos
RF43	Consultar sequência de eventos	Persistência
RF44	Configurar limites	Gerenciar alarmes e eventos
RF45	Gerar alarme/evento violação de limite	Persistência
RF46	Monitorar limites	Persistência
RF47	Verificar limites	Persistência
RF50	Armazenar dados históricos	Persistência
RF51	Receber e armazenar leituras	Histórico
RF52	Enviar atualizações de configurações	Histórico
RF53	Informar equipamentos novos	Histórico
RF60	Visualizar dados históricos	Persistência
RF71	Gerar relatório de sequência de eventos	Persistência
RF72	Gerar relatório de dados climatológicos	Persistência
RF73	Gerar relatório desempenho de planta	Persistência
RF74	Exportar relatórios	Relatórios
RF75	Gerar relatório de eficiência de inversor	Persistência
RF80	Calcular disponibilidade do sistema de comunicação	Gestão de comunicação e coleta de dados
RF81	Monitorar falhas de comunicação	Gestão de comunicação e coleta de dados
RF82	Gerenciar comunicação de equipamento	Gestão de comunicação e coleta de dados
RF100	Classificar alarmes	Alarmes
RF101	Realizar diagnóstico	Alarmes
RF102	Tratar alarmes	Alarmes
RF110	Calcular performance	Controles
RF111	Realizar controle entre geração UFV X concessionária	Controles
RF120	Calcular indisponibilidade programada (IP)	Cálculos
RF121	Calcular indisponibilidade Forçada - TEIF	Cálculos
RF122	Calcular economia de carbono	Cálculos
RF123	Calcular número de famílias beneficiadas	Cálculos
RF140	Monitorar inequações	Inequações
RF150	Realizar previsão de geração diária e semanal	Inequações
RF151	Realizar previsão de geração de curto prazo (horária)	Inequações
RNF01	Arquitetura	Funcionalidade
RNF02	Usabilidade	Usabilidade
RNF03	Portabilidade	Portabilidade
RNF04	Gerar alarmes	Desempenho
RNF05	Disponibilidade do sistema	Disponibilidade

Fonte: Do autor (2020).

7.10.2 Planejamento do Estudo Experimental V

Na Tabela 7.23, são apresentados os interesses do sistema de software *Veredas Sol e Lares* identificados por *ObasCId-Tool+*, com e sem o uso de *templates* para PRS. Cabe ressaltar que *ObasCId-Tool+* usa um catálogo de interesses que não contempla alguns dos

interesses existentes no sistema de software *Veredas Sol e Lares*; por isso, eles não aparecem na tabela.

Tabela 7.23 - Execução do Estudo Experimental V.

Interesse	Oráculo	Sem PRS	Com PRS
Segurança	RF03, RF04, RF05	RF03	RF03
Desempenho	RNF04	-----	RNF04
Portabilidade	RNF03	-----	RNF03
Persistência	RF01, RF02, RF06, RF07, RF08, RF10, RF11, RF15, RF16, RF17, RF20, RF21, RF22, RF23, RF24, RF30, RF41, RF43, RF45, RF46, RF47, RF50, RF60, RF71, RF72, RF73, RF75	RF22	RF01, RF02, RF06, RF07, RF08, RF10, RF11, RF15, RF16, RF17, RF20, RF21, RF22, RF23, RF24, RF30, RF41, RF43, RF45, RF46, RF47, RF50, RF60, RF71, RF72, RF73, RF75
Usabilidade	RNF02	RNF02	RNF02

Fonte: Do autor (2020).

7.10.3 Análise dos Resultados do Estudo Experimental V

Na Tabela 7.24, é apresentado o resultado com os cálculos das medidas descritas na Seção 7.3. Em todos os estudos experimentais realizados, os valores das medidas eram bem próximos, com ou sem a utilização de *templates* para PRS. Pode-se conjecturar que os DRS dos sistemas de software anteriores, por terem sido escritos por Engenheiros de Software da academia, estão mais próximos aos que se espera para a correta identificação dos interesses do software. Porém, ao analisar os dados obtidos com o sistema de software *Veredas Sol e Lares*, nota-se maior diferença nos resultados.

O interesse *Persistência*, sem a utilização dos *templates* para PRS, obteve valor de cobertura muito baixo, tendo em vista que somente foi identificado o interesse no requisito RF22. Para a medida precisão nesse interesse, o resultado foi 100% por ter identificado corretamente esse requisito segundo o oráculo. Porém, ao verificar o resultado da medida *F-Measure*, nota-se valor muito abaixo do registrado nos estudos experimentais anteriores, visto que esse valor foi “puxado para baixo” pela cobertura.

Tabela 7.24 - Resultados do Estudo Experimental V.

Interesse	DRS sem PRS			DRS com PRS		
	Cobertura	Precisão	<i>F-Measure</i>	Cobertura	Precisão	<i>F-Measure</i>
Persistência	3,6%	100%	6,9%	100%	100%	100%
Usabilidade	100%	100%	100%	100%	100%	100%
Segurança	33,3%	100%	50%	33,3%	100%	50%

(continua)

Tabela 7.24 - Resultados do Estudo Experimental V (continuação).

Interesse	DRS sem PRS			DRS com PRS		
	Cobertura	Precisão	<i>F-Measure</i>	Cobertura	Precisão	<i>F-Measure</i>
Desempenho	0%	0%	0%	100%	100%	100%
Portabilidade	0%	0%	0%	100%	100%	100%

Fonte: Do autor (2020).

Para o interesse Usabilidade, as medidas cobertura, precisão e *F-Measure* apresentaram valores satisfatórios. Ao verificar o interesse Segurança, com e sem utilizar *templates* para PRS, ObasCId-Tool+ encontrou os mesmos valores. O valor da cobertura foi 33,3%, o valor da precisão foi 100% e o valor da *F-Measure* foi 50%. Esse fator é levado em consideração, pois pode-se concluir que o catálogo de interesses utilizado no experimento precisa ser atualizado com a adição de novos sinônimos das palavras-chave existentes, referentes ao interesse Usabilidade. Assim, deve-se identificar e classificar corretamente os interesses nos requisitos RF04 e RF05.

No interesse Persistência, com a utilização de *templates* para PRS, o resultado da cobertura foi significativamente maior, obtendo o valor 89,3%. Para os interesses Desempenho e Portabilidade, os resultados com a utilização dos *templates* para PRS foram mais uma vez satisfatórios, pois somente com o uso dos *templates* para PRS foi possível identificar os requisitos RNF04 e RNF03, com os interesses Desempenho e Portabilidade, respectivamente, resultando em 100% como resultado para todas as medidas. Para esses interesses, apesar de somente existir um requisito, o uso de *templates* para PRS permitiu encontrá-los corretamente, ao contrário do que ocorreu sem o uso de *templates* para PRS.

7.11 Interesses Analisados nos Estudos Experimentais

Com o intuito de sumarizar os resultados dos estudos experimentais, exceto o estudo experimental piloto, na Tabela 7.25, são apresentados os interesses tratados neste trabalho sem a utilização dos *templates* para PRS. Nessa tabela, são apresentados os interesses, a quantidade de requisitos funcionais afetados pelo interesse, juntamente com as respectivas medidas de cobertura, precisão e *F-Measure*.

Nota-se que a quantidade de requisitos funcionais e não-funcionais sem *templates* para PRS afetados por algum dos interesses identificados e classificados é baixo se comparado ao total de 156 requisitos utilizados nos estudos experimentais. Visto isso, o valor resultante das medidas calculadas, considerando todos os estudos experimentais, foi diretamente afetado pela quantidade de interesses, pois, dos requisitos cobertos corretamente pela abordagem, nenhum

dos interesses obteve cobertura de 100% sem a utilização de *templates* para PRS. Apesar da medida precisão considerar a identificação e classificação correta dos interesses, no interesse Segurança, houve incidência de falsos positivos, resultando em precisão 70% menor aos demais interesses sem *templates* para PRS. Mesmo que os valores obtidos na precisão dos interesses Persistência e Usabilidade tenham alcançado 100%, ao calcular a medida *F-Measure* obtém-se resultados igual ou inferior a 80%. No interesse Segurança, o valor obtido nas medidas cobertura e precisão afetou diretamente o resultado da medida *F-Measure*, alcançando valor inferior a 50%.

Tabela 7.25 - Interesses Analisados sem PRS.

Interesses	Quant. de Requisitos Afetados	Cobertura	Precisão	<i>F-Measure</i>
Persistência	22	24,7%	100%	39,6%
Segurança	6	66,7%	30%	41,4%
Usabilidade	2	66,7%	100%	80%
Portabilidade	0	0%	0%	0%
Desempenho	0	0%	0%	0%

Fonte: Do autor (2020).

Com o mesmo intuito, na Tabela 7.26, é apresentada uma síntese do resultado dos estudos experimentais, considerando a utilização de *templates* para PRS. Como pode-se observar, houve aumento nas medidas cobertura e *F-Measure* para todos os interesses considerados. Com a utilização de *templates* para PRS, é possível notar que a quantidade de requisitos funcionais e não-funcionais afetados por algum interesse foi 45,5% maior do que sem a utilização de *templates* para PRS, se comparado ao total de requisitos de todos os sistemas de software.

Tabela 7.26 - Interesses Analisados com PRS

Interesses	Quant. de Requisitos Afetados	Cobertura	Precisão	<i>F-Measure</i>
Persistência	89	100%	100%	100%
Segurança	7	77,8%	77,8%	77,8%
Usabilidade	2	66,7%	100%	80%
Portabilidade	2	100%	100%	100%
Desempenho	1	100%	100%	100%

Fonte: Do autor (2020).

Com a utilização de *templates* para PRS obteve-se melhores resultados na medida cobertura em todos os interesses, somente os interesses Segurança e Usabilidade não alcançaram a totalidade, pois palavras-chave não foram suficientes para identificar e classificar os interesses. Para a medida precisão, com exceção do interesse Segurança, por causa da incidência de falsos positivos, os demais interesses atingiram 100%. Com isso, na medida *F-Measure* dos interesses Segurança e Usabilidade obtiveram valores inferiores a 100%.

Portanto, ao analisar ambas sínteses dos resultados sem e com uso de *templates* para PRS, sugere-se que a utilização de *templates* para PRS para escrever DRS contribui para melhoria da efetividade da identificação e classificação de interesses em sistemas de software. Isso deve-se ao fato de que a utilização de *templates* para PRS diminui a quantidade de falsos positivos e aumenta a cobertura, considerando que em todos os estudos experimentais foi utilizado o mesmo catálogo de interesse.

Na Tabela 7.27, são apresentados os sistemas de software utilizados nos estudos experimentais, a quantidade de requisitos funcionais e não funcionais e o valor calculado para cada medida sem e com uso de *templates* para PRS (cobertura, precisão e *F-Measure*).

Tabela 7.27 - Análise das Medidas por Sistemas de Software Sem e Com Uso de PRS.

Sistemas de Software	Quantidade de Requisitos	Cobertura		Precisão		<i>F-Measure</i>	
		Sem PRS	Com PRS	Sem PRS	Com PRS	Sem PRS	Com PRS
I - EasyHalf	12	33,3%	100%	36,4%	100%	34,8%	100%
II - Fake News App	13	36,4%	90,9%	36,6%	83,3%	36,5%	86,9%
III - ObasCId-Tool	30	51,7%	100%	100%	100%	68,2%	100%
IV - SisGAC	20	21%	100%	100%	100%	34,7%	100%
V - Veredas Sol e Lares	60	9,1%	94%	100%	100%	16,7%	96,9%

Fonte: Do autor (2020).

Ao analisar as medidas cobertura, precisão e *F-Measure* resultantes de cada sistema de software utilizado nos estudos experimentais, nota-se que é possível considerar a utilização de *templates* para PRS para escrever DRS, pois contribui para a melhoria da efetividade da identificação e classificação de interesses em sistemas de software. Considerando os 135 requisitos afetados por algum dos interesses presentes nos requisitos funcionais e não-funcionais, os valores resultantes com *templates* para PRS foram superiores, visto que, para algumas medidas e para alguns sistemas de software, os valores foram 100% (EasyHalf, ObasCId-Tool e SisGAC). Nos sistemas de software Fake News App e Veredas Sol e Lares com *templates* para PRS não obtiveram cobertura 100% por alguns requisitos não serem identificados pela falta de palavras-chave no catálogo de interesses. Contudo, mesmo com essa ocorrência, a utilização de *templates* para PRS proporcionou cobertura de 84,9% maior no sistema de software Veredas Sol e Lares e cobertura de 54,5% maior no sistema Fake News App.

7.12 Testes de Hipóteses

Ainda que os valores apresentados anteriormente indiquem que a utilização de *templates* para PRS nos requisitos funcionais e não-funcionais leva a aumento da cobertura, precisão e *F-Measure*, com relação à identificação e classificação de interesses de software, faz-se

necessário realizar a análise estatística dos dados, por meio de testes de hipótese para garantir maior confiabilidade às afirmações feitas.

O objetivo de um teste de hipótese é verificar se a hipótese nula ($H_{0_cobertura}$, $H_{0_precisão}$, $H_{0_F-measure}$) pode ser rejeitada com algum grau de relevância (confiança de tomada a decisão correta), chegando à aceitação da hipótese alternativa ($H_{1_cobertura}$, $H_{1_precisão}$, $H_{1_F-measure}$). Antes de aplicar um teste de hipóteses, é necessário conhecer sobre qual tipo de distribuição de probabilidade os dados coletados no estudo encontram-se organizados, pois alguns testes de hipóteses (*e.g.*, *t-test* (MONTGOMERY, 2017)) possuem como pré-requisito a necessidade dos dados serem distribuídos normalmente.

Para verificar a normalidade dos dados, um teste comumente utilizado é *Shapiro-Wilk* (MONTGOMERY, 2017). A hipótese nula do teste de *Shapiro-Wilk* é os dados estarem normalmente distribuídos. Se os dados não estiverem normalizados, será utilizado o teste de *Mann-Whitney* (WILCOXON *et al.*, 1970), que considera valores de entrada sem estarem normalizados. Caso o valor da probabilidade signifique que deva rejeitar a hipótese nula, a hipótese alternativa deverá ser aceita. Porém, se, para determinado grau de significância (*p-value*) for maior para a hipótese nula, essa deverá ser aceita.

O teste de hipóteses para o estudo experimental piloto com o sistema de software *Health Watcher* não foi aplicado nesse momento, pois não houve significância considerável para aplicação do *t-test*. Isso se deve ao fato dos valores obtidos em ambos os cenários do experimento (com e sem utilização de *templates* para PRS) foram bem próximos. Porém, o experimento piloto serviu para prover indícios de que seria interessante dar continuidade aos estudos experimentais com sistemas de softwares reais.

Na Tabela 7.28, são apresentados os resultados dos testes de hipóteses, realizados por interesse de software, encontrados em cada um dos estudos experimentais por interesse, descrevendo, para cada hipótese nula ou alternativa, o valor do *p-value* obtido com o teste. Cabe ressaltar que o menor valor de *p-value* aceito para rejeitar a hipótese nula neste trabalho é 0,05.

Tabela 7.28 - Teste de Hipótese por Interesse de Software.

Interesse	Cobertura	Precisão	F-Measure
Persistência	$H_{1_cobertura}$ (<i>p-value</i> = 0,0027)	$H_{0_precisão}$ (<i>p-value</i> = 0,92034)	$H_{1_F-measure}$ (<i>p-value</i> = 0,00338)
Segurança	$H_{0_cobertura}$ (<i>p-value</i> = 0,61708)	$H_{0_precisão}$ (<i>p-value</i> = 0,1936)	$H_{1_F-measure}$ (<i>p-value</i> = 0,038063)

(continua)

Tabela 7.28 - Teste de Hipótese por Interesse de Software (continuação).

Interesse	Cobertura	Precisão	<i>F-Measure</i>
Usabilidade	-	-	-
Portabilidade	-	-	-
Desempenho	-	-	-

Fonte: Do autor (2020).

Como pode ser observado, ao verificar o interesse *Persistência*, apenas a hipótese nula da precisão ($H_{0_precisão}$) foi aceita, pois, para essa medida, não houve diferença significativa quanto à precisão global média proporcionado pelo uso dos *templates* para PRS com relação ao não uso de *templates* para PRS no processo de identificação e classificação de interesses de software, admitindo a hipótese nula. Nas demais medidas, houve diferença significativa no grau de significância (cobertura e *F-Measure*).

Para o interesse *Segurança*, há diferença quanto à *F-Measure*, proporcionado pelo uso de *templates* para PRS no processo de identificação e classificação de interesses de software. Somente a medida *F-Measure* (conforme sublinhado) foi possível realizar o *t-test*, pois, ao verificar os dados das amostras para essa medida no teste de *Shapiro-Wilk*, constatou que os dados estavam normalizados.

Considerando os interesses *Usabilidade*, *Portabilidade* e *Desempenho*, não foi possível aplicar os testes de hipóteses para esses interesses, pois em alguns dos sistemas de software utilizados nos estudos experimentais não apresentavam os interesses selecionados, não tendo o mínimo de dados para completar o cálculo do teste de *Mann-Whitney*, que deve ser com amostras que não sejam maiores 200 e menores que 5, visto que os dados não estavam normalizados.

Ao analisar o valor de cobertura para o interesse *Persistência*, dado que $p\text{-value} = 0,0027$, houve diferença estatística com a utilização de *templates* para PRS, corroborando com os resultados obtidos na cobertura dos interesses com PRS apresentados na Tabela 7.26. Isso evidência que a utilização de PRS para escrever DRS contribui significativamente para a melhoria da efetividade na identificação e classificação de interesses em sistemas de software. Porém, na medida precisão não houve diferença estatística ao escrever DRS utilizando *templates* para PRS, confirmando os dados apresentados na Tabela 7.25 e na Tabela 7.26. Para a medida *F-Measure*, obteve-se $p\text{-value} = 0,0038$, mostrando significância estatística ao utilizar *templates* para PRS para escrever DRS, contribuindo positivamente para a melhoria da efetividade na identificação e classificação de interesses em sistemas de software.

Enquanto que, para o interesse *Segurança*, somente a medida *F-Measure* mostrou diferença estatística ao utilizar *templates* para PRS, pois, ao verificar na Seção 7.11, esse

interesse foi o único que gerou falsos positivos em ambos os cenários, sem e com uso de *templates* para PRS. Somente nessa medida para o interesse Segurança foi possível aplicar o *t-test*, pois os valores das amostras estavam normalizados. Para os interesses Usabilidade, Portabilidade e Desempenho, não foi possível aplicar o teste de hipótese, pois não haviam valores disponíveis em todos os requisitos dos sistemas de software, apresentando erro ao serem executados.

Na Tabela 7.29, são apresentados os resultados dos testes de hipóteses realizados em cada sistema de software utilizado nos estudos experimentais. Para isso, descreveu-se cada hipótese nula ou alternativa e o valor do *p-value* obtido com o teste. Conforme nos testes de hipóteses anteriores, o menor valor de *p-value* aceito para rejeitar a hipótese nula é 0,05.

Tabela 7.29 - Teste de Hipótese por Sistema de Software.

Estudo Experimental	Cobertura	Precisão	F-Measure
I - EasyHalf	-	-	-
II - Fake News App	-	-	-
III - ObasCID-Tool	-	-	-
IV - SisGAC	-	-	-
V - Veredas Sol e Lares	H ₀ _cobertura (<i>p-value</i> = 0,06288)	H ₀ _precisão (<i>p-value</i> = 0,34722)	H ₁ _F-measure (<i>p-value</i> = 0,04444)

Fonte: Do autor (2020).

Como pode ser observado, ao verificar os sistemas de software EasyHalf, Fake News App, ObasCID-Tool e SisGAC, não foi possível aplicar os testes de hipóteses, pois possuíam os cinco interesses avaliados, sendo necessário pelo menos cinco valores de amostras para o cálculo do *t-test*, caso os dados estivessem normalizados, ou o teste de *Mann-Whitney*, caso os dados não estivessem normalizados. Para os sistemas de software Fake News App e Veredas Sol e Lares, somente a hipótese alternativa H₁_F-measure faz diferença quanto à medida *F-Measure* com relação ao uso de *templates* para PRS no processo de identificação e classificação de interesses de software, aceitando a hipótese alternativa. Considerando todos os estudos experimentais apenas a hipótese nula H₀_precisão, esse caso não faz diferença quanto à precisão global média proporcionado pelo uso dos *templates* para PRS com relação ao não uso de *templates* para PRS no processo de identificação e classificação de interesses de software, admitindo-se a hipótese nula nesse caso.

Somente no sistema de software Veredas Sol e Lares foi possível executar os testes de hipóteses, visto que somente nesse sistema de software são encontrados todos os interesses de acordo com o oráculo. Ao analisar a Tabela 7.27, percebe-se melhoria nas medidas cobertura e precisão com o uso de *templates* para PRS, contudo, ao aplicar o teste de hipóteses, não houve diferença estatística entre utilizar e não utilizar os *templates* para PRS. Por outro

lado, ao aplicar o teste de hipóteses na medida *F-Measure*, houve melhora significativa ao utilizar *templates* para PRS.

7.13 Ameaças à Validade dos Estudos Experimentais

Qualquer estudo experimental está sujeito a situações que podem ameaçar a validade dos resultados obtidos (WOHLIN *et al.*, 2012). As principais ameaças tratadas neste estudo são:

- a) **Ameaças à validade de conclusão.** Refere-se às questões que afetam a habilidade de tirar conclusões corretas a respeito do objeto de estudo dos estudos experimentais. Um exemplo de ameaça desse tipo diz respeito à escolha do método estatístico adequado para a análise dos dados. Neste trabalho, os testes estatísticos adotados para comparação dos valores médios das métricas foram *t-test* e o teste de *Mann-Whitney*. O *t-test* requer que os dados das amostras estejam normalmente distribuídos; dessa forma, antes de aplicá-lo, tal pressuposto foi verificado por meio do teste de *Shapiro-Wilk*. Para os casos em que o resultado do teste de *Shapiro-Wilk* foi negativo, foi utilizado o teste de *Mann-Whitney*;
- b) **Ameaças à validade interna.** Refere-se às questões que afetam a habilidade de assegurar que os resultados foram obtidos em decorrência dos tratamentos (isto é, da utilização da abordagem com e sem utilizar *templates* para PRS) e não por uma eventualidade. Uma ameaça desse tipo pode ser o modo como os requisitos foram selecionados e agrupados. Outro ponto que pode ter influenciado os resultados foi a utilização de DRS que tenham poucos requisitos para determinados *templates* para PRS, tais como, os de CRUD, de Segurança, de Desempenho, de Usabilidade e de Portabilidade. Como forma de mitigar essas ameaças, o catálogo de interesses e as palavras-chave não foram criados no trabalho, mas adquiridos por terceiros. Assim, foi utilizado o mesmo em ambas as execuções do experimento piloto, com e sem utilizar *templates* para PRS. Dessa forma, foram evitadas discrepâncias com relação à precisão, à cobertura e à *F-Measure*;
- c) **Ameaças à validade externa.** Refere-se às questões que afetam a habilidade de generalizar os resultados do estudo experimental piloto para um contexto mais amplo do que aquele selecionado para o estudo. Sendo assim, os fatores que podem ter influenciado os resultados deste experimento são (i) os sistemas de software utilizados nos experimentos, (ii) a qualidade dos requisitos após a atividade **Transformação em Texto Narrativo** e (iii) a quantidade de amostras (requisitos) utilizadas. Para mitigar essas possíveis ameaças, foram realizados outros estudos experimentais com diferentes sistemas de software e DRS para diminuir essas ameaças.

7.14 Considerações Finais

Este capítulo apresentou o planejamento e a execução de um estudo experimental piloto e mais cinco estudos experimentais que objetivaram verificar a efetividade e a eficiência da utilização de *templates* para PRS para a identificação de interesses de software. Os estudos experimentais utilizaram cinco sistemas de software desenvolvidos na academia (incluindo o piloto) e um sistema real. Assim, 154 requisitos funcionais e não-funcionais foram utilizados nos estudos experimentais. Destes, cerca de 94 requisitos foram padronizados no trabalho.

As principais conclusões obtidas a partir dos estudos experimentais é ObasCIId-Tool+ melhorar em eficiência e eficácia na identificação de interesses ao utilizar os *templates* para PRS, principalmente no que se refere à medida cobertura. Além disso, houve melhora significativa na medida precisão, quando comparada sem o uso de *templates* para PRS, mostrando que essa melhora foi, em alguns momentos, igual (mesmo com o uso de *templates* para PRS).

Conforme apresentado nos estudos experimentais, embora com o uso de *templates* para PRS, alguns interesses não foram identificados ou gerou falsos positivos. O que se consegue perceber, na maioria das vezes, é o ocorrido é por causa do catálogo de interesses utilizado nos estudos experimentais não possuir palavras-chave que contemplem os interesses. Assim, é necessário um catálogo de interesse mais completo para obter resultados melhores. Pode-se dizer que a atualização do catálogo de interesse com palavras-chave utilizadas nos padrões seria uma válida alternativa. Contudo, isso não foi feito no trabalho para não “enviesar” os resultados.

As principais conclusões obtidas a partir dos estudos experimentais é ObasCIId-Tool+, com a utilização dos *templates* para PRS, ser superior nos resultados quando comparada à identificação e classificação dos interesses nos requisitos de cada sistema de software analisado. Notou-se ainda que, apesar da, na maioria dos estudos experimentais, precisão aumentar, em alguns estudos esse valor manteve-se o mesmo dos valores sem *templates* para PRS, não comprometendo o resultado final das análises.

8 TRABALHOS RELACIONADOS

Neste trabalho, foram realizados dois MSL. No primeiro, sobre PRS, foram revisados 217 trabalhos para responder às oito questões de pesquisa. Após a aplicação dos critérios de exclusão e de leitura na íntegra de cada um para responder às questões de pesquisa, apenas 24 trabalhos foram selecionados.

Destes 24 trabalhos, alguns serviram para dar embasamento à pesquisa realizada. Em um desses trabalhos (BARCELOS; PENTEADO, 2017), foi apresentada uma abordagem para a elaboração de DRS no domínio de Sistemas de Informação com base em PRS. Para isso, foram elaborados padrões que auxiliam, principalmente, aos Engenheiros de Software menos experientes, fornecendo uma base para a elaboração de DRS mais completos e consistentes. No mesmo trabalho, foram utilizados diferentes PRS para atender, em sua maioria, os requisitos funcionais. Nesta dissertação, o enfoque foi sobre requisitos não-funcionais (Persistência, Segurança, Desempenho, Usabilidade e Portabilidade).

Para o desenvolvimento do *templates* para PRS de CRUD presente neste trabalho, foi proposta uma forma de escrevê-los para atender às operações básicas de quaisquer sistemas de software (*create, read, update e delete*). Com isso, nos trabalhos encontrados na execução do MSL sobre PRS, foi identificado o desenvolvimento de PRS para as operações básicas em sistemas de software, com o intuito de aprimorar as técnicas de reúso de PRS em requisitos de software (AHMAD *et al.*, 2018; BARCELOS; PENTEADO, 2017).

Com o desenvolvimento do *template* para PRS de Segurança neste trabalho, visou-se a manutenção da confiabilidade dos sistemas de software, visto que, para esses tipos de PRS, está presente na grande maioria dos requisitos, pois apresentam diversos domínios de sistemas de software, por tratar de confiabilidade desses sistemas (WEI *et al.*, 2013; HAGGE; LAPPE, 2005). Foram apontadas semelhanças e diferenças entre os métodos utilizados de diferentes formas para a criação do PRS de Segurança. Com isso, foi realizada uma sumarização dessas diferentes formas do PRS, dando origem ao *template* para PRS de Segurança utilizado nesta dissertação. O foco do PRS elaborado aos estudos experimentais são voltados para autenticação e autorização, seguindo o mesmo propósito dos demais (ZHANG *et al.*, 2020; ZAFAR *et al.*, 2014; POST *et al.*, 2012; NAVARRO *et al.*, 2006).

Ao tratar de desempenho, pode-se observar que está diretamente relacionado com (i) o tempo de resposta, verificar se os sistemas de software respondem a utilização dos usuário em tempo hábil, (ii) a confiabilidade, mostrando que o sistema de software executas as transações, por exemplo, de sistemas bancários, e (iii) após manutenção, os sistemas de software

continuaram com as características anteriores (AYSOLMAZ *et al.*, 2018; MIRANDA *et al.*, 2017; FEITELSON, 2015; VALLES-BARAJAS, 2007; NAVARRO *et al.*, 2006; DÍAZ *et al.*, 2004; SMITH; WILLIAMS, 2001). Pensando nisso, o *template* para PRS de Desempenho foi desenvolvido para garantir que esses processos tenham um *template* para PRS que possa ser genérico e assertivo ao ser escrito.

A usabilidade está presente em diferentes cenários de sistemas de software, sejam eles voltados para *web*, aplicativos móveis ou *desktops*. Porém, ao tentar sintetizar as ideias presentes nos vários conceitos (ISO/IEC 25000, 2014; NIELSEN, 1994), que caracterizam a usabilidade aos Engenheiros de Software, que podem se perder ao escrever requisitos para atender a usabilidade. Alguns autores apresentaram diferentes PRS genéricos para a usabilidade (VALLES-BARAJAS, 2007; JURISTO *et al.*, 2007; HAGGE; LAPPE, 2005). No entanto, nesta pesquisa, foi desenvolvido um *template* para PRS de Usabilidade para abordar alguns dos conceitos de usabilidade, com o intuito de ser fácil a sua utilização, visando cobrir os requisitos para um conceito de usabilidade.

Na maioria dos sistemas de software, a ideia de serem portáteis para outros sistemas operacionais ou arquiteturas robustas de servidores é uma realidade, pois com quantidade exponencial de usuários para acessar esses sistemas de software, em diferentes plataformas, tem-se proporções gigantescas (ISO/IEC 25000, 2014). Para isso, o PRS de Portabilidade fez com que a apresentação pelos (ABRAN *et al.*, 2013; RENAULT *et al.*, 2009; NAVARRO *et al.*, 2006) fosse sintetizada no *template* para PRS apresentado neste trabalho, uma vez que a ideia foi envolver os conceitos de portabilidade no *template* para PRS de Portabilidade.

Contudo, esses trabalhos, que apresentam os PRS de Persistência, de Segurança, de Desempenho, de Usabilidade e de Portabilidade, têm apenas os objetivos de (i) apresentar um PRS novo, (ii) mostrar que o reúso de PRS nos requisitos aumenta a produtividade na escrita e (iii) apresentar os benefícios da utilização daquele PRS desenvolvido. Neste trabalho, há uma diferença significativa, pois, conforme mostrado em diversos estudos encontrados no MSL, a reutilização de PRS traz benefícios no ciclo de desenvolvimento de software, principalmente no início na fase de requisitos de software. Neste trabalho, o diferencial dos demais é a utilização de *templates* para PRS para escrever DRS contribuiu para a melhoria da efetividade da identificação e classificação de interesses em sistemas de software.

O segundo MSL realizado neste trabalho foi sobre abordagens para EROA. Por meio da análise dos 58 estudos primários relacionados às abordagens para EROA, foram catalogadas 45

abordagens, das quais a maioria contempla as atividades de identificação e classificação de interesses e de representação de interesses. Essas abordagens foram categorizadas de acordo com vários critérios, tais como, tipos de atividades contempladas por elas e técnicas para identificação de interesses utilizadas (PARREIRA JÚNIOR; PENTEADO, 2014). Diversas abordagens sugerem a utilização de catálogos de requisitos não-funcionais, para auxiliar os Engenheiros de Software na realização da identificação de interesses e atividades de classificação (CYSNEIROS, 2016; CHUNG; DO PRADO LEITE, 2009; CHITCHYAN *et al.*, 2005; BOEHM; IN, 1996).

Em outras abordagens (SIOBHAN; ELISA, 2005; BANIASSAD; CLARKE, 2004; RASHID *et al.*, 2003), nenhum recurso, como catálogos, é fornecido para auxiliar os Engenheiros de Software durante a identificação e classificação de problemas. Em vez disso, eles apenas sugerem o uso de palavras-chave, previamente identificadas pelos Engenheiros de Software a partir do DRS, como entradas para as atividades de identificação e classificação de interesses.

Em outro trabalho (PARREIRA JÚNIOR; PENTEADO, 2016), foi apresentado um conjunto de diretrizes bem definidos para apoiar Engenheiros de Software nas atividades de identificação e de classificação de interesses, tais como, (i) uma ontologia para especificação de catálogos de interesses de software e (ii) uma abordagem para Identificação e Classificação de Interesses (ObasCIId - *Ontologically-based Concern Identification and Classification*). Essa abordagem foi avaliada com relação a outra abordagem para EROA, bem conhecida na literatura, e os resultados foram significativamente melhores em termos de cobertura e de precisão. Contudo, nenhuma dessas abordagens para EROA considerava a utilização de *templates* para PRS no processo de identificação e classificação de interesses em sistemas de software, potencializando a proposta deste trabalho.

9 AMEAÇAS À VALIDADE

É normal a existência de questões que possam impactar ou limitar a validade dos resultados obtidos ao longo dos estudos experimentais. Essas questões são denominadas ameaças à validade, que podem ser classificadas em quatro tipos principais (TRAVASSOS *et al.*, 2002):

- a) **Externa.** Uma ameaça à validade externa limita a capacidade de generalização dos resultados obtidos nos estudos experimentais para contextos fora do ambiente avaliado. A realização de somente o estudo experimental piloto não representa adequadamente que o uso de *templates* para PRS pode contribuir para a melhoria da efetividade da identificação e classificação de interesses em sistemas de software, comprometendo a generalização dos resultados. A aplicação dos outros estudos experimentais com diferentes sistemas de software pode amenizar essa ameaça;
- b) **Interna.** Uma ameaça à validade interna prejudica o conhecimento do grau em que os resultados da pesquisa refletem a realidade observada (relacionamento de causa e efeito). Neste trabalho, as ameaças internas são:

- Em relação à fase **Transformação do DRS em Texto Narrativo** da abordagem, pois alguns estariam em forma de casos de uso, possíveis erros de interpretação na transformação podem afetar a validade e as características dos DRS e interferir nos resultados de *ObasCId-Tool+*;
- Em relação à utilização dos *templates* para PRS nos requisitos funcionais e não-funcionais, os resultados obtidos podem ocasionar uma descaracterização dos requisitos originais;
- A utilização de somente uma ontologia e um apoio computacional foi desenvolvida utilizando essa ontologia, que pode afetar a identificação e classificação dos interesses;
- Requisitos mal escritos podem levar a diferentes interpretações, interferindo nos resultados obtidos nos estudos experimentais. Apesar de pesquisadores com *expertise* nos sistemas de software utilizados terem revisado os requisitos usados nos estudos experimentais, não é possível garantir que foram extraídas informações essenciais dos DRS originais;
- A tradução dos PRS e de seus elementos do Inglês para o Português do Brasil, pode ter ocasionado interpretações equivocadas de alguns termos da língua estrangeira.

Porém, estes termos e os *templates* para PRS presentes neste trabalho foram validados por outros pesquisadores com experiência em Engenharia de Requisitos;

- c) **Conclusão.** Uma ameaça à validade de conclusão afeta a habilidade de chegar a uma conclusão correta a respeito dos relacionamentos entre o tratamento e o resultado dos estudos experimentais. A falta de testes estatísticos no estudo experimental piloto compromete a obtenção de resultados conclusivos a respeito do funcionamento efetividade da utilização dos *templates* para PRS, porém apresenta resultados que instigaram a realização dos demais estudos experimentais. A análise conseguiu expressar que a utilização dos *templates* para PRS é efetiva para a identificação e classificação dos interesses presentes nos DRS;
- d) **Construção.** Uma ameaça à validade de construção está relacionada à validação do instrumento de pesquisa, definindo se o tratamento reflete, adequadamente, a causa e se os resultados refletem corretamente o efeito. Apesar da utilização de testes estáticos específicos para os estudos experimentais em Engenharia de Software, a interpretação pode ter sido diferente se fossem aplicados por especialistas em estatística. A alteração dos sistemas de software para condução do estudo sem e com o uso de *templates* para PRS pode não ter sido suficiente para evitar algum tipo de vício na obtenção dos requisitos funcionais e não-funcionais utilizados nos estudos experimentais.

10 CONSIDERAÇÕES FINAIS

Conforme comentado anteriormente, a utilização de *templates* para PRS para escrever DRS objetiva contribuir para a melhoria da efetividade da identificação e classificação de interesses em sistemas de software. Promove-se melhorias quanto à separação de interesses durante as fases iniciais do desenvolvimento de sistemas de software, oferecendo maior assertividade para a identificação e classificação de interesses e melhorando a modularização desses sistemas. Com essa melhoria, espera-se que, quando um interesse for modificado, a influência dessas modificações nos demais requisitos do software não acarrete perda de modularização (SAMPAIO *et al.*, 2007).

Dentre os vários PRS propostos nos últimos anos, pode-se notar que tais PRS foram concebidos basicamente de três maneiras: i) evoluindo PRS para ER tradicional, como PRS de segurança (WITHALL *et al.*, 2007) e portabilidade (ABRAN *et al.*, 2013); ii) propondo novos PRS, como para operações básicas, processamento de transação, informações gerenciais, regras de negócio em sistemas de informação (BARCELOS; PENTEADO, 2017), usabilidade (JURISTO *et al.*, 2007) e segurança (ZAFAR *et al.*, 2014); ou iii) adaptando outros PRS existentes para complementá-los ou minimizar algumas de suas limitações (ARORA *et al.*, 2015; MAIDEN *et al.*, 2005; TSENG; JIAO, 1997). Esses PRS foram utilizados para desenvolver os *templates* para PRS utilizados nos estudos experimentais e apresentados no Capítulo 4.

Apesar de existirem diversos PRS na literatura, há escassez de estudos para prover integração dos PRS com outras abordagens na ER, destacando a EROA e sua atividade de identificação e classificação de interesses, quais abordagens se utilizam para essa atividade com os requisitos do software, entre outros. Para atenuar essa deficiência, propôs-se desenvolver *templates* para PRS para escrever DRS que contribuem para a melhoria da efetividade da identificação e classificação de interesses em sistemas de software. Os detalhes desses *templates* para PRS e a forma de obtê-los/desenvolvê-los foram apresentados no Capítulo 4.

Associado ao problema relatado anteriormente, está o fato das abordagens para EROA existentes na literatura deixarem a desejar quanto aos recursos apropriados para apoiar Engenheiros de Software durante a atividade de identificação e classificação de interesses de software. Essa foi a segunda preocupação tratada nesta pesquisa, sendo contemplada em três partes: i) oferecer ao Engenheiro de Software um apoio computacional em que ele pudesse encontrar informações que os auxiliassem no processo de identificação e classificação de interesses a partir de requisitos de software; ii) utilizar *templates* para PRS para auxiliar na

escrita dos requisitos funcionais e não-funcionais dos sistemas de software para melhorar a identificação e classificação dos interesses; e iii) propor uma abordagem para padronizar os requisitos funcionais e não-funcionais, pois muitos deles são escritos em forma de casos de uso e o apoio computacional somente aceita em texto narrativo.

Para atender a primeira parte, foi desenvolvido o apoio computacional *ObasCId-Tool+* (evolução da *ObasCId-Tool*) para atender a abordagem para EROA (*ObasCId - Ontologically-based Concern Identification*) (PARREIRA JÚNIOR; PENTEADO, 2018), que oferece recursos para o Engenheiro de Software utilizar a abordagem com os *templates* para PRS nos requisitos do software, de forma sistemática e planejada para obter melhores resultados com a identificação e classificação de interesses. *ObasCId-Tool+* foi apresentado no Capítulo 6 e utilizado nos estudos experimentais no Capítulo 7.

Para isso, propôs-se o uso dos *templates* para PRS para tipos específicos de interesses de software, que consistem em instâncias dos conceitos e relacionamentos existentes literatura. No Capítulo 3, é apresentado o MSL sobre PRS, que busca exemplos de PRS elaborados a partir de fontes diversas, tais como, PRS de portabilidade (ABRAN *et al.*, 2013; RENAULT *et al.*, 2009; NAVARRO *et al.*, 2006) e a execução de estudos experimentais com projetos de software, em que a identificação e classificação de interesses havia sido concluída e em outros não. No Capítulo 4, é apresentada a abordagem que realiza a transformação dos requisitos em caso de uso para texto narrativo.

Contudo, somente inserir os DRS com e sem utilizar *templates* para PRS não é suficiente para aplicar os *templates* para PRS e verificar se apresentou melhorias. Assim, foi preciso aplicar a abordagem de transformação dos requisitos funcionais e não-funcionais, evitando que o uso de *templates* para PRS se limitasse a somente requisitos em texto narrativo. Com isso, *ObasCId-Tool+* e os *templates* para PRS podem ser aplicados em qualquer um dos requisitos de sistemas de software, conforme apresentado no Capítulo 5.

Para verificar se houve melhora na identificação e classificação dos interesses com o uso de *templates* para PRS em um estudo experimental piloto e mais outros cinco estudos experimentais, foram conduzidos para avaliar que a utilização de *templates* para PRS para escrever DRS pode contribuir para a melhoria da efetividade da identificação e classificação de interesses em sistemas de software. Como resultados, notou-se que o uso de *templates* para PRS pode contribuir significativamente para o aprimoramento da cobertura, precisão e *F-Measure* do processo de identificação e classificação de interesse a partir de requisitos de software, sua eficiência e eficácia.

10.1 Contribuições e Limitações

Com base no que foi exposto anteriormente, bem como nos demais capítulos desta dissertação, ressaltam-se como principais contribuições desta pesquisa:

- a) A investigação do estado da arte referente à PRS e o seu uso na ER, por meio do MSL conduzido neste trabalho;
- b) A investigação do estado da arte referente à EROA e as abordagens para EROA no contexto da ER, por meio do MSL conduzido neste trabalho;
- c) Os PRS, que provêm definição a respeito dos requisitos, destacando os principais conceitos e relacionamentos envolvidos neste PRS. A partir do entendimento e uso dos PRS, pode-se confeccionar *templates* para PRS para aumentar a quantidade de PRS que atende aos diferentes requisitos de software presentes nos sistemas de software acadêmicos ou industrial, sendo mais amplos e que podem ser úteis no processo de identificação e classificação de interesses;
- d) A abordagem para padronizar os requisitos de software, que provê recursos apropriados aos Engenheiros de Software para utilizar requisitos que não estejam em forma de texto narrativo no apoio computacional e realize a identificação e classificação dos interesses;
- e) O desenvolvimento da evolução da ferramenta computacional *ObasCId-Tool+*, que provê suporte adequado à utilização dos *templates* para PRS para e na geração e compartilhamento do conhecimento a respeito dos interesses de software, bem como para a identificação e classificação de interesses a partir de requisitos de software com e sem utilizar *templates* para PRS.

Como principais limitações dos produtos gerados nesta pesquisa, pode-se citar:

- a) A dependência da qualidade dos resultados do processo de identificação e classificação de interesses propostos pelo uso de *templates* para PRS necessita da existência de bons catálogos de interesses de software. Há algumas propostas de catálogos para requisitos não funcionais, porém eles não são totalmente adequados para aplicação no contexto da EROA;
- b) A qualidade dos resultados do processo de identificação e classificação de interesses com o uso de *templates* para PRS é afetada pelo correto entendimento do PRS e do requisito que será padronizado. Deve-se salientar que não adianta *templates* para PRS se o Engenheiro de Software não colocar as informações essenciais do requisito;
- c) A falta de estudos experimentais que comparem *ObasCId-Tool+* com o uso de *templates* para PRS com outras ferramentas/apoios computacionais para EROA propostas na literatura. Isso ocorreu por causa da escassez de ferramentas/apoios computacionais que

ofereçam apoio à atividade de identificação e classificação de interesses. Apenas a ferramenta computacional EA-Miner oferece esse suporte, mas ela não está disponível. A maioria das ferramentas/apoios computacionais propostas são para outras atividades da EROA, como representação e composição de interesses. Além disso, a maioria delas está indisponível ou obsoleta.

10.2 Trabalhos Futuros

A continuidade deste trabalho é necessária para possibilitar que ObasCID-Tool+ e o uso de PRS possa evoluir ainda mais, servindo como parte fundamental na identificação e classificação dos interesses. Como sugestão de trabalhos futuros, pode-se citar:

- a) A elaboração e inserção de novos *templates* para PRS na ObasCID-Tool+ para aumentar o reuso dos *templates* para PRS e, conseqüentemente, a identificação e classificação dos interesses;
- b) O uso mais ativo de ObasCID-Tool+ pode ser realizado para identificar possíveis melhorias quanto à funcionalidade e usabilidade;
- c) Outros estudos podem ser realizados para identificar e reusar *templates* para PRS não contemplados neste trabalho e especificados por outros autores;
- d) Elaboração de mais catálogos de interesses e palavras-chave que contemplem maior quantidade de interesses de software;
- e) Utilização de ObasCID-Tool+ em empresas para avaliar taxas de assertividade na identificação e classificação dos interesses, sua usabilidade, o uso dos *templates* propostos para PRS e a correta modularização dos sistemas desenvolvidos a partir do seu uso com os *templates* para PRS;
- f) A integração de ObasCID-Tool+ a outros ambientes existentes que tenham como uma de suas atividades a elaboração de DRS;
- g) Realização de novos estudos experimentais com ObasCID-Tool+, comparando-a com outras que implementam abordagens para EROA. Dentre os possíveis tipos de estudos experimentais que podem ser realizados com ObasCID-Tool+, destacam-se os testes para verificar sua efetividade em termos de cobertura, precisão e *F-Measure* de interesses, quando seus resultados são confrontados com o uso manual das abordagens para EROA, considerando-se DRS de diferentes tamanhos;

- h) Aplicar os elementos utilizados neste trabalho, tais como a abordagem, ObasCIId-Tool+ e os estudos experimentais em cenários de fábrica de software no contexto de métodos ágeis, assim como Scrum e Kanban.

REFERÊNCIAS

ABRAN, A.; AL-SARAYREH, K. T.; CUADRADO-GALLEGO, J. J. **A standards-based reference framework for system portability requirements**. *Computer Standards & Interfaces*, v. 35, n. 4, p. 380-395, 2013.

AHMAD, S.; ANUAR, U.; EMRAN, N. A. **A Tool-based Boilerplate Technique to Improve SRS Quality: An Evaluation**. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, v. 10, n. 2-7, p. 111-114, 2018.

ARORA, C. *et al.* **Automated checking of conformance to requirements templates using natural language processing**. *IEEE transactions on Software Engineering*, v. 41, n. 10, p. 944-968, 2015.

AYSOLMAZ, B. *et al.* **A semi-automated approach for generating natural language requirements documents based on business process models**. *Information and Software Technology*, v. 93, p. 14-29, 2018.

BANIASSAD, E.; CLARKE, S. **Theme: An approach for aspect-oriented analysis and design**. In: *Proceedings. 26th International Conference on Software Engineering*. IEEE, 2004. p. 158-167.

BARCELOS, L. V.; PENTEADO, R. A. D. **Elaboration of Software Requirements Documents by Means of Patterns Instantiation**. *Journal of Software Engineering Research and Development*, v. 5, n. 1, p. 3, 2017.

BATISTA, H. F. **Sistema Gerenciador de Atividades Curriculares - SISGAC**. Disponível em: <https://github.com/HemersonGH/SisGAC>. Acesso em: 03/ 03/ 2020.

BOEHM, B.; IN, H. **Identifying quality-requirement conflicts**. *IEEE software*, v. 13, n. 2, p. 25-35, 1996.

CASTRO, J.; KOLP, M.; MYLOPOULOS, J. **Towards requirements-driven information systems engineering: the Tropos project.** Information systems, v. 27, n. 6, p. 365-389, 2002.

CHAVEZ, C. *et al.* **Composing architectural aspects based on style semantics.** In: Proceedings of the 8th ACM international conference on Aspect-oriented software development. 2009. p. 111-122.

CHERNAK, Y. **Requirements reuse: the state of the practice.** In: 2012 IEEE International Conference on Software Science, Technology and Engineering. IEEE, 2012. p. 46-53.

CHITCHYAN, R. *et al.* **Report Synthesizing State-of-the-Art in Aspect-Oriented Requirements Engineering, Architectures and Design.** AOSD-Europe Deliverable D, 11, 1-259. 2005.

CHITCHYAN, R. *et al.* **A tool suite for aspect-oriented requirements engineering.** In: Proceedings of the 2006 international workshop on Early aspects at ICSE. 2006. p. 19-26.

CHUNG, L.; DO PRADO LEITE, J. C. S. **On non-functional requirements in software engineering.** In: Conceptual modeling: Foundations and applications. Springer, Berlin, Heidelberg, 2009. p. 363-379.

CYSNEIROS, L. M. **Catalogues on non-functional requirements.** 2016.

DÍAZ, I. *et al.* **A specification pattern for use cases.** Information & management, v. 41, n. 8, p. 961-975, 2004.

DE ALMEIDA BIOLCHINI, J. C. *et al.* **Scientific research ontology to support systematic review in software engineering.** Advanced Engineering Informatics, v. 21, n. 2, p. 133-151, 2007.

DOUG, R.; KENDALL, S. **Use Case Driven Object Modeling with UML: A Practical Approach.** 1999.

FEITELSON, D. G. **Workload modeling for computer systems performance evaluation**. Cambridge University Press, 2015.

GAY, D. *et al.* **The nesC language: A holistic approach to networked embedded systems**. *Acm Sigplan Notices*, v. 38, n. 5, p. 1-11, 2003.

GRUNDY, J. **Aspect-oriented requirements engineering for component-based software systems**. In: *Proceedings IEEE International Symposium on Requirements Engineering (Cat. No. PR00188)*. IEEE, 1999. p. 84-91.

HAGGE, L.; LAPPE, K. **Sharing requirements engineering experience using patterns**. *IEEE software*, v. 22, n. 1, p. 24-31, 2005.

HEALTH WATCHER. **Health Watcher** Disponível em: http://www.univasf.edu.br/~ricardo.aramos/disciplinas/ESI2009_2/exemplosDoc/Health_Watcher_Usecase.pdf. Último acesso em: março de 2020.

HERRERA, J. *et al.* **Revealing Crosscutting Concerns in Textual Requirements Documents: An Exploratory Study with Industry Systems**. In *Software Engineering (SBES), 2012 26th Brazilian Symposium on* (pp. 111-120).

ISO/IEC 25000 - **System and Software Quality Requirements and Evaluation (SQuaRE) series of standards**. Available in: <https://iso25000.com/index.php/en/iso-25000-standards?limit=4&limitstart=0>. Access in: July 15, 2020.

JACOBSON, I.; NG, P-W. **Aspect-oriented software development with use cases (addison-wesley object technology series)**. Addison-Wesley Professional, 2004.

JOANNA BRIGGS INSTITUTE. **Joanna Briggs Institute reviewers' manual: 2014 edition**. Australia: The Joanna Briggs Institute, 2014. In: Chapter Two: Qualitative protocol and title development. <https://nursing.lsuhsu.edu/JBI/docs/ReviewersManuals/ReviewersManual.pdf>.

JURISTO, N.; MORENO, A.; SANCHEZ-SEGURA, M-I. **Guidelines for eliciting usability functionalities**. *IEEE Transactions on Software Engineering*, v. 33, n. 11, p. 744-758, 2007.

KEELE, S. et al. **Guidelines for performing systematic literature reviews in software engineering**. Technical report, Ver. 2.3 EBSE Technical Report. EBSE, 2007.

KITCHENHAM, B. **Procedures for performing systematic reviews**. Keele, UK, Keele University, v. 33, n. 2004, p. 1-26, 2004.

KITCHENHAM, B. *et al.* **Systematic literature reviews in software engineering-a tertiary study**. Information and software technology, v. 52, n. 8, p. 792-805, 2010.

KONRAD, S.; CHENG, B. HC. **Requirements patterns for embedded systems**. In: Proceedings IEEE Joint International Conference on Requirements Engineering. IEEE, 2002. p. 127-136.

MAFRA, S. N.; BARCELOS, R. F; TRAVASSOS, G. H. **Aplicando uma metodologia baseada em evidência na definição de novas tecnologias de software**. In: Proceedings of the 20th Brazilian Symposium on Software Engineering (SBES 2006). 2006. p. 239-254.

MAIDEN, N. AM *et al.* **Generating requirements from systems models using patterns: a case study**. Requirements Engineering, v. 10, n. 4, p. 276-288, 2005.

MASSONI, T.; SOARES, S.; BORBA, P. **Requirements document health-watcher**. Tech. Rep., 2006, version 2.0, 2006.

MIRANDA, M. A. *et al.* **Domain-specific language for automatic generation of UML models**. IET Software, v. 12, n. 2, p. 129-135, 2017.

MONTGOMERY, Douglas C. **Design and analysis of experiments**. John wiley & sons, 2017.

MOREIRA, A.; RASHID, A.; ARAUJO, J. **Multi-dimensional separation of concerns in requirements engineering**. In: 13th IEEE International Conference on Requirements Engineering (RE'05). IEEE, 2005. p. 285-296.

MURDOCH UNIVERSITY. **Systematic Reviews - Research Guide**. 2020. <https://libguides.murdoch.edu.au/systematic/PICO>. Acesso em: 03/06/2018.

MUSSBACHER, G.; KIENZLE, J.; AMYOT, D. **Transformation of aspect-oriented requirements specifications for reactive systems into aspect-oriented design specifications**. In: 2011 Model-Driven Requirements Engineering Workshop. IEEE, 2011. p. 39-47.

NAVARRO, E. *et al.* **A metamodeling approach for requirements specification**. Journal of Computer Information Systems, v. 46, n. 5, p. 67-77, 2006.

NIELSEN, Jakob. **Usability engineering**. Morgan Kaufmann, 1994.

PALOMARES, C.; QUER, C.; FRANCH, X. **PABRE-Man: Management of a Requirement Patterns Catalogue**. In: Requirements Engineering Conference, 2011. p. 341-342.

PARREIRA JÚNIOR, P. A. **ObasCId: Uma Abordagem Ontologicamente Fundamentada para EROA**. 2015.

PARREIRA JÚNIOR, P. A.; PENTEADO, R. A. D. **Aspect-Oriented Requirements Engineering: A Systematic Mapping**. XVI ICEIS, Portugal: Lisboa, 2014.

PARREIRA JÚNIOR, P. A.; PENTEADO, R. A. D. **ObasCId (-Tool): An Ontologically Based Approach for Concern Identification and Classification and its Computational Support**. Journal of the Brazilian Computer Society, v. 24, n. 1, p. 3, 2018.

PARREIRA JÚNIOR, P. A.; PENTEADO, R. A. D. **ObasCId: an ontologically-based approach for concern identification and classification**. In: 2016 X Brazilian Symposium on Software Components, Architectures and Reuse (SBCARS). IEEE, 2016. p. 141-150.

PARREIRA JÚNIOR, P. A. **Fake News App**. Disponível em: <https://github.com/gcc132-mis/fakenewsapp>. Acesso em: 03/ 03/ 2020.

PETERSEN, K. *et al.* **Systematic mapping studies in software engineering**. In: 12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12. 2008. p. 1-10.

POST, A. *et al.* **Automotive behavioral requirements expressed in a specification pattern system: a case study at BOSCH**. Requirements Engineering, v. 17, n. 1, p. 19-33, 2012.

PRODANOV, C. C.; DE FREITAS, E. C. **Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico-2ª Edição**. Editora Feevale, 2013.

RAMOS, R. A.; DE CASTRO, J. B. **Avaliação de uma Metodologia de Medição da Qualidade em um Documento de Requisitos Orientado a Aspectos**. In: WER. 2005. p. 161-172.

REIS, T. PC; CASTRO, J. FB; OLSINA, L. A. **Medição de Qualidade de Aplicações Web na Fase de Requisitos**. Simpósio Brasileiro de Engenharia de Software, Gramado, RS, p. 162-174, 2002.

RENAULT, S. *et al.* **A Pattern-based Method for building Requirements Documents in Call-for-tender Processes**. International journal of computer science & applications, v. 6, n. 5, p. 175-202, 2009.

RASHID, A.; MOREIRA, A.; ARAÚJO, J. **Modularisation and composition of aspectual requirements**. In: Proceedings of the 2nd international conference on Aspect-oriented software development. 2003. p. 11-20.

ROBERTSON, S. **Requirements patterns via events/use cases**. In: Proceedings Pattern Languages of Programming. 1996.

SAMPAIO, A. *et al.* **EA-Miner: a tool for automating aspect-oriented requirements identification**. In: Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering. 2005. p. 352-355.

SAMPAIO, A. *et al.* **A Comparative Study of Aspect-Oriented Requirements Engineering Approaches**. In: International Symposium on Empirical Software Engineering and Measurement, 2007. (pp. 166-175).

SCHNEIDER, G.; WINTERS, J. P. **Applying use cases: a practical guide**. Pearson Education, 2001.

SIOBHAN, C.; ELISA, B. **Aspect-Oriented Analysis and Design: The Theme Approach**. 2005.

SMITH, C. U.; WILLIAMS, L. G. **Performance solutions: a practical guide to creating responsive, scalable software**. Reading: Addison-Wesley, 2002.

SOEIRO, E.; BRITO, I. S.; MOREIRA, A. **An XML-Based Language for Specification and Composition of Aspectual Concerns**. In: ICEIS (3). 2006. p. 410-419.

SOMMERVILLE, I. **Engenharia de Software**. 10^a. edição. Pearson Universidades, 768 p., 2019.

SPEM. **Software and Systems Process Engineering Meta-Model Specification**, OMG Object Management Group, OMG document number formal: 08-04-01, April 2008.

TERRA, V.; INOCENCIO, A. C.; COSTA, H. A. X.; PARREIRA JÚNIOR, P. A. **EasyHalf-Uma Aplicação Web para Aprimoramento do Processo de Obtenção de Descontos de Meia-entrada**. In: Anais do XXVI Workshop de Informática na Escola. SBC, 2020. p. 389-398.

TOVAL, A. *et al.* **Requirements reuse for improving information systems security: a practitioner's approach**. Requirements Engineering, v. 6, n. 4, p. 205-219, 2002.

TRAVASSOS, G. H.; GUROV, D.; AMARAL, EAG. **Introdução à engenharia de software experimental**. 2002. No prelo, 2017.

TSENG, M. M.; JIAO, J. **A variant approach to product definition by recognizing functional requirement patterns**. *Journal of Engineering Design*, v. 8, n. 4, p. 329-340, 1997.

VALLES-BARAJAS, F. **A requirements engineering process for control engineering software**. *Innovations in Systems and Software Engineering*, v. 3, n. 4, p. 217-227, 2007.

VERMA, K.; KASS, A. **Requirements analysis tool: A tool for automatically analyzing software requirements documents**. In: *International semantic web conference*. Springer, Berlin, Heidelberg, 2008. p. 751-763.

WEI, C.; XIAOHONG, B.; XUEFEI, L. **A study on airborne software safety requirements patterns**. In: *2013 IEEE Seventh International Conference on Software Security and Reliability Companion*. IEEE, 2013. p. 131-136.

WIERINGA, R. *et al.* **Requirements engineering paper classification and evaluation criteria: a proposal and a discussion**. *Requirements engineering*, v. 11, n. 1, p. 102-107, 2006.

WILCOXON, F.; KATTI, S. K.; WILCOX, R. A. **Critical values and probability levels for the Wilcoxon rank sum test and the Wilcoxon signed rank test**. *Selected tables in mathematical statistics*, v. 1, p. 171-259, 1970.

WITHALL, S. **Software Requirement Patterns**. Pearson Education, 2007.

WOHLIN, C. *et al.* **Experimentation in software engineering**. Springer Science & Business Media, 2012.

ZAFAR, Nauman *et al.* **System security requirements analysis: A smart grid case study**. *Systems Engineering*, v. 17, n. 1, p. 77-88, 2014.

ZHANG, X. *et al.* **Pattern-based software process modeling for dependability**. *Journal of Software: Evolution and Process*, p. e2262, 2020.

APÊNDICE A - ARTIGOS RESULTANTES DO MSL SOBRE PRS

Tabela 1.A - Artigos Resultantes do MSL sobre PRS.

#	Artigos
P1	Hagge, L.; Lappe, K. Sharing Requirements Engineering Experience Using Patterns. IEEE software, v. 22, n. 1, p. 24-31, 2005.
P2	Tseng, M. M.; Jiao, J. A Variant Approach to Product Definition by Recognizing Functional Requirement Patterns. Journal of Engineering Design, v. 8, n. 4, p. 329-340, 1997.
P3	Aysolmaz, B.; Leopold, H.; Reijers, H. A.; Demirörs, O. A Semi-Automated Approach For Generating Natural Language Requirements Documents Based on Business Process Models. Information and Software Technology, v. 93, p. 14-29, 2018.
P4	Ali, N.; Lai, R. A Method of Software Requirements Specification and Validation for Global Software Development. Requirements Engineering, v. 22, n. 2, p. 191-214, 2017.
P5	Teruel, M. A.; Navarro, E.; López-Jaquero, V.; Montero, F.; González, P. A Comprehensive Framework for Modeling Requirements of CSCW Systems. Journal of Software: Evolution and Process, v. 29, n. 5, p. e1858, 2017.
P6	Kamalrudin, M.; Hosking, J.; Grundy, J. MaramaAIC: Tool Support for Consistency Management and Validation of Requirements. Automated software engineering, v. 24, n. 1, p. 1-45, 2017.
P7	Arora, C.; Sabetzadeh, M.; Briand, L.; Zimmer, F. Automated Checking of Conformance to Requirements Templates Using Natural Language Processing. IEEE transactions on Software Engineering, v. 41, n. 10, p. 944-968, 2015.
P8	Zafar, N.; Arnautovic, E.; Diabat, A.; Svetinovic, D. System Security Requirements Analysis: A Smart Grid Case Study. Systems Engineering, v. 17, n. 1, p. 77-88, 2014.
P9	Abran, A.; Al-Sarayreh, K. T.; Cuadrado-Gallego, J. J. A Standards-Based Reference Framework for System Portability Requirements. Computer Standards & Interfaces, v. 35, n. 4, p. 380-395, 2013.
P10	Al Balushi, T. H.; Sampaio, P. R. Falcone; Loucopoulos, P. Eliciting and Prioritizing Quality Requirements Supported by Ontologies: A Case Study Using the ElicitO Framework and Tool. Expert Systems, v. 30, n. 2, p. 129-151, 2013.
P11	Renault, S.; Méndez-Bonilla, Ó.; Franch-Gutiérrez, J.; Quer, C. A Pattern-Based Method for Building Requirements Documents in Call-for-Tender Processes. International journal of computer science & applications, v. 6, n. 5, p. 175-202, 2009.
P12	Valles-Barajas, F. A Requirements Engineering Process for Control Engineering Software. Innovations in Systems and Software Engineering, v. 3, n. 4, p. 217-227, 2007.
P13	Juristo, N.; Moreno, A.; Sanchez-Segura, M.-I. Guidelines for Eliciting Usability Functionalities. IEEE Transactions on Software Engineering, v. 33, n. 11, p. 744-758, 2007.
P14	Navarro, E.; Letelier, P.; Mocholi, J. A.; Ramos, I. A Metamodeling Approach for Requirements Specification. Journal of Computer Information Systems, v. 46, n. 5, p. 67-77, 2006.
P15	Díaz, I.; Losavio, F.; Matteo, A.; Pastor, O. A Specification Pattern for Use Cases. Information & Management, v. 41, n. 8, p. 961-975, 2004.
P16	Miranda, M. A.; Ribeiro, M. G.; Marques-Neto, H. T.; Song, M. A. J. Domain-Specific Language for Automatic Generation of UML Models. IET Software, v. 12, n. 2, p. 129-135, 2017.
P17	Ahmad, S.; Anuar, U.; Emran, N. A. A Tool-Based Boilerplate Technique to Improve SRS Quality: An Evaluation. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), v. 10, n. 2-7, p. 111-114, 2018.
P18	Barcelos, L. V.; Penteado, R. D. Elaboration of Software Requirements Documents by Means of Patterns Instantiation. Journal of Software Engineering Research and Development, v. 5, n. 1, p. 3, 2017.
P19	Djouab, R.; Abran, A.; Seffah, A. An ASPIRE-Based Method for Quality Requirements Identification from Business Goals. Requirements Engineering, v. 21, n. 1, p. 87-106, 2016.
P20	Post, A.; Menzel, I.; Hoenicke, J.; Podelski, A. Automotive Behavioral Requirements Expressed in a Specification Pattern System: A Case Study at BOSCH. Requirements Engineering, v. 17, n. 1, p. 19-33, 2012.
P21	Meziane, F.; Athanasakis, N.; Ananiadou, S. Generating Natural Language Specifications from UML Class Diagrams. Requirements Engineering, v. 13, n. 1, p. 1-18, 2008.
P22	Maiden, N. A.; Manning, S.; Jones, S.; Greenwood, J. Generating Requirements from Systems Models Using Patterns: A Case Study. Requirements Engineering, v. 10, n. 4, p. 276-288, 2005.
P23	Hajri, I.; Goknil, A.; Briand, L. C.; Stephany, T. Configuring Use Case Models in Product Families. Software & Systems Modeling, v. 17, n. 3, p. 939-971, 2018.
P24	Zhang, X.; Wang, X.; Yun, W.; Gao, C.; Han, M.; Liu, H. Pattern-based software process modeling for dependability. Journal of Software: Evolution and Process, p. e2262, 2020.

Fonte: Do autor (2020).

APÊNDICE B - ARTIGOS RESULTANTES DO MSL SOBRE EROA

Tabela 1.B - Artigos Resultantes do MSL sobre EROA.

#	Artigos
P1	Amroune, M., Zarour, N., Laouar, M. R., Eom, S. B., Bendjenna, H. A Multi-Criteria Process to Resolve Conflict in the Composition of Aspectual Requirements. <i>Human Systems Management</i> , v. 33, n. 1-2, p. 27-34, 2014.
P2	Faßbender, S., Heisel, M., Meis, R. A Problem-, Quality-, and Aspect-Oriented Requirements Engineering Method. In: <i>International Conference on Software Technologies</i> . Springer, Cham, 2014, p. 291-310.
P3	Chernak, Y. Dataflow Modeling with Crosscutting Concerns and a Concept Lattice. <i>IEEE Software</i> , v. 31, n. 6, p. 70-78, 2014.
P4	Sardinha, A.; Chitchyan, R.; Weston, N.; Greenwood, P.; Rashid, A. EA-Analyzer: Automating Conflict Detection in a Large Set of Textual Aspect-Oriented Requirements. <i>Automated Software Engineering</i> , v. 20, n. 1, p. 111-135, 2013.
P5	Barra, E.; Morato, J. Early Knowledge Organization Assisted by Aspects. <i>Science of Computer Programming</i> , v. 121, p. 34-54, 2016.
P6	Zambrano, A.; Fabry, J.; Gordillo, S. Expressing Aspectual Interactions in Requirements Engineering: Experiences, Problems and Solutions. <i>Science of Computer Programming</i> , v. 78, n. 1, p. 65-92, 2012.
P7	Lee, J.; Hsu, K.-H. GEA: A Goal-Driven Approach to Discovering Early Aspects. <i>IEEE Transactions on Software Engineering</i> , n. 6, p. 1-1, 2014.
P8	He, C. GPRN: A Hierarchical Framework for Aspect-oriented Requirement Modeling. <i>Int. J. Digital Content Technology and its Applications</i> , v. 5, n. 2, p. 165-172, 2011.
P9	Gal-Chiș, C. E. N.; Pârv, B. Multicos - A Requirements Engineering Tool. <i>International Journal of Software Engineering and Knowledge Engineering</i> , v. 28, n. 01, p. 37-55, 2018.
P10	Parreira Júnior, P. A.; Penteado, R. A. D. Obascid (Tool): An Ontologically Based Approach for Concern Identification and Classification and its Computational Support. <i>Journal of the Brazilian Computer Society</i> , v. 24, n. 1, p. 3, 2018.
P11	Laurito, A. M. R.; Takada, S. Trade-off Analysis Between Concerns Based on Aspect-Oriented Requirements Engineering. <i>IEICE TRANSACTIONS on Information and Systems</i> , v. 95, n. 4, p. 1003-1011, 2012.
P12	Ali, B. S.; Kasirun, Z. M. 3CI: A Tool for Crosscutting Concern Identification. In: <i>Proceedings of the International Conference on Computational Intelligence for Modeling Control and Automation</i> , Vienna, Austria, pp. 351-355 (2008a).
P13	Duan, C.; Cleland-Huang, J. A Clustering Technique for Early Detection of Dominant and Recessive Crosscutting Concerns. In: <i>Proceedings of the International Conference on Software Engineering</i> , Minneapolis, MN (2007).
P14	Lau, Y.; Zhao, W.; Peng, X.; Tang, S. A Connector-Centric Approach to Aspect-Oriented Software Evolution. In: <i>Proceedings of the International Computer Software and Applications Conference</i> , Beijing, China, pp. 391-396 (2007).
P15	Weston, N.; Chitchyan, R.; Rashid, A. A Formal Approach to Semantic Composition of Aspect-Oriented Requirements. In: <i>Proceedings of the 16th IEEE International Requirements Engineering Conference</i> , Catalunya, Spain, pp. 173-182 (2008).
P16	Agostinho, S.; Moreira, A.; Marques, A.; Araújo, J.; Brito, I. S.; Ferreira, R.; Chevalley, P. A Metadata-Driven Approach for Aspect-Oriented Requirements Analysis. In: <i>Proceedings of the 10th International Conference on Enterprise Information Systems</i> , Barcelona, Spain, pp. 129-136 (2008).
P17	Medeiros, M.; Silva, L.; Medeiros, A. L. A Semi-Automatic Strategy to Identify Crosscutting Concerns in PI-Aov Graph Requirement Models. In: <i>CIbSE</i> . 2013. p. 285-298.
P18	Liu, X.; Liu, S.; Zheng, X. Adapting the NFR Framework to Aspectual Use-Case Driven Approach. In: <i>Proceedings of the 7th International Conference on Software Engineering Research, Management and Applications</i> , Hainan Island, China, pp. 209-214 (2009).
P19	Ali, B. S.; Kasirun, Z. M. D. An Approach for Crosscutting Concern Identification at Requirements Level Using NLP. <i>Int. J. Phy. Sci.</i> 6(11), 2718-2730 (2011).
P20	Wehrmeister, M. A.; Freitas, E. P.; Pereira, C. E.; Wagner, F. R. An Aspect-Oriented Approach for Dealing with Non-Functional Requirements in a Model-Driven Development of Distributed Embedded Real-Time Systems. In: <i>Proceedings of the 10th International Symposium on Object and Component-Oriented Real-Time Distributed Computing</i> , Orlando, Florida, USA, pp. 428-432 (2008).
P21	Araújo, J.; Zowghi, D.; Moreira, A. An Evolutionary Model of Requirements Correctness with Early Aspects. In: <i>Proceedings of the 9th International Workshop on Principles of Software Evolution</i> , Dubrovnik, Croatia, pp. 67-70 (2007).
P22	Soeiro, E.; Brito, I. S.; Moreira, A. An XML-based Language for Specification and Composition of Aspectual Concerns. In: <i>Proceedings of the 8th International Conference on Enterprise Information Systems</i> , Paphos, Cyprus (2006).

(continua)

Tabela 1.B - Artigos Resultantes do MSL sobre EROA (continuação).

#	Artigos
P23	Clarke, S.; Baniassad, E. Aspect-Oriented Analysis and Design: The Theme Approach. Addison-Wesley, New York (2005).
P24	Kienzle, J.; Abed, W. A.; Klein, J. Aspect-Oriented Multi-View Modeling. In: Proceedings of the 8th International Conference on AOSD, New York, USA, pp. 87-98 (2009).
P25	Ribeiro, J. C.; Araújo, J. AspOrAs: A Requirements Agile Approach Based on Scenarios and Aspects. In: Proceedings of the 2nd International Conference on Research Challenges in Information Science, Marrakech, Morocco, pp. 313-323 (2008).
P26	Ali, B. S.; Kasirun, Z. M. Developing Tool for Crosscutting Concern Identification Using NLP. In: Proceedings of the International Symposium on Information Technology, Kuala Lumpur, Malaysia (2008b).
P27	Sardinha, A.; Chitchyan, R.; Weston, N.; Greenwood, P.; Rashid, A. Ea-Analyzer: Automating Conflict Detection in Aspect-Oriented Requirements. In: Proceedings of the 24th International Conference on Automated Software Engineering, Auckland, New Zealand, pp. 530-534 (2009).
P28	Sampaio, A.; Chitchyan, R.; Rashid, A.; Rayson, P. Ea-Miner: A Tool for Automating Aspect-Oriented Requirements Identification. In: Proceedings of the International Conference Automated Software Engineering, California, USA, pp. 353-355 (2005).
P29	Rashid, A.; Sawyer, P.; Moreira, A.; Araújo, J. Early Aspects: A Model for Aspect-Oriented Requirements Engineering. In: International Conference on Requirements Engineering (RE) (2002).
P30	Katz, S.; Rashid, A. From Aspectual Requirements to Proof Obligations for Aspect-Oriented Systems. In: Proceedings of the IEEE International Conference on Requirements Engineering, Kyoto, Japan, pp. 48-57 (2004).
P31	Yijun, Y.; Leite, J. C. S. do P.; Mylopoulos, J. From Goals to Aspects: Discovering Aspects from Requirements Goal Models. In: International Conference on Requirements Engineering (RE) (2004).
P32	Penim, A. S.; Araújo, J. Identifying and Modeling Aspectual Scenarios with Theme and MATA. In: Proceedings of the ACM Symposium on Applied Computing, Switzerland, pp. 287-291 (2010).
P33	Marques, G.; Araújo, J.; Lencastre, M. Integrating Problem Frames with Aspects. In: Proceedings of the 23rd Brazilian Symposium on Software Engineering, Fortaleza/CE, pp. 196-206 (2009).
P34	Mehner, K.; Monga, M.; Taentzer, G. Interaction Analysis in Aspect-Oriented Models. In: Proceedings of the 14th IEEE International Conference Requirements Engineering, Minnesota, USA, pp. 69-78 (2006).
P35	Kit, L. K.; Man, C. K.; Baniassad, E. Isolating and Relating Concerns in Requirements Using Latent Semantic Analysis. ACM SIGPLAN Not. 41(10), 383-396 (2006).
P36	Meier, S.; Reinhard, T.; Stoiber, R.; Glinz, M. Modeling and Evolving Crosscutting Concerns in ADORA. In: Proceedings of the International Conference on Software Engineering, Minneapolis, MN (2007).
P37	Rashid, A.; Moreira, A.; Araújo, J. Modularisation and Composition of Aspectual Requirements. In: 2nd International Conference on Aspect-Oriented Software Development (AOSD 2003). ACM (2003).
P38	Marques, A.; Moreira, A.; Araújo, J. Multi-Dimensional Composition by Objective. In: Proceedings of the International Conference on Software Engineering, Leipzig, Germany, pp. 19-25 (2008).
P39	Moreira, A.; Rashid, A.; Araújo, J. Multi-Dimensional Separation of Concerns in Requirements Engineering. In: Proceedings of the 13th International Conference on Requirements Engineering, pp. 285-296 (2005).
P40	Mussbacher, G.; Amyot, D. On Modeling Interactions of Early Aspects with Goals. In: Proceedings of the Workshop on Aspect-Oriented Requirements Engineering and Architecture Design, Charlottesville, VA, USA, pp. 14-19 (2009).
P41	Hamza, H. S.; Darwish, D. On the Discovery of Candidate Aspects in Software Requirements. In: Proceedings of the 6th International Conference on Information Technology: New Generations, Las Vegas, USA, pp. 819-824 (2009).
P42	Pinto, M.; Fuentes, L.; Valenzuela, J. A.; Pires, P. F.; Delicato, F. C. Promoting the Software Evolution in AOSD with Early Aspects: Architecture-Oriented Model-Based Pointcuts. In: Proceedings of the Workshop on Aspect-Oriented Requirements Engineering and Architecture Design, Charlottesville, VA, USA, pp. 31-37 (2009).
P43	Chernak, Y. Requirements Composition Table Explained. In: Proceedings of the 20th IEEE International Requirements Engineering Conference, Chicago, Illinois, USA, pp. 273-278 (2012).
P44	Mussbacher, G.; Amyot, D.; Araújo, J.; Moreira, A. Requirements Modeling with the Aspect-Oriented User Requirements Notation (AoURN): A Case Study. In: Katz, S., Mezini, M., Kienzle, J. (eds.) Transactions on Aspect-Oriented Software Development VII. LNCS, vol. 6210, pp. 23-68. Springer, Heidelberg (2010).
P45	Paula, V. D.; Batista, T. Revisiting a Formal Framework for Modeling Aspects in the Design Phase. In: Proceedings of the International Conference on Software Engineering, Minneapolis, MN (2007).
P46	Whittle, J.; Araújo, J. Scenario modeling with aspects. IEE Proceedings-Software, v. 151, n. 4, p. 157-171, 2004.
P47	Chitchyan, R.; Greenwood, P.; Sampaio, A.; Rashid, A.; Garcia, A.; Fernandes da Silva, L. Semantic vs. Syntactic Compositions in Aspect-Oriented Requirements Engineering: An Empirical Study. In: Proceedings of the 8th International Conference on AOSD, Virginia, USA, pp. 149-160 (2009).

(continua)

Tabela 1.B - Artigos Resultantes do MSL sobre EROA (continuação).

#	Artigos
P48	Mussbacher, G.; Whittle, J.; Amyot, D. Semantic-Based Interaction Detection in Aspect-Oriented Scenarios. In: Proceedings of the IEEE International Conference on Requirements Engineering, Georgia, USA, pp. 203-212 (2009).
P49	Oliveira, A. R.; Araújo, J.; Amaral, V. The VisualAORE DSL. In: Proceedings of the 5th International Workshop on Requirements Engineering Visualization, Sydney, Australia, pp. 11-19 (2010).
P50	Baniassad, E.; Clarke, S. Theme: An Approach for Aspect-Oriented Analysis and Design. In: 26th International Conference on Software Engineering (ICSE 2004) (2004).
P51	Monteiro, M. P.; Fernandes, J. M. Towards a Catalogue of Refactorings and Code Smells for AspectJ. In: Rashid, A., Akşit, M. (eds.) Transactions on Aspect-Oriented Software Development I. LNCS, vol. 3880, pp. 214-258. Springer, Heidelberg (2006). Zheng, X., Liu, X., Liu, S.: Use case and non-functional scenario template-based approach to identify aspects. In: Proceedings of the 2nd International Conference on Computer Engineering and Applications, Bali Island, Indonesia, pp. 89-93 (2010).
P52	Brito, I.; Moreira, A. Towards a Composition Process for Aspect-Oriented Requirements. In: Proceedings of the Early Aspects Workshop at AOSD, Massachusetts, USA (2003).
P53	Araújo, J.; Ribeiro, J. C. Towards an Aspect-Oriented Agile Requirements Approach. In: Proceedings of the International Workshop on Principles of Software Evolution, Lisbon, Portugal, pp. 140-143 (2005).
P54	Alencar, F.; Castro, J.; Lucena, M.; Santos, E.; Silva, C.; Araújo, J.; Moreira, A. Towards Modular I* Models. In: ACM Symposium on Applied Computing, pp. 292-297 (2010).
P55	Pinto, M.; Gamez, N.; Fuentes, L. Towards the Architectural Definition of the Health Watcher System with AO-ADL. In: Proceedings of the Workshop in Aspect-Oriented Requirements Engineering and Architecture Design, Minneapolis (2007).
P56	Mussbacher, G.; Kienzle, J.; Amyot, D. Transformation of Aspect-Oriented Requirements Specifications for Reactive Systems into Aspect-Oriented Design Specifications. In: Proceedings of the Model-Driven Requirements Engineering Workshop, Trento, Italy, pp. 39-47 (2011).
P57	Ossher, H.; Amid, D.; Anaby-Tavor, A.; Bellamy, R.; Callery, M.; Desmond, M.; Swart, C. Using Tagging to Identify and Organize Concerns During Pre-Requirements Analysis. In: Proceedings of the Aspect-Oriented Requirements Engineering and Architecture Design, Charlottesville, VA, USA, pp. 25-30 (2009).
P58	Mussbacher, G.; Amyot, D.; Weiss, M. Visualizing Aspect-Oriented Requirements Scenarios with Use Case Maps. In: Proceedings of the First International Workshop on Visualization in Requirements Engineering (2007).

Fonte: Do autor (2020).

APÊNDICE C - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE HEALTH WATCHER

Tabela 1.C - Requisitos Funcionais e Não-Funcionais do Sistema de Software Health Watcher.

Requisitos Funcionais e Não-Funcionais
[RF01] Consultar Informações
<p>Este caso de uso tem como propósito possibilitar as consultas para o cidadão.</p> <p>Consultar Guia de Saúde</p> <p>O cidadão poderá solicitar:</p> <ul style="list-style-type: none"> • Quais as unidades de saúde que atendem determinada especialidade; • Quais as especialidades de uma unidade de saúde. <p>Consultar Informações Diversas</p> <p>O cidadão poderá solicitar:</p> <ul style="list-style-type: none"> ➤ Informações sobre a queixa feita pelo cidadão: <ul style="list-style-type: none"> ✓ Especificação da queixa; ✓ Situação (ABERTA, SUSPensa ou FECHADA); ✓ Parecer técnico; ✓ Data do parecer; ✓ Funcionário que realizou o parecer; ➤ Informações sobre doenças: <ul style="list-style-type: none"> ✓ O que é; ✓ Sintomas; ✓ Manifestação; ✓ Tempo de duração. <p>Prioridade: Importante.</p> <p>Entradas e pré-condições:</p> <ul style="list-style-type: none"> • Os dados a serem consultados devem estar cadastrados no sistema. <p>Saídas e pós-condições:</p> <ul style="list-style-type: none"> • Resultado da consulta do cidadão. <p>Fluxo de eventos principal</p> <ol style="list-style-type: none"> 1. O cidadão escolhe o tipo de consulta; 2. O cidadão informa dados para consulta; <ol style="list-style-type: none"> 2.1. Caso seja consulta de especialidades agrupadas por unidade de saúde, o cidadão seleciona a unidade de saúde a ser pesquisada; 2.2. Caso seja consulta de unidades de saúde agrupadas por especialidades, o cidadão seleciona a especialidade a ser pesquisada; 2.3. Caso seja consulta de queixas, o cidadão entra com o código da queixa; 2.4. Caso seja consulta a doenças, o cidadão seleciona a doença a ser pesquisada; 3. O sistema apresenta o resultado da consulta.

(continua)

Tabela 1.C - Requisitos Funcionais e Não-Funcionais do Sistema de Software Health Watcher (continuação).

Requisitos Funcionais e Não-Funcionais
[RF02] Especificar Queixa
<p>Este caso de uso tem como propósito o registro de queixas. As queixas podem ser:</p> <p>Queixa Animal - DVA</p> <ul style="list-style-type: none"> • Casos de apreensão de animais; • Controle de vetores e animais sinantrópicos (roedores, escorpiões, morcegos, etc.); • Doenças associadas ao pernilongo (dengue, filariose); • Maus tratos com animais. <p>Queixa Alimentar - DVISA</p> <ul style="list-style-type: none"> • Casos de suspeita por ingestão de alimentos estragados. <p>Queixa diversa - DVISA</p> <ul style="list-style-type: none"> • Casos relacionados a diversos motivos, motivos estes que não têm ligações com as queixas citadas anteriormente (restaurante suspeito quanto à higiene, fossas a céu aberto, carros pipas de procedimento suspeito, etc.); <p>Os três tipos de queixa têm as seguintes informações em comum:</p> <ul style="list-style-type: none"> • Dados da queixa: descrição (obrigatório) e observações (opcional); • Dados do reclamante: nome, rua, complemento, bairro, cidade, estado, cep, número do telefone e e-mail. Todas estas informações são opcionais; • Situação da queixa (obrigatório), que pode ser: ABERTA, SUSPENSA ou FECHADA. No registro da queixa a sua situação deve ser ABERTA; • O sistema deve registrar a data de registro da queixa. <p>Além destas cada queixa tem suas informações específicas. São elas:</p> <p>Queixa Animal - DVA</p> <ul style="list-style-type: none"> • Tipo de animal (obrigatório), quantidade de animais (obrigatório), data do incômodo (obrigatório); • Rua, complemento, bairro, cidade, estado, cep e número do telefone do local de ocorrência. Todas estas informações são opcionais. <p>Queixa Alimentar - DVISA</p> <ul style="list-style-type: none"> • Nome da vítima (obrigatório); • Rua, complemento, bairro, cidade (ou cidade mais próxima), estado, cep e número do telefone da vítima. Todas opcionais; • Quantidade de comensais (pessoas que comeram a comida), quantidade de doentes, número de pessoas internadas e número de óbitos. Todos obrigatórios; • Local em que os pacientes foram atendidos, refeição suspeita. Todas opcionais. <p>Queixa diversa - DVISA</p> <ul style="list-style-type: none"> • Idade (obrigatório), escolaridade (opcional) e ocupação (opcional); • Rua, complemento, bairro, cidade, estado, cep e número do telefone do local mais próximo da ocorrência da queixa. Todas estas informações são opcionais. <p>Prioridade: Essencial.</p> <p>Entradas e pré-condições:</p> <ul style="list-style-type: none"> • Nenhuma. <p>Saídas e pós-condições:</p> <ul style="list-style-type: none"> • A queixa deve estar registrada no sistema. <p>Fluxo de eventos principal</p> <ol style="list-style-type: none"> 6. O cidadão informa o tipo de queixa; 7. O sistema registra o tipo, a data/ hora do atendimento; 8. O sistema apresenta a tela específica para cada tipo de queixa; 9. O cidadão informa os dados; 10. O sistema registra a queixa (com a situação ABERTA), retornando o número da ocorrência para que o cidadão anote e possa acompanhar o andamento da mesma. <p>O funcionário necessita do login do sistema para acessar os casos de uso, que são os seguintes:</p>

(continua)

Tabela 1.C - Requisitos Funcionais e Não-Funcionais do Sistema de Software Health Watcher (continuação).

Requisitos Funcionais e Não-Funcionais
[RF10] Login
<p>Este caso de uso tem como propósito permitir o acesso do funcionário a operações restritas no sistema Health-Watcher.</p> <p>Prioridade: Essencial.</p> <p>Entradas e pré-condições:</p> <ul style="list-style-type: none"> • Nenhuma. <p>Saídas e pós-condições:</p> <ul style="list-style-type: none"> • Senha validada pelo sistema. <p>Fluxo de eventos principal</p> <ul style="list-style-type: none"> • O funcionário entra com o login e senha; • O sistema verifica a senha digitada. <p>Fluxo alternativo</p> <p>No passo 2 caso a senha ou funcionário não seja válido, mostrar uma mensagem de erro.</p>
[RF11] Cadastrar Tabelas
<p>Este caso de uso tem como propósito o cadastramento das tabelas do sistema. Para este caso de uso estarão previstas as operações de inclusão, alteração, exclusão, consulta e impressão.</p> <p>As tabelas são as seguintes:</p> <ul style="list-style-type: none"> • Unidade de saúde (código da unidade, descrição da unidade); • Especialidade (código e descrição); • Unidade de saúde/ Especialidade (unidade de saúde e especialidade); • Funcionário (login, nome e senha); • Tipo de doença (código, nome, descrição, manifestação e duração); • Sintoma (código e descrição); • Tipo de doença/ Sintoma (tipo de doença e sintoma). <p>Prioridade: Essencial.</p> <p>Entradas e pré-condições:</p> <ul style="list-style-type: none"> • Funcionário logado no sistema. <p>Saídas e pós-condições:</p> <ul style="list-style-type: none"> • Dados atualizados nas tabelas. <p>Fluxo de eventos principal</p> <ol style="list-style-type: none"> 2. O funcionário escolhe a opção para cadastrar (inserir/atualizar) uma das tabelas; 3. O funcionário informa os dados; 4. O sistema registra a informação.
[RF12] Atualizar Queixa
<p>Este caso de uso tem como propósito realizar a atualização do andamento de uma queixa. Prioridade: Essencial.</p> <p>Entradas e pré-condições:</p> <ul style="list-style-type: none"> • A queixa deve estar cadastrada e com a situação ABERTA; • Funcionário logado no sistema. <p>Saídas e pós-condições:</p> <ul style="list-style-type: none"> • Queixa atualizada no sistema e com a sua situação FECHADA. <p>Fluxo de eventos principal</p> <ol style="list-style-type: none"> 1. O funcionário ativa opção para atualizar queixa; 2. O sistema solicita o número da queixa; 3. O sistema mostra o código da queixa e seu tipo; 4. O funcionário dá o parecer da queixa; 5. O sistema atualiza o parecer da queixa e a informação do funcionário que deu o parecer.

(continua)

Tabela 1.C - Requisitos Funcionais e Não-Funcionais do Sistema de Software Health Watcher (continuação).

Requisitos Funcionais e Não-Funcionais
[RF13] Cadastrar Novo Funcionário
<p>Este caso de uso tem como propósito permitir o cadastramento de novos funcionários no sistema.</p> <p>Prioridade: Essencial</p> <p>Entradas e pré-condições:</p> <ul style="list-style-type: none"> • Funcionário logado no sistema. <p>Saídas e pós-condições:</p> <ul style="list-style-type: none"> • Novo funcionário cadastrado no sistema. <p>Fluxo de eventos principal</p> <ol style="list-style-type: none"> 5. O funcionário entra com as seguintes informações do novo funcionário: <ul style="list-style-type: none"> ✓ Nome; ✓ Login. 6. Senha (com campo especial para confirmação); 7. O funcionário confirma a inserção; 8. O sistema cadastra os dados do novo funcionário. <p>Fluxo alternativo</p> <p>No passo 2 caso algum dos dados não tenha sido informado, mostrar uma mensagem de erro.</p>
[RF14] Atualizar Funcionário
<p>Este caso de uso tem como propósito permitir a atualização de dados do funcionário no sistema.</p> <p>Prioridade: Essencial.</p> <p>Entradas e pré-condições:</p> <ul style="list-style-type: none"> • Funcionário logado no sistema. <p>Saídas e pós-condições:</p> <ul style="list-style-type: none"> • Informações do funcionário atualizadas no sistema. <p>Fluxo de eventos principal</p> <ol style="list-style-type: none"> 2. O funcionário ativa a opção atualizar funcionário; 3. O funcionário entra com as suas informações para atualização: <ul style="list-style-type: none"> ✓ Nome; ✓ Nova senha (com campo extra para confirmação); ✓ Senha atual; 4. O funcionário confirma a atualização; 5. O sistema atualiza os dados do funcionário. <p>Fluxo alternativo</p> <p>No passo 3 caso algum dos dados não tenha sido informado o nome ou a senha atual, mostrar uma mensagem de erro.</p>
[RF15] Atualizar Unidade de Saúde
<p>Este caso de uso tem como propósito permitir a atualização de dados da unidade de saúde no sistema.</p> <p>Prioridade: Essencial.</p> <p>Entradas e pré-condições:</p> <ul style="list-style-type: none"> • Funcionário logado no sistema. <p>Saídas e pós-condições:</p> <ul style="list-style-type: none"> • Informações da unidade de saúde atualizadas no sistema. <p>Fluxo de eventos principal</p> <ol style="list-style-type: none"> 1. O funcionário ativa a opção atualizar unidade de saúde; 2. O funcionário entra com o código da unidade de saúde a ser atualizada; 3. O sistema mostra os dados da unidade de saúde; 4. O funcionário modifica os dados necessários; 5. O funcionário confirma a alteração; 6. O sistema atualiza os dados da unidade de saúde. <p>Fluxo alternativo</p> <p>Nos passos 2 e 3, caso algum dos dados não tenha sido informado, mostrar uma mensagem de erro.</p>

(continua)

Tabela 1.C - Requisitos Funcionais e Não-Funcionais do Sistema de Software Health Watcher (continuação).

Requisitos Funcionais e Não-Funcionais
[RF16] Mudar Funcionário Logado
<p>Este caso de uso tem como propósito permitir que outro funcionário se “logue” no sistema.</p> <p>Prioridade: Essencial.</p> <p>Entradas e pré-condições:</p> <ul style="list-style-type: none"> • Funcionário logado no sistema. <p>Saídas e pós-condições:</p> <ul style="list-style-type: none"> • Primeiro funcionário deslogado e novo funcionário logado no sistema. <p>Fluxo de eventos principal</p> <ol style="list-style-type: none"> 1. O funcionário ativa a opção mudar funcionário logado; 2. O sistema exibe a tela de login. A partir daqui o fluxo principal e secundários seguem o mesmo fluxo do [RF10] Login; <p>Fluxo alternativo</p> <p>No passo 2 caso a senha ou funcionário não seja válido, mostrar uma mensagem de erro.</p>
[RNF01] Usabilidade
<p>O sistema deve ter uma interface de fácil utilização, visto que o sistema pode ser utilizado por qualquer pessoa que tem acesso à Internet.</p> <p>O sistema deve ter um HELP on-line para ser consultado por qualquer pessoa que acesse o sistema.</p> <p>Prioridade: Importante.</p>
[RNF02] Confiabilidade
<p>O sistema deve estar disponível 24 horas por dia durante os 7 dias da semana. Por não ser um sistema crítico, o sistema poderá ficar fora do ar até que seja corrigida alguma falha que possa ocorrer.</p> <p>Prioridade: Importante.</p>
[RNF03] Desempenho
<p>O sistema deve prover acesso a 20 usuários simultaneamente.</p> <p>O tempo de resposta não deve ultrapassar 05 segundos por acesso.</p> <p>Prioridade: Essencial.</p>
[RNF04] Segurança
<p>O sistema deve utilizar algum protocolo de segurança para envio de dados pela Internet.</p> <p>Para ter acesso aos recursos de registro das queixas, o usuário deve estar habilitado pelo controle de acesso ao sistema.</p> <p>Prioridade: Importante.</p>
[RNF05] Padrões
<p>O sistema deve ser desenvolvido dentro dos padrões estabelecidos pela Emprel, responsável pelas normas de padronização de sistemas da Prefeitura da Cidade do Recife.</p> <p>Prioridade: Importante.</p>
[RNF06] Portabilidade
<p>SOFTWARE</p> <ul style="list-style-type: none"> • Uma Licença de uso do Microsoft Windows para a estação de trabalho <p>HARDWARE</p> <ul style="list-style-type: none"> • Um microcomputador Pentium III, com 256 Megas de Memória, Placa de rede 3Com 10/100. Este equipamento será utilizado como estação de trabalho para a atendente. <p>Prioridade: Importante.</p>
[RNF07] Distribuição
<p>O sistema deve ser capaz de ser executado em máquinas diferentes. Por exemplo, o núcleo do sistema pode estar sendo executado em uma máquina e os servlets em outra.</p> <p>Prioridade: Importante.</p>

(continua)

Tabela 1.C - Requisitos Funcionais e Não-Funcionais do Sistema de Software Health Watcher (continuação).

Requisitos Funcionais e Não-Funcionais
[RNF08] Interface com o Usuário
A interface com o usuário deve ser implementada em servlets. Prioridade: Essencial.
[RNF09] Meio de Armazenamento
O sistema deve ser passível de extensão no que se refere ao meio de armazenamento, podendo, por exemplo, serem usados arrays ou diferentes bancos de dados (MySQL, Oracle, etc.). Prioridade: Essencial.
[RNF10] Manipulação de Erros e Exceções
<ul style="list-style-type: none"> • Várias funcionalidades podem disparar erros enquanto o usuário interage com o sistema e requer diferentes técnicas de manipulação. • Erros gerais que se aplicam à maioria dos casos são devido à falta de informação e o sistema envia um sinal do erro e quais campos devem ser preenchidos. • Outros erros podem estar relacionados com entradas inválidas de dados e o mecanismo de manipulação de erros deve tentar evitá-los ou levantar o erro e sugerir a correção. Prioridade: Importante.

Fonte: Do autor (2020).

APÊNDICE D - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE HEALTH WATCHER EM TEXTO NARRATIVO

Tabela 1.D - Requisitos Funcionais e Não-Funcionais do Sistema de Software Health Watcher em Texto Narrativo.

Requisitos Funcionais e Não-Funcionais
[RF01] Consultar Informações
<p>Tem como propósito possibilitar as consultas para o cidadão. O cidadão poderá solicitar: (i) quais unidades de saúde atendem determinada especialidade; (ii) quais são as especialidades de uma unidade de saúde; (iii) informações sobre a queixa feita pelo cidadão, como especificação da queixa, situação (ABERTA, SUSPensa ou FECHADA), parecer técnico, data e hora do parecer e funcionário que realizou o parecer; e (iv) informações sobre as doenças, tais como descrição da doença, quais são os sintomas, qual é a forma de manifestação, qual é o tempo de duração e os locais de referência para tratamento da doença. O cidadão escolhe o tipo de consulta, informa dados para consulta e caso a consulta seja de especialidades agrupadas por unidade de saúde, o cidadão seleciona a unidade de saúde a ser pesquisada. Caso seja consulta de unidades de saúde agrupadas por especialidades, o cidadão seleciona a especialidade a ser pesquisada. Caso seja consulta de queixas, o cidadão entra com o código da queixa e/ou caso seja consulta a doenças, o cidadão seleciona a doença a ser pesquisada, assim o sistema apresenta o resultado da consulta.</p>
[RF02] Especificar Queixa
<p>Tem como propósito o registro de queixas. As queixas podem ser: (i) Queixa Animal: casos de apreensão de animais, controle de vetores e animais sinantrópicos (roedores, escorpiões, morcegos, etc.), doenças associadas ao pernilongo (dengue, filariose), maus tratos com animais; (ii) Queixa Alimentar: casos de suspeita por ingestão de alimentos estragados; e (iii) Queixa Diversa: casos relacionados a diversos motivos, motivos estes que não têm ligações com as queixas citadas anteriormente (restaurante suspeito quanto à higiene, fossas a céu aberto, carros pipas de procedimento suspeito, etc). Os três tipos de queixa têm as seguintes informações em comum: (i) Dados da queixa: descrição (obrigatório) e observações (opcional); (ii) Dados do reclamante: nome, rua, complemento, bairro, cidade, estado, CEP, número do telefone e e-mail. Todas estas informações são opcionais; e (iii) Situação da queixa (obrigatório), que pode ser: ABERTA, SUSPensa ou FECHADA. No registro da queixa a sua situação deve ser ABERTA. O sistema deve registrar a data de registro da queixa. Além desta informação, cada queixa tem suas informações específicas. São elas: (i) Queixa Animal: tipo de animal (obrigatório), quantidade de animais (obrigatório) e data do incômodo (obrigatório); rua, complemento, bairro, cidade, estado, CEP e número do telefone do local de ocorrência (todas estas informações são opcionais); (ii) Queixa Alimentar: nome da vítima (obrigatório); rua, complemento, bairro, cidade, estado, CEP e número do telefone da vítima (todas opcionais); quantidade de comensais (pessoas que comeram a comida), quantidade de doentes, número de pessoas internadas e número de óbitos (todos obrigatórios); local em que os pacientes foram atendidos e refeição suspeita (todas opcionais); e (iii) Queixa Diversa: idade (obrigatório), escolaridade (opcional) e ocupação (opcional); rua, complemento, bairro, cidade, estado, CEP e número do telefone.</p>
[RF10] Login
<p>Tem como propósito permitir o acesso do funcionário a operações restritas no sistema Health Watcher. A senha será validada pelo sistema, onde o funcionário entra com o login e senha é realizado a verificação da senha digitada e caso a senha ou funcionário não seja válido, mostrar uma mensagem de erro.</p>
[RF11] Cadastrar Tabelas
<p>Tem como propósito o cadastramento das tabelas do sistema. Para este requisito estão previstas as operações de inclusão, alteração, exclusão, consulta e impressão. O funcionário deve estar logado no sistema. As tabelas são as seguintes: (i) Unidade de saúde (código da unidade, descrição da unidade); (ii) Especialidade (código e descrição); (iii) Unidade de saúde/ Especialidade (unidade de saúde e especialidade); (iv) Funcionário (login, nome e senha); (v) Tipo de doença (código, nome, descrição, manifestação e duração); (vi) Sintoma (código e descrição); e (vii) Tipo de doença / Sintoma (tipo de doença e sintoma).</p>

(continua)

Tabela 1.D - Requisitos Funcionais e Não-Funcionais do Sistema de Software Health Watcher em Texto Narrativo (continuação).

Requisitos Funcionais e Não-Funcionais
[RF13] Cadastrar Novo Funcionário
Tem como propósito realizar o cadastro de um novo funcionário no sistema. O funcionário deverá estar logado no sistema, assim entrar com as seguintes informações do novo funcionário devem ser inseridas, tais como Nome, Login e Senha (com campo especial para confirmação). Após, o funcionário confirma a inserção e o sistema cadastra os dados do novo funcionário. Caso algum dos dados não tenha sido informado, mostrar uma mensagem de erro. Ao final o novo funcionário cadastrado no sistema.
[RF14] Atualizar Funcionário
Tem como propósito permitir a atualização de dados do funcionário no sistema. O funcionário deverá estar logado no sistema e ativa a opção atualizar funcionário, entra com as informações para atualização, assim como Nome, Nova senha (com campo extra para confirmação) e Senha atual. O funcionário confirma a atualização e o sistema atualiza os dados do funcionário. Caso algum dos dados não tenha sido informado o nome ou a senha atual, mostrar uma mensagem de erro.
[RF15] Atualizar Unidade de Saúde
Tem como propósito permitir a atualização de dados da unidade de saúde no sistema. O funcionário deverá estar logado no sistema e ativa a opção atualizar unidade de saúde, entra com o código da unidade de saúde a ser atualizada, o sistema mostra os dados da unidade de saúde e o funcionário modifica os dados necessários, após isso é confirmado a alteração e o sistema atualiza os dados da unidade de saúde. Caso algum dos dados não tenha sido informado, mostrar uma mensagem de erro.
[RF16] Mudar Funcionário Logado
Tem como propósito permitir que outro funcionário se “logue” no sistema. O funcionário deverá estar logado no sistema e ativa a opção mudar funcionário logado, assim o sistema exibe a tela de login. A partir daqui o fluxo principal e secundários seguem o mesmo fluxo do [RF10] Login. Caso a senha ou funcionário não seja válido, mostrar uma mensagem de erro.
[RNF01] Usabilidade
O sistema deve ter uma interface de fácil utilização, visto que o sistema pode ser utilizado por qualquer pessoa que tem acesso à Internet. O sistema deve ter um HELP on-line para ser consultado por qualquer pessoa que acesse o sistema.
[RNF02] Confiabilidade
O sistema deve estar disponível 24 horas por dia durante os 7 dias da semana. Por não ser um sistema crítico, o sistema poderá ficar fora do ar até que seja corrigida alguma falha que possa ocorrer.
[RNF03] Desempenho
O sistema deve prover acesso a 20 usuários simultaneamente. O tempo de resposta não deve ultrapassar 05 segundos por acesso.
[RNF04] Segurança
O sistema deve utilizar algum protocolo de segurança para envio de dados pela Internet. Para ter acesso aos recursos de registro das queixas, o usuário deve estar habilitado pelo controle de acesso ao sistema.
[RNF05] Padrões
O sistema deve ser desenvolvido dentro dos padrões estabelecidos pela Emprtel, responsável pelas normas de padronização de sistemas da Prefeitura da Cidade do Recife.
[RNF06] Portabilidade
Uma Licença de uso do Microsoft Windows para a estação de trabalho. Um microcomputador Pentium III, com 256 Megas de Memória, Placa de rede 3Com 10/100. Este equipamento será utilizado como estação de trabalho para a atendente.
[RNF07] Distribuição
O sistema deve ser capaz de ser executado em máquinas diferentes. Por exemplo, o núcleo do sistema pode estar sendo executado em uma máquina e os servlets em outra.

(continua)

Tabela 1.D - Requisitos Funcionais e Não-Funcionais do Sistema de Software Health Watcher em Texto Narrativo (continuação).

Requisitos Funcionais e Não-Funcionais
[RNF08] Interface com o Usuário
A interface com o usuário deve ser implementada em servlets.
[RNF09] Meio de Armazenamento
O sistema deve ser passível de extensão no que se refere ao meio de armazenamento, podendo, por exemplo, serem usados arrays ou diferentes bancos de dados (MySQL, Oracle, etc.).
[RNF10] Manipulação de Erros e Exceções
Várias funcionalidades podem disparar erros enquanto o usuário interage com o sistema e requer diferentes técnicas de manipulação. Erros gerais que se aplicam à maioria dos casos são devido à falta de informação e o sistema envia um sinal do erro e quais campos devem ser preenchidos. Outros erros podem estar relacionados com entradas inválidas de dados e o mecanismo de manipulação de erros deve tentar evitá-los ou levantar o erro e sugerir a correção.

Fonte: Do autor (2020).

APÊNDICE E - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE HEALTH WATCHER UTILIZANDO PRS

Antes de iniciar o estudo experimental piloto, a abordagem apresentada no Capítulo 5 para padronização de requisitos de software foi utilizada no DRS do sistema de software Health Watcher, conforme apresentado na Seção 7.2 deste trabalho. Para este estudo experimental piloto, foram utilizados *templates* para PRS de CRUD, de Segurança, de Desempenho, de Usabilidade e de Portabilidade. Assim, dos 19 requisitos do sistema de software Health Watcher, em 11 requisitos foram aplicados os *templates* para PRS contidos no Capítulo 4.

O resultado obtido ao realizar a atividade de **Padronização dos DRS**, utilizando os *templates* para PRS selecionados do Capítulo 4 e a abordagem proposta no Capítulo 5, é apresentado no APÊNDICE D, contendo os requisitos funcionais e os requisitos não-funcionais em texto narrativo. Após a aplicação da abordagem de transformação do DRS em texto narrativo conforme Capítulo 5. Em 11 requisitos, sendo sete requisitos funcionais e 4 requisitos não-funcionais, foram aplicados os *templates* para PRS, conforme apresentado na Tabela E a descrição de suas respectivas funcionalidades, atentando-se para não modificar o sentido dos requisitos presentes no DRS, escritos em notação de casos de uso originalmente e os *templates* para PRS utilizado na padronização do requisito.

Tabela 1.E - Requisitos Funcionais e Requisitos Não-Funcionais do Sistema de Software Health Watcher utilizando PRS.

Código	Descrição do Requisito	PRS Utilizado
RF01 - Consultar Informações	Permitir a recuperação de informação de UnidadeDeSaude pelo(s) atributo(s) especialidade, informações sobre a queixa, situação da queixa (ABERTA, SUSPENSA ou FECHADA), parecer técnico, data e hora do parecer e funcionário que realizou o parecer e informações sobre as doenças, tais como descrição da doença, quais são os sintomas, qual é a forma de manifestação, qual é o tempo de duração e os locais de referência para tratamento da doença, esta funcionalidade é executada quando o usuário seleciona os atributos na tela.	PRS de CRUD - Recuperar Informação

(continua)

Tabela 1.E - Requisitos Funcionais e Requisitos Não-Funcionais do Sistema de Software Health Watcher utilizando PRS (continuação).

Código	Descrição do Requisito	PRS Utilizado
RF02 - Especificar Queixa	Permitir a inclusão de informações de Queixas, contendo os seguintes atributos: queixa animal: tais como, apreensão de animais, controle de vetores e animais sinantrópicos (roedores, escorpiões, morcegos, etc.), doenças associadas ao pernilongo (dengue, filariose), maus tratos com animais. De QueixaAlimentar contendo os seguintes atributos: casos de suspeita por ingestão de alimentos estragados e a inclusão de queixa diversa: casos relacionados a diversos motivos, motivos estes que não têm ligações com as queixas citadas anteriormente (restaurante suspeito quanto à higiene, fossas a céu aberto, carros pipas de procedimento suspeito, etc). De TiposDeQueixa, contendo os seguintes atributos: descrição (obrigatório) e observações (opcional), dados do reclamante, nome, rua, complemento, bairro, cidade, estado, CEP, número do telefone e e-mail e Situação da queixa (obrigatório), que pode ser: ABERTA, SUSPENSA ou FECHADA. De OutraInformacoesDeQueixas, contendo os seguintes atributos: data de registro da queixa, Queixa Animal: tipo de animal (obrigatório), quantidade de animais (obrigatório) e data do incômodo (obrigatório), rua, complemento, bairro, cidade, estado, CEP e número do telefone do local de ocorrência (todas estas informações são opcionais), Queixa Alimentar: nome da vítima (obrigatório); rua, complemento, bairro, cidade, estado, CEP e número do telefone da vítima (todas opcionais); quantidade de comensais (pessoas que comeram a comida), quantidade de doentes, número de pessoas internadas e número de óbitos (todos obrigatórios); local em que os pacientes foram atendidos e refeição suspeita (todas opcionais); Queixa Diversa: idade (obrigatório), escolaridade (opcional) e ocupação (opcional); rua, complemento, bairro, cidade, estado, CEP e número do telefone, a queixa deve ser cadastrada e com a situação ABERTA.	PRS de CRUD - Incluir Informação
RF11 - Cadastrar Tabelas	Permitir a inclusão de informações de UnidadeDeSaude, contendo os seguintes atributos: código da unidade, descrição da unidade. De Especialidade, contendo os seguintes atributos: código e descrição. De UnidadeDeSaúdeEEspecialidade, contendo os seguintes atributos: unidade de saúde e especialidade. De Funcionario, contendo os seguintes atributos: login, nome e senha. De TipoDeDoenca, contendo os seguintes atributos: código, nome, descrição, manifestação e duração. De Sintoma, contendo os seguintes atributos: código e descrição. De TipoDeDoencaESintoma, contendo os seguintes atributos: tipo de doença e sintoma.	PRS de CRUD - Incluir Informação
RF12 - Atualizar Queixa	Permitir a alteração de AndamentoDaQueixa no(s) seguinte(s) atributo(s) queixa animal: apreensão de animais, controle de vetores e animais sinantrópicos (roedores, escorpiões, morcegos, etc.), doenças associadas ao pernilongo (dengue, filariose), maus tratos com animais, com situação ABERTA.	PRS de CRUD - Alterar Informação
RF13 - Cadastrar Novo Funcionário	Permitir a inclusão de informações de Funcionario, contendo os seguintes atributos: Nome, Login e Senha (com campo especial para confirmação), funcionário deve estar acessando o sistema.	PRS de CRUD - Incluir Informação
RF14 - Atualizar Funcionário	Permitir a alteração de Funcionario no(s) seguinte(s) atributo(s) Nome, Nova Senha (com campo extra para confirmação) e Senha atual, caso algum dos dados não tenha sido informado o nome ou a senha atual, mostrar uma mensagem de erro e caso sejam informados o funcionário confirma a atualização, sistema atualiza os dados do funcionário.	PRS de CRUD - Alterar Informação
RF15 - Atualizar unidade de saúde	Permitir a alteração de UnidadeDeSaude no(s) seguinte(s) atributo(s) código da unidade, descrição da unidade, entra com o código da unidade a ser alterada.	PRS de CRUD - Alterar Informação

(continua)

Tabela 1.E - Requisitos Funcionais e Requisitos Não-Funcionais do Sistema de Software Health Watcher utilizando PRS (continuação).

Código	Descrição do Requisito	PRS Utilizado
RNF01 - Usabilidade	Permitir que o Usuario realize suas tarefas, utilizando a funcionalidade receber e controlar as denúncias, notificações do sistema de software, com a condição de poder ser utilizado por qualquer pessoa que tem acesso à Internet e deve ter um HELP on-line para ser consultado por qualquer pessoa que acesse o sistema.	PRS de Usabilidade
RNF03 - Desempenho	Permitir acesso à funcionalidade receber e controlar as denúncias, notificações do sistema de software e o tempo de resposta não deve exceder 5 segundos por acesso de 20 usuários simultaneamente.	PRS de Desempenho
RNF04 - Segurança	Permitir acesso a EnvioDeDados e RegistroDasQueixas na funcionalidade de envio de dados pela Internet e recursos de registro das queixas, apenas quando estiver autenticado no sistema o usuário deve estar habilitado pelo controle de acesso ao sistema.	PRS de Segurança
RNF06 - Portabilidade	Permitir que o Health Watcher seja executado em diferentes hardwares, softwares, sistemas operacionais ou ambientes de uso, com a condição de uma Licença de uso do Microsoft Windows para a estação de trabalho e um microcomputador Pentium III, com 256 Megas de Memória, Placa de rede 3Com 10/100 mantendo a capacidade de eficiência e eficácia.	PRS de Portabilidade

Fonte: Do autor (2020).

APÊNDICE F - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE **EASYHALF**

Tabela 1.F - Requisitos Funcionais e Requisitos Não-Funcionais do Sistema de Software EasyHal f.

Código	Descrição do Requisito
RF01 - Cadastrar Estudantes	A aplicação deve permitir que estudantes façam seus cadastros. Para isso, o estudante interessado deverá informar <i>nome, e-mail, senha, CPF, data de nascimento, matrícula e instituição de ensino</i> . Além disso, ele deve concordar com os termos de uso da aplicação.
RF02 - Alterar Conta de Estudante	A aplicação deve permitir que estudantes desativem suas contas a qualquer momento, impossibilitando que os estabelecimentos cadastrados na aplicação tenham acesso às suas informações.
RF03 - Cadastrar Representantes de Estabelecimentos	A aplicação de permitir que representantes de estabelecimentos façam seu cadastro. Para isso, o representante do estabelecimento interessado deve informar o <i>nome do estabelecimento, nome, e-mail, senha, CPF, telefone, endereço do representante e CNPJ do estabelecimento</i> . Além disso, o representante do estabelecimento deve concordar com os termos de uso da aplicação e prover os documentos necessários para validação do cadastro do estabelecimento, junto às instituições de ensino, via edital específico.
RF04 - Cadastrar Instituição de Ensino	A aplicação deve permitir representantes de instituições de ensino façam seus cadastros. Para isso, o representante da instituição interessada deve informar o <i>nome da instituição, nome, e-mail, senha, CPF, telefone e endereço</i> do representante da instituição de ensino e nome da classe de consulta aos dados dos estudantes da instituição.
RF05 - Consultar Informações de Estudantes	A aplicação deve permitir que o representante de um estabelecimento, devidamente autenticado, possa consultar a situação da matrícula de um estudante. A consulta deve retornar à situação atual da matrícula consultada (ativa ou inativa), bem como informações do estudante vinculado a essa matrícula, tais como nome e instituição de ensino do estudante.
RF06 - Consultar Situação dos Representantes de Estabelecimentos	A aplicação deve permitir que o representante de um estabelecimento, devidamente autenticado, possa consultar a situação de seu cadastro junto às instituições de ensino cadastradas na aplicação.
RF07 - Alterar Informações dos Representantes de Estabelecimentos	A aplicação deve permitir que o representante de um estabelecimento, devidamente autenticado, possa gerenciar as informações de seu cadastro, tais como <i>telefone, endereço e demais dados do representante do estabelecimento</i> .
RF08 - Alterar Informações dos Estudantes	A aplicação deve permitir que um estudante, devidamente autenticado, gere as informações de seu cadastro, tais como <i>nome, instituição de ensino e matrícula</i> , entre outros.
RF09 - Alterar Autorização dos Estabelecimentos	A aplicação deve permitir que o representante de uma instituição de ensino, devidamente autenticado, autorize ou desautorize um estabelecimento cadastrado na aplicação a consultar os dados de seus estudantes. No caso de não autorização, a instituição de ensino deve ser capaz de descrever um parecer sobre essa ação.
RF10 - Alterar Informações da Instituição de Ensino	A aplicação deve permitir que o representante de uma instituição de ensino, devidamente autenticado, gere suas informações de cadastro.
RF11 - Listar Estabelecimentos	A aplicação deve permitir que um estudante, devidamente autenticado, possa visualizar a lista de estabelecimentos cadastrados na aplicação.
RNF01 - Autenticação de Usuários	A aplicação deve permitir que o usuário seja autenticado, informando seu e-mail e senha cadastrados, para que ele possa utilizar as funções disponíveis na aplicação.

Fonte: Do autor (2020).

APÊNDICE G - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE EASYHALF UTILIZANDO PRS

Tabela 1.G - Requisitos Funcionais e Requisitos Não-Funcionais do Sistema de Software EasyHalf utilizando PRS.

Código	Descrição do Requisito	PRS Utilizado
RF01 - Cadastrar Estudantes	Permitir a inclusão de informações de Estudante, contendo os seguintes atributos: nome, e-mail, senha, CPF, data de nascimento, matrícula e instituição de ensino, deve concordar com os termos de uso da aplicação.	PRS de CRUD - Incluir Informação
RF02 - Alterar Conta de Estudante	Permitir a alteração de Estudante no(s) seguinte(s) atributo(s) desativem suas contas, qualquer momento.	PRS de CRUD - Alterar Informação
RF03 - Cadastrar Representantes de Estabelecimentos	Permitir a inclusão de informações de RepresentanteEstabelecimento, contendo os seguintes atributos: nome do estabelecimento, nome, e-mail, senha, CPF, telefone, endereço do representante e CNPJ do estabelecimento, deve concordar com os termos de uso da aplicação e prover os documentos necessários para validação do cadastro do estabelecimento, junto às instituições de ensino, via edital específico.	PRS de CRUD - Incluir Informação
RF04 - Cadastrar Instituição de Ensino	Permitir a inclusão de informações de RepresentanteInstituicao, contendo os seguintes atributos: nome da instituição, nome, e-mail, senha, CPF, telefone e endereço do representante da instituição de ensino e nome da classe de consulta aos dados dos estudantes da instituição, condição não aplicada.	PRS de CRUD - Incluir Informação
RF05 - Consultar Informações de Estudantes	Permitir a recuperação de informação de Estudante pelo(s) atributo(s) situação atual da matrícula consultada (ativa ou inativa), nome e instituição de ensino do estudante, o representante de um estabelecimento, pode consultar a situação da matrícula de um estudante devida.	PRS de CRUD - Recuperar Informação
RF06 - Consultar Situação dos Representantes de Estabelecimentos	Permitir a recuperação de informação de RepresentanteEstabelecimento pelo(s) atributo(s) situação de seu cadastro, o representante de um estabelecimento, pode consultar a situação de seu cadastro junto às instituições de ensino cadastradas na aplicação.	PRS de CRUD - Recuperar Informação
RF07 - Alterar Informações dos Representantes de Estabelecimentos	Permitir a alteração de RepresentanteEstabelecimento no(s) seguinte(s) atributo(s) telefone, endereço e demais dados do representante do estabelecimento, o representante de um estabelecimento, possa gerenciar as informações de seu cadastro.	PRS de CRUD - Alterar Informação
RF08 - Alterar Informações dos Estudantes	Permitir a alteração de Estudante no(s) seguinte(s) atributo(s) nome, instituição de ensino e matrícula, entre outros, o estudante gerencie as informações de seu cadastro.	PRS de CRUD - Alterar Informação
RF09 - Alterar Autorização dos Estabelecimentos	Permitir a alteração de RepresentanteInstituicao no(s) seguinte(s) atributo(s) autorize ou desautorize a consultar os dados de seus estudantes, caso negativo, a instituição de ensino deve ser capaz de descrever um parecer sobre essa ação.	PRS de CRUD - Alterar Informação
RF10 - Alterar Informações da Instituição de Ensino	Permitir a alteração de RepresentanteInstituicao no(s) seguinte(s) atributo(s) informações de cadastro, o representante de uma instituição de ensino.	PRS de CRUD - Alterar Informação
RF11 - Listar Estabelecimentos	Permitir a recuperação de informação de Estabelecimento pelo(s) atributo(s) dados dos estabelecimentos, um estudante possa visualizar a lista de estabelecimentos cadastrados na aplicação.	PRS de CRUD - Recuperar Informação

(continua)

Tabela 1.G - Requisitos Funcionais e Requisitos Não-Funcionais do Sistema de Software EasyHalf utilizando PRS (continuação).

Código	Descrição do Requisito	PRS Utilizado
RNF01 - Autenticação de Usuários	Permitir acesso a Usuario na funcionalidade de funções disponíveis na aplicação, apenas quando estiver autenticado no sistema informando seu e-mail e senha cadastrados.	PRS de Segurança

Fonte: Do autor (2020).

APÊNDICE H - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE FAKE NEWS APP

Tabela 1.H - Requisitos Funcionais e Não-Funcionais do Sistema de Software Fake News App.

Requisitos Funcionais e Não-Funcionais
[RF01] - Criar conta
<p>Breve descrição: Permite ao usuário criar uma conta para autenticar-se no aplicativo e utilizar suas funções.</p> <p>Atores: Usuário e FakeNews API</p> <p>Casos de uso incluídos: Nenhum</p> <p>Casos de uso estendidos: Nenhum</p> <p>Pré-condições: Nenhuma</p> <p>Fluxo básico:</p> <ol style="list-style-type: none"> 1. O usuário inicia o caso de uso. 2. O aplicativo solicita e-mail, senha e confirmação de senha ao usuário. 3. O usuário informa os dados necessários e confirma o cadastramento. 4. O aplicativo valida os dados do usuário e os envia ao FakeNews API [E1, E2]. 5. O FakeNews API armazena os dados do usuário e confirma o cadastro da conta [E3]. 6. O aplicativo envia a mensagem “Conta criada com sucesso!” ao usuário e encerra o caso de uso. <p>Exceção E1: uma ou mais informações não foram informadas pelo usuário.</p> <ol style="list-style-type: none"> 1. O aplicativo envia a mensagem “Por favor, preencha todos os campos do formulário!” e volta para o passo 2 do fluxo básico. <p>Exceção E2: as senhas informadas pelo usuário não conferem.</p> <ol style="list-style-type: none"> 1. O aplicativo envia a mensagem “As senhas informadas não conferem!” e volta para o passo 2 do fluxo básico. <p>Exceção E3: já existe uma conta cadastrada com o endereço de e-mail informado.</p> <ol style="list-style-type: none"> 1. O aplicativo envia a mensagem “Já existe uma conta cadastrada com esse endereço de e-mail!” e volta para o passo 2 do fluxo básico.
[RF02] - Autenticar usuário
<p>Breve descrição: Permite ao usuário se autenticar no aplicativo para utilização de suas funções.</p> <p>Atores: Usuário e FakeNews API</p> <p>Casos de uso incluídos: Nenhum</p> <p>Casos de uso estendidos: Nenhum</p> <p>Pré-condições: Nenhuma</p> <p>Fluxo básico:</p> <ol style="list-style-type: none"> 1. O usuário inicia o caso de uso. 2. O aplicativo solicita e-mail e senha ao usuário. 3. O usuário informa os dados necessários e solicita entrar no aplicativo. 4. O aplicativo valida os dados do usuário e os envia ao FakeNews API [E1]. 5. O FakeNews API autentica o usuário e envia seu id e seu token de acesso [E2]. 6. O aplicativo armazena o id e token de acesso do usuário e encerra o caso de uso. <p>Exceção E1: uma ou mais informações não foram informadas pelo usuário.</p> <ol style="list-style-type: none"> 1. O aplicativo envia a mensagem “Por favor, preencha todos os campos do formulário!” e volta para o passo 2 do fluxo básico. <p>Exceção E2: a autenticação falhou.</p> <ol style="list-style-type: none"> 1. O aplicativo envia a mensagem “Login/senha inválidos!” e volta para o passo 2 do fluxo básico.

(continua)

Tabela 1.H - Requisitos Funcionais e Não-Funcionais do Sistema de Software Fake News App (continuação).

Requisitos Funcionais e Não-Funcionais
[RF03] - Sair
<p>Breve descrição: Permite ao usuário desconectar-se do sistema.</p> <p>Atores: Usuário</p> <p>Casos de uso incluídos: Nenhum</p> <p>Casos de uso estendidos: Nenhum</p> <p>Pré-condições:</p> <ol style="list-style-type: none"> 1. O usuário deve estar autenticado no aplicativo. <p>Fluxo básico:</p> <ol style="list-style-type: none"> 1. O usuário inicia o caso de uso. 2. O aplicativo remove o id e token de acesso do usuário e encerra o caso de uso.
[RF04] - Listar/buscar notícias
<p>Breve descrição: Apresenta a lista de notícias cadastradas até o momento.</p> <p>Atores: Usuário e FakeNews API</p> <p>Casos de uso incluídos: Nenhum</p> <p>Casos de uso estendidos: Nenhum</p> <p>Pré-condições:</p> <ol style="list-style-type: none"> 1. O usuário deve estar autenticado no aplicativo. <p>Fluxo básico:</p> <ol style="list-style-type: none"> 1. O usuário inicia o caso de uso. 2. O aplicativo solicita ao FakeNews API a lista de notícias cadastradas, passando o token de acesso do usuário e nenhuma string de busca [A1]. 3. O FakeNews API retorna a lista com todas as notícias cadastradas, em ordem decrescentemente de data da publicação. 4. O aplicativo exibe a lista de notícias retornada e encerra o caso de uso. Para cada notícia, tem-se: título, período de tempo desde a publicação da notícia, imagem de capa e quantidade de likes [E1]. <p>Fluxo alternativo A1: o usuário informa uma string de busca.</p> <ol style="list-style-type: none"> 1. O aplicativo solicita ao FakeNews API a lista de notícias cadastradas, passando o token de acesso e a string de busca informada pelo usuário. 2. O FakeNews API retorna a lista de notícias cadastradas, cujo título seja compatível com a string de busca informada. 3. Vai para o passo 4 do fluxo básico. <p>Exceção E1: não há notícias cadastradas no momento.</p> <ol style="list-style-type: none"> 1. O aplicativo envia a mensagem “Lista de notícias vazia!” e encerra o caso de uso.
[RF05] - Listar notícias favoritas
<p>Breve descrição: Apresenta a lista de notícias favoritas pelo usuário até o momento.</p> <p>Atores: Usuário e FakeNews API</p> <p>Casos de uso incluídos: Nenhum</p> <p>Casos de uso estendidos: Nenhum</p> <p>Pré-condições:</p> <ol style="list-style-type: none"> 1. O usuário deve estar autenticado no aplicativo. <p>Fluxo básico:</p> <ol style="list-style-type: none"> 1. O usuário inicia o caso de uso. 2. O aplicativo solicita ao FakeNews API a lista de notícias favoritas do usuário, passando o id e token de acesso do usuário. 3. O FakeNews API retorna a lista com todas as notícias favoritas pelo usuário, em ordem decrescentemente de data da publicação. 4. O aplicativo exibe a lista de notícias retornada e encerra o caso de uso. Para cada notícia, tem-se: título, período de tempo desde a publicação da notícia, imagem de capa e quantidade de likes [E1]. 5. O aplicativo encerra o caso de uso. <p>Exceção E1: não há notícias favoritas no momento.</p> <ol style="list-style-type: none"> 1. O aplicativo envia a mensagem “Lista de notícias vazia!” e encerra o caso de uso.

(continua)

Tabela 1.H - Requisitos Funcionais e Não-Funcionais do Sistema de Software Fake News App (continuação).

Requisitos Funcionais e Não-Funcionais
[RF06] - Gerenciar notícias favoritas
<p>Breve descrição: Adiciona uma notícia à lista de notícias favoritas do usuário.</p> <p>Atores: Usuário e FakeNews API</p> <p>Casos de uso incluídos: Nenhum</p> <p>Casos de uso estendidos: Nenhum</p> <p>Pré-condições:</p> <ol style="list-style-type: none"> 1. O usuário deve estar autenticado no aplicativo. <p>Fluxo básico:</p> <ol style="list-style-type: none"> 1. O usuário inicia o caso de uso, a partir da notícia apresentada no caso de uso “RF08 - Exibir notícia”. 2. O aplicativo solicita ao FakeNews API a inclusão da notícia selecionada na lista de favoritas do usuário, passando o id e o token de acesso do usuário, bem como o id da notícia selecionada [A1]. 3. O aplicativo envia a mensagem “Notícia favoritada com sucesso!” e encerra o caso de uso. <p>Fluxo alternativo A1: a notícia já está na lista de favoritas do usuário.</p> <ol style="list-style-type: none"> 1. O aplicativo solicita ao FakeNews API que remova a notícia selecionada da lista de favoritas, passando o id da notícia, o id e o token de acesso do usuário. 2. O FakeNews API remove a notícia da lista de favoritas do usuário. <p>O aplicativo envia a mensagem “Notícia removida com sucesso!” e encerra o caso de uso.</p>
[RF07] - Gerenciar like da notícia
<p>Breve descrição: Atribui ou remove o “like” de uma notícia.</p> <p>Atores: Usuário e FakeNews API</p> <p>Casos de uso incluídos: Nenhum</p> <p>Casos de uso estendidos: Nenhum</p> <p>Pré-condições:</p> <ol style="list-style-type: none"> 1. O usuário deve estar autenticado no aplicativo. <p>Fluxo básico:</p> <ol style="list-style-type: none"> 1. O usuário inicia o caso de uso, a partir da notícia apresentada no caso de uso “RF08 - Exibir notícia”. 2. O aplicativo solicita ao FakeNews API que atribua um “like” à notícia selecionada, passando o id da notícia, o id e o token de acesso do usuário [A1]. 3. O FakeNews API atribui o “like” do usuário à notícia selecionada. 4. O aplicativo envia a mensagem “Like atribuído com sucesso!” e encerra o caso de uso. <p>Fluxo alternativo A1: a notícia já possui um “like” atribuído pelo usuário.</p> <ol style="list-style-type: none"> 1. O aplicativo solicita ao FakeNews API que remova o “like” da notícia selecionada, passando o id da notícia, o id e o token de acesso do usuário. 2. O FakeNews API remove o “like” do usuário para a notícia selecionada. 3. O aplicativo envia a mensagem “Like removido com sucesso!” e encerra o caso de uso.
[RF08] - Exibir notícia
<p>Breve descrição: Apresenta o texto que descreve os motivos pelos quais a notícia é considerada uma fake news.</p> <p>Atores: Usuário e FakeNews API</p> <p>Casos de uso incluídos: Nenhum</p> <p>Casos de uso estendidos: Nenhum</p> <p>Pré-condições:</p> <ol style="list-style-type: none"> 1. O usuário deve estar autenticado no aplicativo. <p>Fluxo básico:</p> <ol style="list-style-type: none"> 1. O usuário inicia o caso de uso, selecionando uma notícia na lista de notícias apresentadas no caso de uso “RF04 - Listar/buscar notícias” ou “RF05 - Listar notícias favoritas”. 2. O aplicativo solicita ao FakeNews API a notícia, passando o id da notícia selecionada e o token de acesso do usuário. 3. O FakeNews API retorna a notícia. 4. O aplicativo exibe os dados da notícia retornada (título, período de tempo desde a publicação da notícia, imagem de capa, quantidade de likes e link de acesso ao texto completo da notícia) e encerra o caso de uso.

(continua)

Tabela 1.H - Requisitos Funcionais e Não-Funcionais do Sistema de Software Fake News App (continuação).

Requisitos Funcionais e Não-Funcionais
[RF09] - Compartilhar notícia
<p>Breve descrição: Compartilha o título e o link de acesso ao texto completo da notícia com outros indivíduos, via WhatsApp.</p> <p>Atores: Usuário e WhatsApp</p> <p>Casos de uso incluídos: Nenhum</p> <p>Casos de uso estendidos: Nenhum</p> <p>Pré-condições:</p> <ol style="list-style-type: none"> 1. O usuário deve estar autenticado no aplicativo. <p>Fluxo básico:</p> <ol style="list-style-type: none"> 1. O usuário inicia o caso de uso, a partir da notícia apresentada no caso de uso “RF08 - Exibir notícia”. 2. O aplicativo solicita ao WhatsApp o compartilhamento da notícia, passando a imagem de capa, o título e o link de acesso ao texto completo da notícia [E1]. 3. O aplicativo envia a mensagem “Notícia compartilhada com sucesso!” e encerra o caso de uso. <p>Exceção E1: a notícia já está na lista de favoritas do usuário.</p> <ol style="list-style-type: none"> 1. O aplicativo envia a mensagem “Não foi possível compartilhar a notícia!” e encerra o caso de uso.
[RF10] - Ler mais sobre a notícia
<p>Breve descrição: Acessa o link com o texto completo da notícia.</p> <p>Atores: Usuário e Browser</p> <p>Casos de uso incluídos: Nenhum</p> <p>Casos de uso estendidos: Nenhum</p> <p>Pré-condições:</p> <ol style="list-style-type: none"> 1. O usuário deve estar autenticado no aplicativo. <p>Fluxo básico:</p> <ol style="list-style-type: none"> 1. O usuário inicia o caso de uso, a partir da notícia apresentada no caso de uso “RF08 - Exibir notícia”. 2. O aplicativo solicita ao Browser que abra o website com o texto completo da notícia, passando o link de acesso e encerra o caso de uso.
[RNF01] - Facilidade de uso
<p>O aplicativo deve permitir que um usuário, tendo passado por um treinamento de 15 (quinze) minutos quanto ao uso de suas funções, consiga executar mais de 75% dessas funções com sucesso.</p>
[RNF02] Portabilidade
<p>O aplicativo deve estar disponível para dispositivos móveis Android (tablets e smartphones), a partir da versão 5.0 (Lollipop), contemplando aproximadamente 90% dos usuários Android.</p>
[RNF03] Segurança
<p>Para cada função do aplicativo que exija autenticação do usuário, deve-se verificar se o token de acesso e o id do usuário existem. Caso contrário, o usuário deve ser redirecionado para o caso de uso “RF02 - Autenticar usuário”.</p>

Fonte: Do autor (2020).

APÊNDICE I - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE FAKE NEWS APP EM TEXTO NARRATIVO

Tabela 1.I - Requisitos Funcionais e Não-Funcionais do Sistema de Software Fake News App em Texto Narrativo.

Requisitos Funcionais e Não-Funcionais
[RF01] - Criar conta
Permite ao usuário criar uma conta para autenticar-se no aplicativo e utilizar suas funções. O aplicativo solicita e-mail, senha e confirmação de senha ao usuário. O aplicativo valida os dados do usuário e envia a mensagem “Conta criada com sucesso!”. Caso algum problema seja encontrado durante a validação, as seguintes mensagens de erro podem ocorrer: (1) “Por favor, preencha todos os campos do formulário!”; e (2) “As senhas informadas não conferem”.
[RF02] - Autenticar usuário
Permite ao usuário se autenticar no aplicativo para utilização de suas funções. O aplicativo valida os dados do usuário e armazena o id e token do usuário e permite o acesso. Caso algum problema seja encontrado durante a validação, as seguintes mensagens de erro podem ocorrer: (1) “Por favor, preencha todos os campos do formulário!”; e (2) “Login/senha inválidos!”.
[RF03] - Sair
Permite ao usuário desconectar-se do sistema. O usuário deve estar autenticado no aplicativo. O aplicativo remove o id e token de acesso do usuário.
[RF04] - Listar/ buscar notícias
Permite ao usuário analisar a lista de notícias cadastradas até o momento. O usuário deve estar autenticado no aplicativo. A lista de notícias cadastradas é apresentada ao usuário após o envio da string de busca informada. A lista de notícias é retornada com todas as notícias cadastradas, em ordem decrescentemente de data da publicação. Para cada notícia, tem-se: título, período de tempo desde a publicação da notícia, imagem de capa e quantidade de likes. Caso não haja notícias, a seguinte mensagem pode ocorrer: (1) “Lista de notícias vazia!”.
[RF05] - Listar notícias favoritas
Permite ao usuário analisar a lista de notícias favoritas por ele até o momento. O usuário deve estar autenticado no aplicativo. A lista de notícias favoritas do usuário é apresentada em ordem decrescentemente de data da publicação. Para cada notícia, tem-se: título, período de tempo desde a publicação da notícia, imagem de capa e quantidade de likes. Caso não haja notícias, a seguinte mensagem pode ocorrer: (1) “Lista de notícias vazia!”.
[RF06] - Gerenciar notícias favoritas
Permite ao usuário adicionar uma notícia à lista de notícias favoritas do usuário. O usuário deve estar autenticado no aplicativo. A partir da notícia apresentada o usuário solicita a inclusão da notícia selecionada na lista de favoritas. O aplicativo valida os dados do usuário e da notícia, tais como, id, token de acesso do usuário, id da notícia selecionada e envia a mensagem “Notícia favoritada com sucesso!”. Caso a notícia já esteja na lista de favoritas do usuário, a seguinte mensagem pode ocorrer: (1) “Notícia removida com sucesso!”.
[RF07] - Gerenciar like da notícia
Permite ao usuário atribuir ou remover o “like” de uma notícia. O usuário deve estar autenticado no aplicativo. A partir da notícia apresentada o usuário atribui um “like” à notícia selecionada. O aplicativo solicita a atribuição de um “like” à notícia selecionada, passando o id da notícia, o id e o token de acesso do usuário e envia a mensagem “Like atribuído com sucesso!”. Caso a notícia já possui um “like” atribuído pelo usuário, a seguinte mensagem pode ocorrer: (1) “Like removido com sucesso!”.
[RF08] - Exibir notícia
Permite ao usuário analisar o texto que descreve os motivos pelos quais a notícia é considerada uma fake news. O usuário deve estar autenticado no aplicativo. A partir da notícia selecionada na lista de notícias, exhibe os dados da notícia (título, período de tempo desde a publicação da notícia, imagem de capa, quantidade de likes e link de acesso ao texto completo da notícia).

(continua)

Tabela 1.I - Requisitos Funcionais e Não-Funcionais do Sistema de Software Fake News App em Texto Narrativo (continuação).

Requisitos Funcionais e Não-Funcionais	
[RF09] - Compartilhar notícia	Permite ao usuário compartilhar o título e o link de acesso ao texto completo da notícia com outros indivíduos, via WhatsApp. O usuário deve estar autenticado no aplicativo. A partir da notícia selecionada na lista de notícias, solicita ao WhatsApp o compartilhamento da notícia, passando a imagem de capa, o título e o link de acesso ao texto completo da notícia e envia a mensagem “Notícia compartilhada com sucesso!”. Caso algum problema seja encontrado durante envio, a seguinte mensagem de erro pode ocorrer: (1) “Não foi possível compartilhar a notícia!”.
[RF10] - Ler mais sobre a notícia	Permite ao usuário acessar o link com o texto completo da notícia. O usuário deve estar autenticado no aplicativo. A partir da notícia selecionada na lista de notícias, é solicitado ao Browser que abra o website com o texto completo da notícia, passando o link de acesso.
[RNF01] - Facilidade de uso	O aplicativo deve permitir que um usuário, tendo passado por um treinamento de 15 (quinze) minutos quanto ao uso de suas funções, consiga executar mais de 75% dessas funções com sucesso.
[RNF02] - Portabilidade	O aplicativo deve estar disponível para dispositivos móveis Android (tablets e smartphones), a partir da versão 5.0 (Lollipop), contemplando aproximadamente 90% dos usuários Android.
[RNF03] - Segurança	Para cada função do aplicativo que exija autenticação do usuário, deve-se verificar se o token de acesso e o id do usuário existem. Caso contrário, o usuário deve ser redirecionado para o uma nova autenticação.

Fonte: Do autor (2020).

APÊNDICE J - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE FAKE NEWS APP UTILIZANDO PRS

Tabela 1.J - Requisitos Funcionais e Não-Funcionais do Sistema de Software Fake News App utilizando PRS.

Código	Descrição do Requisito	PRS Utilizado
RF01 - Criar conta	Permitir a inclusão de informações de ContaUsuario, contendo os seguintes atributos: e-mail, senha e confirmação de senha ao usuário, caso algum problema seja encontrado durante a validação, as seguintes mensagens de erro podem ocorrer: (1) "Por favor, preencha todos os campos do formulário!"; e (2) "As senhas informadas não conferem".	PRS de CRUD - Incluir Informação
RF02 - Autenticar usuário	Permitir acesso a Usuario na funcionalidade de aplicativo para utilização de suas funções, apenas quando estiver autenticado no sistema valida os dados do usuário e armazena o id e token do usuário, caso algum problema seja encontrado durante a validação, as seguintes mensagens de erro podem ocorrer: (1) "Por favor, preencha todos os campos do formulário!"; e (2) "Login/senha inválidos!".	PRS de Segurança
RF03 - Login	Permitir acesso a Usuario na funcionalidade de desconectar-se do sistema, apenas quando estiver autenticado no sistema o aplicativo remove o id e token de acesso do usuário.	PRS de Segurança
RF04 - Listar/ buscar notícias	Permitir a recuperação de informação de Noticias pelelo(s) atributo(s) título, período de tempo desde a publicação da notícia, imagem de capa e quantidade de likes, as notícias cadastradas são apresentadas ao usuário após o envio da string de busca informada, caso não haja notícias, a seguinte mensagem pode ocorrer: (1) "Lista de notícias vazia!".	PRS de CRUD - Recuperar Informação
RF05 - Listar notícias favoritas	Permitir a recuperação de informação de NoticiasFavoritadas pelelo(s) atributo(s) título, período de tempo desde a publicação da notícia, imagem de capa e quantidade de likes, as notícias favoritas do usuário são apresentadas em ordem decrescentemente de data da publicação, caso não haja notícias, a seguinte mensagem pode ocorrer: (1) "Lista de notícias vazia!".	PRS de CRUD - Recuperar Informação
RF06 - Gerenciar notícias favoritas	Permitir a alteração de Noticia no(s) seguinte(s) atributo(s) id, token de acesso do usuário, id da notícia selecionada, usuário solicita a inclusão da notícia selecionada na lista de favoritas, caso a notícia já esteja na lista de favoritas do usuário, a seguinte mensagem pode ocorrer: (1) "Notícia removida com sucesso!".	PRS de CRUD - Alterar Informação
RF07 - Gerenciar like da notícia	Permitir a alteração de LikeNoticia no(s) seguinte(s) atributo(s) id da notícia, o id e o token de acesso do usuário, partir da notícia apresentada o usuário atribui um "like" à notícia selecionada, caso a notícia já possui um "like" atribuído pelo usuário, a seguinte mensagem pode ocorrer: (1) "Like removido com sucesso!".	PRS de CRUD - Alterar Informação
RF08 - Exibir notícia	Permitir a recuperação de informação de NoticiaFake pelelo(s) atributo(s) título, período de tempo desde a publicação da notícia, imagem de capa, quantidade de likes e link de acesso ao texto completo da notícia, texto que descreve os motivos pelos quais a notícia é considerada uma fake news.	PRS de CRUD - Recuperar Informação
RNF01 - Facilidade de uso	Permitir que o Usuário realize suas tarefas, utilizando a funcionalidade criar conta, sair, buscar notícias, notícias favoritas, favoritar notícias, like na notícia, exibir notícia, compartilhar notícia e ler mais sobre a notícia do sistema de software, com a condição de tendo passado por um treinamento de 15 (quinze) minutos quanto ao uso de suas funções.	PRS de Usabilidade
RNF02 - Portabilidade	Permitir que o Fake News App seja executado em diferentes hardwares, softwares, sistemas operacionais ou ambientes de uso, com a condição estar disponível para dispositivos móveis Android (tablets e smartphones), a partir da versão 5.0 (Lollipop) mantendo a capacidade de eficiência e eficácia.	PRS de Portabilidade

(continua)

Tabela 1.J - Requisitos Funcionais e Não-Funcionais do Sistema de Software Fake News App utilizando PRS (continuação).

Código	Descrição do Requisito	PRS Utilizado
RNF03 - Segurança	Permitir acesso a AutenticacaoUsuario na funcionalidade cada função do aplicativo que exija autenticação do usuário, apenas quando estiver autenticado no sistema deve-se verificar se o token de acesso e o id do usuário existem.	PRS de Segurança

Fonte: Do autor (2020).

APÊNDICE K - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE OBASCID-TOOL

Tabela 1.K - Requisitos Funcionais e Não-Funcionais do Sistema de Software OBASCID-TOOL.

Código	Descrição do Requisito	Requisitos Relacionados
RF01	O software deve permitir o cadastramento, alteração e exclusão de catálogos de interesses de software. Cada catálogo deve conter nome, descrição (opcional) e tipo de licença, que pode ser pública ou privada.	RNF-1, RNF-2, RNF-3
RF02	O software deve permitir o cadastramento de pesquisadores. Cada pesquisador deve conter, obrigatoriamente nome, e-mail e senha. Opcionalmente, pode-se cadastrar: cidade, estado, país e nome da instituição a qual o pesquisador está vinculado.	RNF-1, RNF-3
RF03	O software deve a alteração do perfil de pesquisadores. Todos os dados de um pesquisador, com exceção do seu e-mail, podem ser atualizados.	RNF-1, RNF-2, RNF-3
RF04	Para cada pesquisador, o software deve cadastrar automaticamente e permitir a atualização da seguinte lista de <i>stopwords</i> : of, the, what, and, from, in, at, on, a, an, for, to, de, a, o, que, e, do, da, em, um, para.	RNF-1, RNF-3
RF05	Para cada pesquisador, o software deve cadastrar automaticamente e permitir a atualização de uma lista de configurações padrão (<i>default</i>) para entrada de dados, conforme descrito a seguir: <ol style="list-style-type: none"> 1. Opção padrão para o tipo de licença de um catálogo de interesses de software: privada; 2. Opção padrão para o tipo de licença de um documento de requisitos: privada; 3. Opção padrão para o tipo de interesses de software: não funcional; 4. Opção padrão para o tipo de requisito de software: funcional; 5. Opção padrão para o tipo de fonte de um interesse: catálogo; 6. Opção padrão para a prioridade de um interesse: <i>Not specified</i>; e 7. Opção padrão para o tipo de relacionamento entre interesses: dependência. 	RNF-1, RNF-3
RF06	O software deve permitir o cadastramento, alteração e exclusão de documentos de requisitos de software. Cada documento de requisito deve conter nome, descrição (opcional) e tipo de licença, que pode ser pública ou privada.	RNF-1, RNF-2, RNF-3
RF07	O software deve permitir o cadastramento, alteração e exclusão de interesses de software. Cada interesse deve estar vinculado a um catálogo e conter nome, descrição (opcional), tipo (funcional ou não funcional) e prioridade (baixa, média, alta ou "não especificada").	RNF-1, RNF-2, RNF-3
RF08	O software deve permitir o cadastramento, alteração e exclusão de fontes de um interesse de software. Cada fonte deve estar vinculada a um determinado interesse e conter nome, descrição (opcional) e tipo (<i>stakeholder</i> , documento de negócios ou catálogo).	RNF-1, RNF-2, RNF-3
RF09	O software deve permitir o cadastramento, alteração e exclusão de palavras-chave de um interesse de software. Cada palavra-chave deve estar vinculada a um determinado interesse, conter sua descrição e não estar na lista de <i>stopwords</i> cadastradas para o pesquisador.	RNF-1, RNF-2, RNF-3
RF10	O software deve permitir a importação/exportação das palavras-chave de um interesse por meio de arquivos de texto.	RNF-1, RNF-2, RNF-3
RF11	O software deve permitir o cadastramento, alteração e exclusão de relacionamentos entre interesses de software. Cada relacionamento deve estar vinculado a um catálogo de interesses e conter tipo (dependência, contribuição negativa ou contribuição positiva), um interesse fonte e um interesse alvo.	RNF-1, RNF-2, RNF-3
RF12	O software deve permitir o cadastramento, alteração e exclusão de requisitos de software. Cada requisito deve estar vinculado a um documento de requisitos e conter, obrigatoriamente, nome, descrição e tipo (funcional ou não funcional).	RNF-1, RNF-2, RNF-3
RF13	O software deve permitir o cadastramento, alteração e exclusão de dependências entre requisitos de software. Cada dependência deve estar vinculada a um documento de requisitos e conter tipo um requisito fonte (que depende de outro requisito) e um requisito alvo.	RNF-1, RNF-2, RNF-3

(continua)

**Tabela 1.K - Requisitos Funcionais e Não-Funcionais do Sistema de Software
OBASCID-TOOL (continuação).**

Código	Descrição do Requisito	Requisitos Relacionados
RF14	O software deve permitir o cadastramento, alteração, exclusão de unidades de identificação. Cada unidade de identificação deve estar vinculada a um documento de requisitos, conter um nome único e o catálogo a ser utilizado na identificação e classificação de interesses de software.	RNF-1, RNF-2, RNF-3
RF15	O software deve permitir a geração de um relatório geral de um catálogo de interesses de software, o qual deve conter: o nome, a descrição e os interesses do catálogo, juntamente com as palavras-chave, fontes e prioridade dos mesmos.	RNF-1, RNF-2, RNF-3
RF16	O software deve permitir a geração de um relatório de relacionamentos entre interesses de um catálogo, o qual deve conter: uma matriz de dependência e uma matriz de contribuição entre esses interesses.	RNF-1, RNF-2, RNF-3
RF17	O software deve permitir a geração de relatório geral de um documento de requisitos, o qual deve conter: o nome, a descrição e os requisitos do documento, juntamente com seu tipo e suas dependências com relação a outros requisitos.	RNF-1, RNF-2, RNF-3
RF18	O software deve permitir a identificação de interesses de software, a partir de uma unidade de identificação. Como resultado, deve-se gerar uma lista de requisitos e interesses identificados para os mesmos, destacando-se os interesses principais de cada requisito.	RNF-1, RNF-2, RNF-3
RF19	O software deve permitir a filtragem dos requisitos da lista de requisitos e interesses identificados pelo nome do "interesse principal" e do "interesse transversal".	RNF-1, RNF-2, RNF-3
RF20	O software deve gerar, caso necessário, uma lista de ocorrências a respeito da identificação de interesses do software, conforme definido na abordagem <i>ObasCId</i> .	RNF-1, RNF-2
RF21	O software deve permitir que seja informado qual é o interesse principal de um determinado requisito. Para isso, deve-se ter executado uma unidade de identificação anteriormente.	RNF-1, RNF-2, RNF-3
RF22	O software deve permitir a classificação dos interesses identificados em um documento de requisitos. Para isso, deve se ter executado uma unidade de identificação anteriormente, bem como escolhido o interesse principal de cada requisito. Como resultado, deve-se gerar uma matriz de entrelaçamentos existentes entre os interesses identificados, conforme definido na abordagem <i>ObasCId</i> .	RNF-1, RNF-2
RF23	O software deve permitir consultas ao repositório de catálogos de interesses de software. Para isso, deve-se informar um trecho do nome ou da descrição do catálogo a ser pesquisado. Apenas catálogos públicos podem ser apresentados como resultados de uma busca.	RNF-1, RNF-3
RF24	O software deve permitir consultas ao repositório de documentos de requisitos. Para isso, deve-se informar um trecho do nome ou da descrição do documento de requisitos a ser pesquisado. Apenas documentos de requisitos públicos podem ser apresentados como resultados de uma busca.	RNF-1, RNF-3
RF25	O software deve permitir a importação de catálogos de interesses de software desenvolvidos por outros. Para isso, deve-se informar um novo nome para o catálogo, sua descrição (opcional) e tipo de licença. Apenas catálogos públicos podem ser importados.	RNF-1, RNF-2, RNF-3
RF26	O software deve permitir a importação de documentos de requisitos desenvolvidos por outros. Para isso, deve-se informar um novo nome para o documento de requisitos, sua descrição (opcional) e tipo de licença. Apenas documentos de requisitos públicos podem ser importados.	RNF-1, RNF-2, RNF-3
RF27	O software deve permitir a união (<i>merge</i>) de dois catálogos de interesses de software. Todos os interesses e relacionamentos dos dois catálogos serão intercalados. Para isso, deve-se informar um novo nome para o catálogo de interesse de software, sua descrição (opcional) e tipo de licença. Caso haja nomes de interesses duplicados nos dois catálogos, um novo nome deve ser gerado automaticamente.	RNF-1, RNF-2, RNF-3

(continua)

**Tabela 1.K - Requisitos Funcionais e Não-Funcionais do Sistema de Software
OBASCID-TOOL (continuação).**

Código	Descrição do Requisito	Requisitos Relacionados
RNF01	A interface de todas as funções do software deve ser responsiva, de forma que os elementos da mesma se adaptem a dispositivos com telas menores (tais como, smartphones e tablets) com no mínimo 5 polegadas.	-
RNF02	Para utilizar as funções do software, o usuário deve estar autenticado por meio de seu e-mail e senha.	-
RNF03	O software deve ser de fácil utilização, permitindo que seus usuários, tendo passado por um treinamento de 40 (quarenta) minutos, consigam executar corretamente a maioria de suas funções.	-

Fonte: Do autor (2020).

APÊNDICE L - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE OBASCID-TOOL UTILIZANDO PRS

Tabela 1.L - Requisitos Funcionais e Não-Funcionais do Sistema de Software OBASCID-TOOL utilizando PRS.

Código	Descrição do Requisito	PRS Utilizado
RF01 - RF-01	Permitir a inclusão de informações de CatálogoInteresseSoftware, contendo os seguintes atributos: nome, descrição (opcional) e tipo de licença, licença pode ser pública ou privada.	PRS de CRUD - Incluir Informação
RF02 - RF-02	Permitir a inclusão de informações de Pesquisador, contendo os seguintes atributos: nome, e-mail e senha, opcionalmente, podem-se cadastrar: cidade, estado, país e nome da instituição a qual o pesquisador está vinculado.	PRS de CRUD - Incluir Informação
RF03 - RF-03	Permitir a alteração de PerfilPesquisador no(s) seguinte(s) atributo(s) todos os dados de um pesquisador, com exceção do seu email.	PRS de CRUD - Alterar Informação
RF04 - RF-04	Permitir a inclusão de informações de PesquisadorListaStopword, contendo os seguintes atributos: of, the, what, and, from, in, at, on, a, an, for, to, de, a, o, que, e, do, da, em, um, para, cadastrar automaticamente.	PRS de CRUD - Incluir Informação
RF05 - RF-05	Permitir a inclusão de informações de PesquisadorListaConfiguracaoPadrao, contendo os seguintes atributos: tipo de licença de um catálogo de interesses de software, tipo de licença de um documento de requisitos, tipo de interesses de software, tipo de requisito de software, tipo de fonte de um interesse, prioridade de um interesse, tipo de relacionamento entre interesses, deve cadastrar automaticamente.	PRS de CRUD - Incluir Informação
RF06 - RF-06	Permitir a inclusão de informações de DocumentoRequisitoSoftware, contendo os seguintes atributos: nome, descrição (opcional) e tipo de licença, licença pode ser pública ou privada.	PRS de CRUD - Incluir Informação
RF07 - RF-07	Permitir a inclusão de informações de InteresseSoftware, contendo os seguintes atributos: nome, descrição (opcional), tipo (funcional ou não funcional) e prioridade (baixa, média, alta ou "não especificada"), cada interesse deve estar vinculado a um catálogo.	PRS de CRUD - Incluir Informação
RF08 - RF-08	Permitir a inclusão de informações de FonteInteresseSoftware, contendo os seguintes atributos: nome, descrição (opcional) e tipo (stakeholder, documento de negócios ou catálogo), cada fonte deve estar vinculada a um determinado interesse.	PRS de CRUD - Incluir Informação
RF09 - RF-09	Permitir a inclusão de informações de PalavrasChave, contendo os seguintes atributos: descrição, cada palavra-chave deve estar vinculada a um determinado interesse e não estar na lista de stopwords cadastradas para o pesquisador.	PRS de CRUD - Incluir Informação
RF10 - RF-10	Permitir a inclusão de informações de PalavrasChave, contendo os seguintes atributos: tipo (dependência, contribuição negativa ou contribuição positiva), um interesse fonte e um interesse alvo, a importação/exportação das palavras-chave de um interesse por meio de arquivos de texto.	PRS de CRUD - Incluir Informação
RF11 - RF-11	Permitir a inclusão de informações de RelacionamentosEntreInteressesSoftware, contendo os seguintes atributos: tipo (dependência, contribuição negativa ou contribuição positiva), um interesse fonte e um interesse alvo, cada relacionamento deve estar vinculado a um catálogo de interesses.	PRS de CRUD - Incluir Informação
RF12 - RF-12	Permitir a inclusão de informações de RequisitosSoftware, contendo os seguintes atributos: obrigatoriamente, nome, descrição e tipo (funcional ou não funcional), cada requisito deve estar vinculado a um documento de requisitos.	PRS de CRUD - Incluir Informação

(continua)

Tabela 1.L - Requisitos Funcionais e Não-Funcionais do Sistema de Software OBASCID-TOOL utilizando PRS (continuação).

Código	Descrição do Requisito	PRS Utilizado
RF13 - RF-13	Permitir a inclusão de informações de DependenciaEntreRequisitosSoftware, contendo os seguintes atributos: tipo um requisito fonte (que depende de outro requisito) e um requisito alvo, cada dependência deve estar vinculada a um documento de requisitos.	PRS de CRUD - Incluir Informação
RF14 - RF-14	Permitir a inclusão de informações de UnidadeIdentificacao, contendo os seguintes atributos: nome único e o catálogo de interesses de software, cada unidade de identificação deve estar vinculada a um documento de requisitos.	PRS de CRUD - Incluir Informação
RF15 - RF-15	Permitir a inclusão de informações de RelatorioCatalogoInteresseSoftware, contendo os seguintes atributos: o nome, a descrição e os interesses do catálogo, juntamente com as palavras-chave, fontes e prioridade dos mesmos, permitir a geração de um relatório.	PRS de CRUD - Incluir Informação
RF16 - RF-16	Permitir a inclusão de informações de RelatorioRelacionamentoEntreInteresse, contendo os seguintes atributos: uma matriz de dependência e uma matriz de contribuição entre esses interesses, permitir a geração de um relatório.	PRS de CRUD - Incluir Informação
RF17 - RF-17	Permitir a inclusão de informações de RelatorioDocumentoRequisito, contendo os seguintes atributos: nome, a descrição e os requisitos do documento, juntamente com seu tipo e suas dependências com relação a outros requisitos, permitir a geração de um relatório.	PRS de CRUD - Incluir Informação
RF18 - RF-18	Permitir a recuperação de informação de IdentificacaoInteresseSoftware pelo(s) atributo(s) gerar uma lista de requisitos e interesses identificados, destacando-se os interesses principais de cada requisito.	PRS de CRUD - Recuperar Informação
RF19 - RF-19	Permitir a recuperação de informação de Requisito pelo(s) atributo(s) nome do “interesse principal” e do “interesse transversal”, filtragem dos requisitos da lista de requisitos e interesses identificados.	PRS de CRUD - Recuperar Informação
RF20 - RF-20	Permitir a inclusão de informações de Ocorrencia, contendo os seguintes atributos: uma lista de ocorrências a respeito da identificação de interesses do software, conforme definido na abordagem <i>ObasCId</i> , caso necessário.	PRS de CRUD - Incluir Informação
RF21 - RF-21	Permitir a inclusão de informações de InteressePrincipal, contendo os seguintes atributos: interesse principal de um determinado requisito, deve-se ter executado uma unidade de identificação anteriormente.	PRS de CRUD - Incluir Informação
RF22 - RF-22	Permitir a inclusão de informações de ClassificacaoIntressIdentificado, contendo os seguintes atributos: uma matriz de entrelaçamentos existentes entre os interesses identificados, deve se ter executado uma unidade de identificação anteriormente, bem como escolhido o interesse principal de cada requisito e conforme definido na abordagem <i>ObasCId</i> .	PRS de CRUD - Incluir Informação
RF23 - RF-23	Permitir a recuperação de informação de RepositorioCatalogoInteresseSoftware pelo(s) atributo(s) nome ou da descrição do catálogo a ser pesquisado, apenas catálogos públicos podem ser apresentados como resultados de uma busca.	PRS de CRUD - Recuperar Informação
RF24 - RF-24	Permitir a recuperação de informação de DocumentoRequisitoSoftware pelo(s) atributo(s) deve-se informar um trecho do nome ou da descrição do documento de requisitos a ser pesquisado, apenas documentos de requisitos públicos podem ser apresentados como resultados de uma busca.	PRS de CRUD - Recuperar Informação
RF25 - RF-25	Permitir a inclusão de informações de CatalogoInteresseSoftware, contendo os seguintes atributos: novo nome para o documento de requisitos, sua descrição (opcional) e tipo de licença, importação de catálogos de interesses de software desenvolvidos por outros e apenas catálogos de interesses de software públicos podem ser importados.	PRS de CRUD - Incluir Informação

(continua)

Tabela 1.L - Requisitos Funcionais e Não-Funcionais do Sistema de Software OBASCID-TOOL utilizando PRS (continuação).

Código	Descrição do Requisito	PRS Utilizado
RF26 - RF-26	Permitir a inclusão de informações de DocumentoRequisitsSoftware, contendo os seguintes atributos: novo nome para o documento de requisitos, sua descrição (opcional) e tipo de licença, importação de documentos de requisitos desenvolvidos por outros e apenas documentos de requisitos públicos podem ser importados.	PRS de CRUD - Incluir Informação
RF27 - RF-27	Permitir a inclusão de informações de CatalogoInteresseSoftware, contendo os seguintes atributos: novo nome para o catálogo de interesse de software, sua descrição (opcional) e tipo de licença, caso haja nomes de interesses duplicados nos dois catálogos, um novo nome deve ser gerado automaticamente e a união (merge) de dois catálogos de interesses de software.	PRS de CRUD - Incluir Informação
RNF02 - RNF-2	Permitir acesso a AutenticacaoUsuario na funcionalidade de funções do software, apenas quando estiver autenticado no sistema por meio de seu e-mail e senha.	PRS de Segurança
RNF03 - RNF-3	Permitir que o Usuario realize suas tarefas, utilizando a funcionalidade maioria de suas funções do sistema de software, com a condição ser de fácil utilização e tendo passado por um treinamento de 40 (quarenta) minutos.	PRS de Usabilidade

Fonte: Do autor (2020).

APÊNDICE M - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE SISGAC

Tabela 1.M - Requisitos Funcionais e Não-Funcionais do Sistema de Software SisGAC.

Código	Descrição do Requisito
RF01	A ferramenta deve permitir que o usuário consiga realizar seu cadastro. Para isso, o usuário deverá informar nome, CPF, data de nascimento, e-mail, senha e o tipo de usuário (Professor ou Aluno).
RF03	A ferramenta deve permitir que o usuário consiga realizar a alteração dos seus dados pessoais cadastrados, sendo eles, CPF.
RF05	A ferramenta deve possuir um menu de opções para o usuário do tipo Professor, contendo: Matrículas, Disciplinas, Atividades, Avaliações e Solicitações.
RF06	A ferramenta deve possuir um menu de opções para o usuário do tipo Aluno, contendo: Matrícula, Disciplinas, Atividades e Troféus.
RF07	A ferramenta deve permitir que o usuário, como professor, possa cadastrar conjuntos de tarefas. Para isso, o mesmo deverá informar um nome para o conjunto de atividades que deseja cadastrar.
RF08	A ferramenta deve permitir que o usuário, como professor, possa adicionar atividades para o conjunto de atividades cadastrados. Para isso, o mesmo deverá informar um nome para a atividade, prazo de entrega, tipos de troféus, sendo eles, ouro, prata e bronze, que se pode obter ao concluir a atividade e uma descrição.
RF09	A ferramenta deve permitir que o usuário, como professor, possa cadastrar disciplinas. Para isso, o mesmo deverá informar um nome para a disciplina, código e descrição.
RF10	A ferramenta deve permitir que o usuário, como professor, possa adicionar conjunto de atividades para as suas disciplinas.
RF11	A ferramenta deve permitir que o usuário, como professor, possa disponibilizar (tornar visível) uma disciplina, de modo que os alunos possam participar da mesma.
RF12	A ferramenta deve permitir que o usuário, como professor, possa encerrar suas disciplinas, de modo que os alunos matriculados não consigam mais visualizá-la.
RF13	A ferramenta deve permitir que o usuário, como professor, possa aceitar ou recusar as solicitações de matrículas de suas disciplinas, realizadas pelos alunos.
RF14	A ferramenta deve permitir que o usuário, como professor, possa desmatricular alunos de suas disciplinas.
RF15	A ferramenta deve permitir que o usuário, como professor, possa avaliar as atividades realizadas pelos alunos, em suas disciplinas. Para isso, o professor poderá analisar o anexo enviado, bem como a descrição feita pelo aluno e, a partir disso, realizar sua avaliação, aceitando ou recusando a atividade. Além disso, o professor pode classificar a atividade como troféu de ouro, prata e bronze, além de poder escrever uma resposta sobre a atividade do aluno.
RF16	A ferramenta deve permitir que o usuário, como aluno, possa solicitar matrícula para as disciplinas disponibilizadas pelos professores. Para isso, o mesmo deverá selecionar a disciplina desejada e informar uma justificativa para tal matrícula.
RF17	A ferramenta deve permitir que o usuário, como aluno, possa realizar as atividades das disciplinas que ele está matriculado. Para isso, o mesmo deverá selecionar a atividade que deseja realizar, preencher o campo resposta, anexar um arquivo referente a sua atividade e enviar para o professor avaliar.
RF18	A ferramenta deve permitir que o usuário, como aluno, caso tenha uma matrícula recusada pelo professor, possa realizar solicitação de matrícula novamente.
RF19	A ferramenta deve permitir que o usuário, como aluno, caso tenha uma atividade recusada pelo professor, possa realizar novamente a atividade e enviar para o professor avaliá-la novamente, desde que esteja dentro do prazo estabelecido.
RF20	A ferramenta deve permitir que o usuário, como aluno, tenha acesso a uma tela, onde é exibido a contabilização de todos os troféus recebidos após realizar as atividades das disciplinas que está matriculado.
RNF01	A ferramenta deve permitir que o usuário consiga realizar autenticação na mesma, de modo a utilizar sua funcionalidade. Para isso, o usuário deverá informar seu e-mail e senha.
RNF02	A ferramenta deve permitir que o usuário possa encerrar a sua sessão (logout).

Fonte: Do autor (2020).

APÊNDICE N - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE SISGAC UTILIZANDO PRS

Tabela 1.N - Requisitos Funcionais e Não-Funcionais do Sistema de Software SisGAC utilizando PRS.

Código	Descrição do Requisito	PRS Utilizado
RF01 - 1	Permitir a inclusão de informações de Usuario, contendo os seguintes atributos: nome, CPF, data de nascimento, e -mail, senha e o tipo de usuário (Professor ou Aluno), o usuário deverá informar para realizar o cadastro.	PRS de CRUD - Incluir Informação
RF03 - 3	Permitir a alteração de Usuario no(s) seguinte(s) atributo(s) CPF, permitir que o usuário consiga realizar a alteração dos seus dados pessoais cadastrados.	PRS de CRUD - Alterar Informação
RF05 - 5	Permitir a inclusão de informações de Menu, contendo os seguintes atributos: Matrículas, Disciplinas, Atividades, Avaliações e Solicitações, para o usuário do tipo Professor.	PRS de CRUD - Incluir Informação
RF06 - 6	Permitir a inclusão de informações de Menu, contendo os seguintes atributos: Matrícula, Disciplinas, Atividades e Troféus, para o usuário do tipo Aluno.	PRS de CRUD - Incluir Informação
RF07 - 7	Permitir a inclusão de informações de ConjuntoTarefa, contendo os seguintes atributos: nome para o conjunto de atividades, usuário, como professor, possa cadastrar conjuntos de tarefas.	PRS de CRUD - Incluir Informação
RF08 - 8	Permitir a inclusão de informações de AtividadeConjuntoTarefa, contendo os seguintes atributos: nome para a atividade, prazo de entrega, tipos de troféus, sendo eles, ouro, prata e bronze, usuário, como professor, possa adicionar atividades para o conjunto de atividades cadastrados.	PRS de CRUD - Incluir Informação
RF09 - 9	Permitir a inclusão de informações de Disciplina, contendo os seguintes atributos: nome para a disciplina, código e descrição, usuário, como professor, possa cadastrar disciplinas.	PRS de CRUD - Incluir Informação
RF10 - 10	Permitir a inclusão de informações de Disciplina, contendo os seguintes atributos: conjunto de atividades, usuário, como professor.	PRS de CRUD - Incluir Informação
RF11 - 11	Permitir a inclusão de informações de Disciplina, contendo os seguintes atributos: disponibilizar (tornar visível) uma disciplina, usuário, como professor de modo que os alunos possam participar da mesma.	PRS de CRUD - Incluir Informação
RF12 - 12	Permitir a inclusão de informações de Disciplina, contendo os seguintes atributos: encerrar disciplina, usuário, como professor de modo que os alunos matriculados não consigam mais visualizá-la.	PRS de CRUD - Incluir Informação
RF13 - 13	Permitir a inclusão de informações de Disciplina, contendo os seguintes atributos: aceitar ou recusar as solicitações de matrículas de suas disciplinas, usuário, como professor solicitações realizadas pelos alunos.	PRS de CRUD - Incluir Informação
RF14 - 14	Permitir a inclusão de informações de Disciplina, contendo os seguintes atributos: desmatricular alunos de suas disciplinas, usuário, como professor.	PRS de CRUD - Incluir Informação
RF15 - 15	Permitir a inclusão de informações de AtividadeAvaliadaDisciplina, contendo os seguintes atributos: analisar o anexo enviado, bem como a descrição feita pelo aluno e, a partir disso, realizar sua avaliação, aceitando ou recusando a atividade, classificar a atividade como troféu de ouro, prata e bronze, além de poder escrever uma resposta sobre a atividade do aluno, usuário, como professor.	PRS de CRUD - Incluir Informação
RF16 - 16	Permitir a inclusão de informações de SolicitarMatricula, contendo os seguintes atributos: disciplina desejada e informar uma justificativa para tal matrícula, usuário, como aluno, possa solicitar matrícula para as disciplinas disponibilizadas pelos professores.	PRS de CRUD - Incluir Informação
RF17 - 17	Permitir a inclusão de informações de AtividadeDisciplina, contendo os seguintes atributos: atividade que deseja realizar, preencher o campo resposta, anexar um arquivo, usuário, como aluno, possa realizar as atividades das disciplinas que ele está matriculado.	PRS de CRUD - Incluir Informação

(continua)

Tabela 1.N - Requisitos Funcionais e Não-Funcionais do Sistema de Software SisGAC utilizando PRS (continuação).

Código	Descrição do Requisito	PRS Utilizado
RF18 - 18	Permitir a inclusão de informações de Matrícula, contendo os seguintes atributos realizar solicitação de matrícula novamente, usuário, como aluno, caso tenha uma matrícula recusada pelo professor.	PRS de CRUD - Incluir Informação
RF19 - 19	Permitir a inclusão de informações de AtividadeDisciplina, contendo os seguintes atributos realizar novamente a atividade e enviar para o professor avaliá-la novamente, usuário, como aluno, caso tenha uma atividade recusada pelo professor, desde que esteja dentro do prazo estabelecido.	PRS de CRUD - Incluir Informação
RF20 - 20	Permitir a recuperação de informação de Trofeu pelo(s) atributo(s) contabilização de todos os troféus recebidos, usuário, como aluno, tenha acesso a uma tela, onde é exibido a contabilização de todos os troféus recebidos.	PRS de CRUD - Recuperar Informação
RNF01 - RNF-1	Permitir acesso a Usuario na funcionalidade de utilizar as funcionalidades da ferramenta, apenas quando estiver autenticado no sistema deverá informar seu e-mail e senha.	PRS de Segurança

Fonte: Do autor (2020).

APÊNDICE O - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE VEREDAS SOL E LARES

Tabela 1.O - Requisitos Funcionais e Não-Funcionais do Sistema de Software Veredas Sol e Lares.

Código	Descrição do Requisito
RF01 - Manter usuário	O sistema deverá permitir a manutenção dos usuários que terão acesso ao sistema. Inserindo obrigatoriamente as informações de nome, telefone, celular, CPF, e-mail, senha e usina associada.
RF02 - Manter perfis de acesso	O sistema deverá manter os perfis de acesso ao sistema, bem como funções e usuários associados. Informando o usuário e as funções que o usuário terá acesso no sistema.
RF03 - Realizar login	O sistema deverá permitir o login de acesso dos usuários ao sistema realizando o processo de autenticação e autorização conforme perfis de acesso associados. Informando CPF e senha cadastrados.
RF04 - Realizar logout	O sistema deverá permitir ao usuário realizar o processo de logout (sair) do sistema.
RF05 - Lembrar senha	O sistema deverá permitir ao usuário redefinir sua senha quando a mesma for esquecida. O processo será realizado mediante a informação de dados básicos, conferência e geração de senha temporária para criação de nova senha.
RF06 - Alterar senha	O sistema deverá permitir ao usuário alterar a senha de acesso ao sistema mediante conferência de senha atual. Informando CPF, senha atual, nova senha e confirmar nova senha.
RF07 - Registrar ações	Todas as ações realizadas no sistema deverão ser logadas para auditoria, conforme mapeamento. Informando usuário, CPF, data e hora do acesso.
RF08 - Desativar e reativar item	O sistema deverá permitir desativar e reativar itens como usuários e perfis de acesso. Informando qual dos itens o usuário terá acesso. Somente administradores podem realizar essa funcionalidade.
RF10 - Manter UFV	O sistema deverá permitir a manutenção dos dados das UFVs a serem monitoradas pelo sistema. Permitir associar os dados da usina hidráulica a usina fotovoltaica. Informando dados das usinas, tais como, nome da usina, localidade, responsável técnico, potência de geração de energia e data de ativação.
RF11 - Manter planta	O sistema deverá permitir o armazenamento e manutenção de forma hierárquica, das informações dos equipamentos integrantes da planta solar de acordo com suas especificidades. Alguns destes equipamentos compreendem: String, string box, módulo, medidor de fronteira, inversor, transformador e relés de proteção.
RF14 - Gerenciar comandos	Deve ser possível gerenciar as ações de controle que serão geradas pelas funções aplicativos (automatismos) e por demanda do operador. Deve permitir: (i) Para inversores - partir, parar, aumentar ou diminuir potência; e (ii) Para string Box - ligar ou desligar.
RF15 - Exibir diagrama unifilar	Deverá ser apresentado o diagrama unifilar da rede elétrica na qual será conectada a usina solar fotovoltaica. Neste diagrama será exibida a configuração da rede elétrica com os equipamentos de conexão, as grandezas analógicas (potência ativa, reativa e tensão) e o estado (aberto ou fechado) dos disjuntores/chaves seccionadoras. Além das informações relacionadas será exibida ainda a geração totalizada da usina fotovoltaica no ponto de conexão.
RF16 - Manter Contrato	O sistema deverá permitir o cadastro dos dados básicos dos contratos associados a cada usina para controle. Os usuários só terão acesso aos dados das usinas associadas ao mesmo. As informações são de nome da usina, data de ativação responsável técnico e potência de geração de energia.
RF17 - Manter parâmetros gerais	O sistema deverá permitir a manutenção dos parâmetros gerais do sistema. Tais como, limite máximo e mínimo dos inversores, limite de geração de energia, valores de temperatura e valores de irradiação solar.
RF20 - Visualizar painel principal	O sistema deverá permitir a visualização do painel principal com o resumo dos principais dados gerados pelo sistema para monitoramento. Deve exibir: (i) Previsão de geração para a próxima hora; (ii) Potência total; (iii) Irradiância; (iv) Produção total; (v) Economia de carbono; (vi) Quantidade de famílias beneficiadas; (vii) Temperatura ambiente; (viii) Temperatura módulo de referência; (ix) Dados meteorológicos; (x) Alertas; (xi) Alarmes; (xii) Previsão horária; (xiii) Energia efetivamente gerada; (xiv) Previsão de meteorologia; e (xv) Dados meteorológicos.

(continua)

Tabela 1.O - Requisitos Funcionais e Não-Funcionais do Sistema de Software Veredas Sol e Lares (continuação).

Código	Descrição do Requisito
RF21 - Visualizar esquemático de processo da usina	O sistema deverá permitir a visualização do esquemático de processo da usina que contempla a exibição dos principais equipamentos que compõem a planta, como: (i) Strings; (ii) String box; (iii) Inversores; (iv) Módulos; e (v) Medidores. As funções de manutenção e operação poderão ser realizadas a partir desta interface.
RF22 - Gerenciar manutenções	Deve permitir colocar ou retirar um equipamento em manutenção, bem como visualizar de forma gráfica todos os equipamentos em manutenção através da utilização de etiquetas. Será possível registrar dados básicos da manutenção, como: período de manutenção, responsável, tipo de manutenção, observações gerais.
RF23 - Acompanhar manutenções	Deve permitir ao operador acompanhar o estado de cada equipamento da planta que se encontra em manutenção através do histórico de ações.
RF24 - Gerar curva I/V	Deve permitir a geração de curva I/V (corrente X tensão) característica de dispositivos fotovoltaicos.
RF30 - Realizar balanço de energia após o faturamento	O sistema deverá realizar o balanço de energia após o faturamento (gerada X consumida).
RF31 - Calcular eficiência por inversor	O sistema deverá calcular a eficiência de geração por inversor (irradiância x energia gerada x potência instalada).
RF32 - Calcular energia estimada	O sistema deverá calcular utilizando estimativas a quantidade de energia a gerar.
RF33 - Calcular energia gerada	Deverá ser possível identificar a quantidade de energia gerada por hora, dia, mês, ano, total (desde o início da operação).
RF34 - Calcular desempenho	O sistema deverá realizar o cálculo de desempenho do sistema fotovoltaico.
RF40 - Gerenciar alarmes e eventos	O sistema deverá efetuar a gestão dos alarmes das funções de supervisão e controle e proteção, ordenando os alarmes de forma cronológica.
RF41 - Categorizar alarme	O sistema deverá realizar a categorização do alarme em função do grau de criticidade e natureza do alarme.
RF42 - Alertar operadores	O sistema deverá enviar alertas relativos aos alarmes para os operadores da planta, indicando o valor percentual que a violação representa.
RF43 - Consultar sequência de eventos	Deverá ser permitida a visualização dos eventos ocorridos no sistema de acordo com a cronologia de acontecimentos.
RF44 - Configurar limites	O sistema deverá permitir a parametrização de limites para as grandezas a serem monitoradas. Estes limites serão enviados aos equipamentos que farão uso destes limites para disparar eventos caracterizados como alarmes e alertas.
RF45 - Gerar alarme/evento violação de limite	O sistema deverá gerar alarmes/eventos relativos à violação dos limites relativos as grandezas telemedidas, elétricas e meteorológicas de acordo com os limites parametrizados.
RF46 - Monitorar limites	O sistema deverá realizar o monitoramento contínuo das grandezas telemedidas, elétricas e meteorológicas através da comparação dos valores adquiridos em tempo real em relação aos limites estabelecidos.
RF47 - Verificar limites	O sistema deverá verificar os limites configurados para cada equipamento para gerar os alarmes e alertas necessários.
RF50 - Armazenar dados históricos	Deverá haver armazenamento dos dados históricos (alarmes, eventos, tendências e dados estatísticos). A função de arquivamento de dados deve considerar os seguintes períodos de tempo: (i) Alarmes: 30 dias; (ii) SOE: 30 dias; (iii) Tendências Históricas: 7dias; (iv) Valores Médios Horários: 7dias; (v) Valores Médios Diários: 7dias; e (vi) Dados Meteorológicos: 12 meses.
RF51- Receber e armazenar leituras	O sistema deverá receber e armazenar as leituras provenientes das plantas.
RF52 - Enviar atualizações de configurações	O sistema deverá enviar as mensagens de atualizações de configurações de serviços, modelos e equipamentos a serem utilizados na planta.
RF53 - Informar equipamentos novos	O sistema deverá gerar e enviar mensagens sempre que um novo equipamento for cadastrado.
RF60 - Visualizar dados históricos	Esta função deve possibilitar a visualização das evoluções das medições em função do tempo (janela de curto prazo de 30 a 50 minutos) de grandezas telemedidas e calculadas. Grandezas telemedidas: (i) Tensão e corrente CC (módulo); (ii) Tensão e corrente CA (inversor); (iii) Irradiância (módulo); (iv) Temperatura (módulo); (v) Velocidade do vento; (b) Grandezas calculadas: (i) Geração do módulo (potência ativa); (ii) Geração inversor (potência ativa e reativa); e (iii) Fator de potência.

(continua)

Tabela 1.O - Requisitos Funcionais e Não-Funcionais do Sistema de Software Veredas Sol e Lares (continuação).

Código	Descrição do Requisito
RF71 - Gerar relatório de sequência de eventos	O sistema deverá gerar o relatório de sequência de eventos.
RF72 - Gerar relatório de dados climatológicos	O sistema deve gerar o relatório de dados climatológicos medidos pelas estações meteorológicas da planta e eletrocentro.
RF73 - Gerar relatório desempenho de planta	O sistema deverá permitir a geração do relatório do desempenho da planta.
RF74 - Exportar relatórios	O sistema deverá permitir a exportação dos relatórios para arquivos no formato pdf ou xls (excel).
RF75 - Gerar relatório de eficiência de inversor	O sistema deverá gerar o relatório de eficiência de inversor, permitindo apurar valores como média diária, máxima e mínima performance no dia.
RF80 - Calcular disponibilidade do sistema de comunicação	Será calculada a disponibilidade do sistema de comunicação entre os equipamentos da planta e o supervisor.
RF81 - Monitorar falhas de comunicação	Esta função tem como objetivo efetuar o monitoramento das falhas de comunicação identificadas no processo de transmissão de dados.
RF82 - Gerenciar comunicação de equipamento	Será responsável também pelo gerenciamento das comunicações dos equipamentos, levando-se em consideração os protocolos que permitem a comunicação entre os equipamentos do sistema.
RF100 - Classificar alarmes	Deverá possibilitar uma análise a partir da correlação simples de alarmes gerados pelos diversos módulos/funções, de modo a diagnosticar possíveis problemas de operação e manutenção que podem estar acontecendo na planta solar.
RF101 - Realizar diagnóstico	Deverá permitir realizar o diagnóstico de falhas operativas e de manutenção, esta função deve evitar avalanches de alarmes para o operador da usina. Em processos críticos da operação da planta solar é de fundamental importância que o operador receba alarmes que possam ajudar no processo de tomada de decisão para normalização da operação da usina.
RF102 - Tratar alarmes	Deverá permitir realizar o tratamento dos alarmes gerados nos processos críticos da operação da planta solar.
RF110 - Calcular performance	O sistema deverá realizar o cálculo de performance da planta, considerando as diversas etapas do processo.
RF111 - Realizar controle entre geração UFV X concessionária	Deve permitir efetuar o controle de tensão e fator de potência no Ponto de Conexão, ou seja, a interligação da geração solar com a concessionária. Esta função deve considerar os seguintes controles: (i) Controle em malha aberta do fator de potência no ponto de conexão; (ii) Controle proporcional integral (PI) em malha fechada do fator de potência no ponto de conexão, baseado no fator de potência medido; (iii) Limitação de potência gerada, em malha aberta e fechada; e (iv) Integração do controle de inversores e banco de capacitores para aprimorar a qualidade do fornecimento de energia.
RF120 - Calcular indisponibilidade programada (IP)	O sistema deverá calcular automaticamente a indisponibilidade programada que corresponde ao tempo em que a operação da UFV é interrompida, devido à execução de manutenção programada. A IP compõe uma componente da rede em que a UFV é conectada a outra componente de indisponibilidade da própria UFV. A principal fonte de paradas da UFV para manutenção são os inversores.
RF121 - Calcular indisponibilidade Forçada - TEIF	Deverá ser possível calcular automaticamente a indisponibilidade forçada (TEIF) que representa o percentual do tempo em que a usina não está apta a operar, devido à ocorrência de falha ou interrupção de emergência, em condições não programadas, abatidas as paradas programadas.
RF122 - Calcular economia de carbono	O sistema deverá calcular automaticamente a redução das emissões de dióxido de carbono (CO ₂) obtidas através da geração de energia solar pela planta.
RF123 - Calcular número de famílias beneficiadas	O sistema deverá calcular o número de famílias beneficiadas pela geração da usina fotovoltaica.
RF140 - Monitorar inequações	O sistema deverá possibilitar ao operador montar uma inequação, de forma on-line e através de telas adequadas, envolvendo grandezas telemedidas da planta solar, com utilização dos diversos operadores lógicos e de comparação. Em caso dos valores limites estabelecidos serão enviados alarmes para o operador da planta solar. Esta função será de grande importância para o monitoramento em tempo real da planta.

(continua)

Tabela 1.O - Requisitos Funcionais e Não-Funcionais do Sistema de Software Veredas Sol e Lares (continuação).

Código	Descrição do Requisito
RF150 - Realizar previsão de geração diária e semanal	O sistema deverá utilizar as informações meteorológicas para calcular o valor da geração que poderá ser produzida na planta solar. Poderão ser utilizadas técnicas de inteligência computacionais e os métodos estatísticos para efetuar essa previsão da geração.
RF151 - Realizar previsão de geração de curto prazo (horária)	O sistema deverá calcular o valor da energia gerada que poderá ser gerado em função das informações climatológicas obtidas através da estação meteorológica utilizando projeções matemáticas.
RNF01 - Arquitetura	O sistema de monitoramento e controle contará com dois subsistemas: (i) Sistema de aquisição; e (ii) Sistema de monitoramento.
RNF02 - Usabilidade	O sistema deverá ser de fácil utilização, permitindo que seus usuários, tendo passado por um treinamento, consigam executar corretamente a maioria de suas funções de monitoramento e controle.
RNF03 - Portabilidade	O sistema deverá ter uma interface de com todas as funções do software tendo que ser responsiva, de forma que os elementos da mesma se adaptem a dispositivos com telas menores e/ ou maiores (tais como, smartphones, tablets e smart tvs).
RNF04 - Gerar alarmes	O sistema deverá disponibilizar alarmes de equipamentos críticos aos funcionamentos da geração de energia fotovoltaica com situação crítica de tempos em tempos que não deve exceder 4 segundos.
RNF05 - Disponibilidade do sistema	O sistema deverá estar disponível 99,9% do tempo.

Fonte: Do autor (2020).

APÊNDICE P - REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS DO SISTEMA DE SOFTWARE VEREDAS SOL E LARES UTILIZANDO PRS

Tabela 1.P - Requisitos Funcionais e Não-Funcionais do Sistema de Software Veredas Sol e Lares utilizando PRS.

Código	Descrição do Requisito	PRS Utilizado
RF01 - Manter usuário	Permitir a inclusão de informações de Usuario, contendo os seguintes atributos: nome, telefone, celular, CPF, e-mail, senha e usina associada, inserindo obrigatoriamente.	PRS de CRUD - Incluir Informação
RF02 - Manter perfis de acesso	Permitir a inclusão de informações de PerfilAcesso, contendo os seguintes atributos: usuário, funções e usuários associados, informando obrigatoriamente o usuário e as funções que o usuário terá acesso no sistema.	PRS de CRUD - Incluir Informação
RF03 - Realizar login	Permitir acesso a Usuario na funcionalidade de ao sistema, apenas quando estiver autenticado no sistema conforme perfis de acesso associados e informando CPF e senha cadastrados.	PRS de Segurança
RF06 - Alterar senha	Permitir a alteração de Usuario no(s) seguinte(s) atributo(s) CPF, senha atual, nova senha e confirmar nova senha, mediante conferência de senha atual.	PRS de CRUD - Alterar Informação
RF07 - Registrar ações	Permitir a inclusão de informações de HistoricoAcoes, contendo os seguintes atributos: usuário, CPF, data e hora do acesso, ações realizadas no sistema.	PRS de CRUD - Incluir Informação
RF08 - Desativar e reativar item	Permitir a alteração de Usuario e PerfilAcesso no(s) seguinte(s) atributo(s) desativar e reativar itens, informando qual dos itens o usuário terá acesso e somente administradores podem realizar essa funcionalidade.	PRS de CRUD - Alterar Informação
RF10 - Manter UFV	Permitir a inclusão de informações de UFV, contendo os seguintes atributos: nome da usina, localidade, responsável técnico, potência de geração de energia e data de ativação, permitir associar os dados da usina hidráulica a usina fotovoltaica.	PRS de CRUD - Incluir Informação
RF11 - Manter planta	Permitir a inclusão de informações de FormaHierarquicaEquipamento, contendo os seguintes atributos: String, string box, módulo, medidor de fronteira, inversor, transformador e relés de proteção, equipamentos integrantes da planta solar de acordo com suas especificidades.	PRS de CRUD - Incluir Informação
RF15 - Exibir diagrama unifilar	Permitir a recuperação de informação de DiagramaUnifilar pele(s) atributo(s) configuração da rede elétrica com os equipamentos de conexão, as grandezas analógicas (potência ativa, reativa e tensão) e o estado (aberto ou fechado) dos disjuntores/chaves seccionadoras, geração totalizada da usina fotovoltaica no ponto de conexão, rede elétrica na qual será conectada a usina.	PRS de CRUD - Recuperar Informação
RF16 - Manter Contrato	Permitir a inclusão de informações de ContratosAssociados, contendo os seguintes atributos: nome da usina, data de ativação responsável técnico e potência de geração de energia, os usuários só terão acesso aos dados das usinas associadas ao mesmo.	PRS de CRUD - Incluir Informação
RF17 - Manter parâmetros gerais	Permitir a inclusão de informações de ParametrosGerais, contendo os seguintes atributos: limite máximo e mínimo dos inversores, limite de geração de energia, valores de temperatura e valores de irradiação solar, parâmetros gerais do sistema.	PRS de CRUD - Incluir Informação

(continua)

Tabela 1.P - Requisitos Funcionais e Não-Funcionais do Sistema de Software Veredas Sol e Lares utilizando PRS (continuação).

Código	Descrição do Requisito	PRS Utilizado
RF20 - Visualizar painel principal	Permitir a recuperação de informação de PainelPrincipal pelo(s) atributo(s) (i) Previsão de geração para a próxima hora; (ii) Potência total; (iii) Irradiância; (iv) Produção total; (v) Economia de carbono; (vi) Quantidade de famílias beneficiadas; (vii) Temperatura ambiente; (viii) Temperatura módulo de referência; (ix) Dados meteorológicos; (x) Alertas; (xi) Alarmes; (xii) Previsão horária; (xiii) Energia efetivamente gerada; (xiv) Previsão de meteorologia; e (xv) Dados meteorológicos, resumo dos principais dados gerados pelo sistema para monitoramento.	PRS de CRUD - Recuperar Informação
RF21 - Visualizar esquemático de processo da usina	Permitir a recuperação de informação de EsquemáticoProcessoUsina pelo(s) atributo(s) (i) Strings; (ii) String box; (iii) Inversores; (iv) Módulos; e (v) Medidores, exibição dos principais equipamentos que compõem a planta e as funções de manutenção e operação poderão ser realizadas a partir desta interface.	PRS de CRUD - Recuperar Informação
RF22 - Gerenciar manutenções	Permitir a inclusão de informações de Equipamento, contendo os seguintes atributos: em manutenção, registrar dados básicos da manutenção, como período de manutenção, responsável, tipo de manutenção, observações gerais. visualizar de forma gráfica todos os equipamentos em manutenção através da utilização de etiquetas.	PRS de CRUD - Incluir Informação
RF23 - Acompanhar manutenções	Permitir a recuperação de informação de HistoricoManutencao pelo(s) atributo(s) equipamento, equipamento da planta que se encontra em manutenção.	PRS de CRUD - Recuperar Informação
RF24 - Gerar curva I/V	Permitir a recuperação de informação de Equipamento pelo(s) atributo(s) curva I/V (corrente X tensão), característica de dispositivos fotovoltaicos.	PRS de CRUD - Recuperar Informação
RF30 - Realizar balanço de energia após o faturamento	Permitir a inclusão de informações de BalancoEnergia, contendo os seguintes atributos: faturamento (gerada X consumida), após o faturamento.	PRS de CRUD - Incluir Informação
RF41 - Categorizar alarme	Permitir a recuperação de informação de Alarme pelo(s) atributo(s) grau de criticidade e natureza do alarme, categorização do alarme em função do grau de criticidade e natureza do alarme.	PRS de CRUD - Recuperar Informação
RF43 - Consultar sequência de eventos	Permitir a recuperação de informação de Evento pelo(s) atributo(s) eventos, ocorridos no sistema de acordo com a cronologia de acontecimentos.	PRS de CRUD - Recuperar Informação
RF45 - Gerar alarme/evento violação de limite	Permitir a recuperação de informação de AlarmesEventos pelo(s) atributo(s) grandezas telemedidas, elétricas e meteorológicas de acordo com os limites parametrizados, relativos à violação dos limites relativos.	PRS de CRUD - Recuperar Informação
RF46 - Monitorar limites	Permitir a recuperação de informação de GrandezasTelemedidas e EletricasMeteorológicas pelo(s) atributo(s) valores adquiridos em tempo real em relação aos limites estabelecidos, monitoramento contínuo.	PRS de CRUD - Recuperar Informação
RF47 - Verificar limites	Permitir a recuperação de informação de Equipamento pelo(s) atributo(s) limites configurados, gerar os alarmes e alertas necessários.	PRS de CRUD - Recuperar Informação
RF50 - Armazenar dados históricos	Permitir a inclusão de informações de Alarme, Evento, Tendencia e DadosEestatisticos, contendo os seguintes atributos: dados históricos, considerar os seguintes períodos de tempo: (i) Alarmes: 30 dias; (ii) SOE: 30 dias; (iii) Tendências Históricas: 7dias; (iv) Valores Médios Horários: 7dias; (v) Valores Médios Diários: 7dias; e (vi) Dados Meteorológicos: 12 meses.	PRS de CRUD - Incluir Informação

(continua)

Tabela 1.P - Requisitos Funcionais e Não-Funcionais do Sistema de Software Veredas Sol e Lares utilizando PRS (continuação).

Código	Descrição do Requisito	PRS Utilizado
RF60 - Visualizar dados históricos	Permitir a recuperação de informação de MedicoesGrandezasTelemidas e MedicoesGrandezasCalculadas pelo(s) atributo(s) Grandezas telemidas: (i) Tensão e corrente CC (módulo); (ii) Tensão e corrente CA (inversor); (iii) Irradiância (módulo); (iv) Temperatura (módulo); (v) Velocidade do vento; (b) Grandezas calculadas: (i) Geração do módulo (potência ativa); (ii) Geração inversor (potência ativa e reativa); e (iii) Fator de potência, evoluções das medições em função do tempo (janela de curto prazo de 30 a 50 minutos).	PRS de CRUD - Recuperar Informação
RF71 - Gerar relatório de sequência de eventos	Permitir a recuperação de informação de Evento pelo(s) atributo(s) sequência de eventos, gerar o relatório.	PRS de CRUD - Recuperar Informação
RF72 - Gerar relatório de dados climatológicos	Permitir a recuperação de informação de EstacaoMeteorologicas pelo(s) atributo(s) dados climatológicos medidos, gerar o relatório.	PRS de CRUD - Recuperar Informação
RF73 - Gerar relatório desempenho de planta	Permitir a recuperação de informação de Planta pelo(s) atributo(s) desempenho da planta, geração do relatório.	PRS de CRUD - Recuperar Informação
RF75 - Gerar relatório de eficiência de inversor	Permitir a recuperação de informação de Inversor pelo(s) atributo(s) média diária, máxima e mínima performance no dia, gerar o relatório de eficiência.	PRS de CRUD - Recuperar Informação
RNF02 - Usabilidade	Permitir que o Usuario realize suas tarefas, utilizando a funcionalidade a maioria de suas funções de monitoramento e controle ser de fácil utilização do sistema de software, com a condição tendo passado por um treinamento.	PRS de Usabilidade
RNF03 - Portabilidade	Permitir que o Veredas Sol e Lares seja executado em diferentes hardwares, softwares, sistemas operacionais ou ambientes de uso, com a condição todas as funções do software tendo que ser responsiva, de forma que os elementos da mesma se adaptem a dispositivos com telas menores e/ ou maiores (tais como, smartphones, tablets e smart tvs) mantendo a capacidade de eficiência e eficácia.	PRS de Portabilidade
RNF04 - Gerar alarmes	Permitir acesso à funcionalidade Alarme do sistema de software e o tempo de resposta não deve exceder 4 segundos para alarmes críticos aos funcionamentos da geração de energia fotovoltaica.	PRS de Desempenho

Fonte: Do autor (2020).