



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE QUIXADÁ
CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE

ERIKA GONZAGA DA SILVA

**CATÁLOGOS DE CARACTERÍSTICAS DE QUALIDADE DE SOFTWARE: UM
MAPEAMENTO SISTEMÁTICO**

QUIXADÁ

2019

ERIKA GONZAGA DA SILVA

CATÁLOGOS DE CARACTERÍSTICAS DE QUALIDADE DE SOFTWARE: UM
MAPEAMENTO SISTEMÁTICO

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Engenharia de Software
do Campus de Quixadá da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Engenharia de Software.

Orientadora: Prof^a. M^a. Rainara Maia
Carvalho

QUIXADÁ

2019

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- S579c Silva, Erika Gonzaga da.
Catálogos de características de qualidade de software : um mapeamento sistemático / Erika Gonzaga da Silva. – 2019.
56 f. : il. color.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Engenharia de Software, Quixadá, 2019.
Orientação: Profa. Ma. Rainara Maia Carvalho.
1. Requisitos não funcionais (engenharia de sistemas). 2. Catálogos. 3. Mapeamento sistemático. I.
Título.

CDD 005.1

ERIKA GONZAGA DA SILVA

CATÁLOGOS DE CARACTERÍSTICAS DE QUALIDADE DE SOFTWARE: UM
MAPEAMENTO SISTEMÁTICO

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Engenharia de Software
do Campus de Quixadá da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Engenharia de Software.

Aprovada em: __/__/____.

BANCA EXAMINADORA

Prof^a. M^a. Rainara Maia Carvalho (Orientadora)
Universidade Federal do Ceará (UFC)

Prof^a. Dr^a. Jéssyka Flavyanne Ferreira Vilela
Universidade Federal de Pernambuco (UFPE)

Prof. Dr. Ismayle de Sousa Santos
Universidade Federal do Ceará (UFC)

A Deus.

À minha mãe e ao meu tio, por acreditarem no meu potencial e me dar todo suporte necessário para continuar.

AGRADECIMENTOS

À Professora Rainara por me orientar neste trabalho, pela sua dedicação e paciência, e principalmente, pelos conselhos.

À minha família, em especial, minha mãe Rozineide, minha avó Maria, meus irmãos Jefferson, Jessica e Paulo Sergio, meus tios, Francisco, Ivo, Odair, Tarciana, Zildene e meu primo Alexandre.

Aos amigos que fiz durante minha graduação, em especial Alex, Arina, Cezar, Felipe, Igor, Juliana, Leo, Thomás, Vinicius e Yuri. E aos professores, Carla, Diana, Diego, Livia e Paulyne. Aos servidores em especial, Natália e Venicio.

Aos amigos que fiz no PET-TI, em especial, Júlio, Lucas, Raul, Rayanne e Tiago.

Aos meus amigos do ensino médio, em especial, Érika Farias, Erika Façanha, Iury, Jackson, Jardel, Jordão, Linconl e Raquel. E os professores, em especial, Charles, Gabriela, Gilmário e Wellington.

Às minhas amigas do ensino fundamental Gessika e Mirna.

“Faça a si mesmo a seguinte pergunta: você tem força, coragem e sabedoria para alcançar um novo nível de grandiosidade?”

(Hotel Valhala - Rick Riordan)

RESUMO

Desenvolver um software de qualidade envolve vários fatores, um deles é a elicitación de requisitos não-funcionais (RNFs). Esses requisitos são conhecidos pela sua alta facilidade de se relacionar entre si, positiva e negativamente. Um problema comum é o comprometimento de desenvolvedores em satisfazer certos RNFs que se prejudicam. Outro problema também, é que os RNFs podem ser interpretados de formas diferentes, por pessoas diferentes em contextos diferentes. Uma solução encontrada na literatura é o uso de catálogos de RNFs, por isso é importante o conhecimento sobre os catálogos existentes. Este trabalho tem o objetivo de entender o estado da arte dos catálogos existentes. Para isso um mapeamento sistemático foi realizado, onde foi descoberto 102 catálogos diferentes que apoiam 86 RNFs, e que 13 são do domínio da saúde. Dentre os catálogos encontrados 50 são representados por Gráfico de Interdependência Softgoal e metade deles (25) apresentam subcaracterísticas, métodos e correlações. Os resultados encontrados fornecem contribuições para a indústria (possível redução do tempo de elicitación de requisitos com uso de catálogos) e oportunidades de estudos acadêmicos (comparar os tipos de representações dos catálogos).

Palavras-chave: Requisito não-funcional. Catálogo. Mapeamento sistemático

ABSTRACT

Developing quality software involves several factors, one of which is the elicitation of non-functional requirements (NFRs). These requirements are known for their high ease of relating to each other, positively and negatively. A common problem is the commitment of developers to satisfy certain NFRs that are harmed. Another problem is that NFRs can be interpreted in different ways by different people in different contexts. One solution found in the literature is the use of NFR catalogs, so it is important to know about existing catalogs. This work aims to understand the state of the art of existing catalogs. For this, a systematic mapping was carried out, where 102 different catalogs were found that support 86 NFRs, and that 13 are of the health domain. Among the catalogs found 50 are represented by Softgoal Interdependence Graph and half of them (25) present subcharacteristics, methods and correlations. The results provide contributions to industry (possible reduction of elicitation time using catalogs) and academic study opportunities (compare types of catalog representations).

Keywords: Non-Functional Requirement. Catalog. Systematic Mapping

LISTA DE FIGURAS

Figura 1 – Modelo de qualidade em uso	17
Figura 2 – Modelo de qualidade do produto de software	17
Figura 3 – Matriz com interações entre características de qualidade	19
Figura 4 – SIG de segurança	20
Figura 5 – Processo do mapeamento sistemático	26
Figura 6 – Aplicação dos critérios nos filtros	29
Figura 7 – Filtro dos artigos das bases scopus e web of science	31
Figura 8 – Filtro dos artigos do <i>snowballing</i>	32
Figura 9 – Filtro dos artigos do WER	32
Figura 10 – Filtro geral	33
Figura 11 – Quantidade de artigos por base	34
Figura 12 – Quantidade de artigos por ano	35
Figura 13 – Quantidade de artigos por tipo de pesquisa	35
Figura 14 – Quantidade de artigos por tipo de publicação	36
Figura 15 – Pesquisadores com mais artigos publicados	36
Figura 16 – Requisitos mais citados	37
Figura 17 – Quantidade de catálogos por ano	39
Figura 18 – Quantidade de catálogos por pesquisador	39

LISTA DE TABELAS

Tabela 1 – Características de qualidade externa	18
Tabela 2 – Características de qualidade interna	18
Tabela 3 – <i>String</i> de busca usando a estratégia PICO	28
Tabela 4 – Tipos dos catálogos	37
Tabela 5 – Domínio dos catálogos	38
Tabela 6 – Tipos de representação dos catálogos	40
Tabela 7 – Lista de artigos - parte 1	48
Tabela 8 – Lista de artigos - parte 2	49
Tabela 9 – Lista de artigos - parte 3	50
Tabela 10 – Lista de artigos - parte 4	51
Tabela 11 – Lista de catálogos - parte 1	52
Tabela 12 – Lista de catálogos - parte 2	53
Tabela 13 – Lista de catálogos - parte 3	54
Tabela 14 – Características de qualidade e a quantidade de citações	55

LISTA DE QUADROS

Quadro 1 – Tipo de catálogo vs tipo de representação	40
Quadro 2 – Tipo de correlação por nível de correlação	41

SUMÁRIO

1	INTRODUÇÃO	13
2	OBJETIVOS	15
2.1	Objetivo Geral	15
2.2	Objetivos Específicos	15
3	FUNDAMENTAÇÃO TEÓRICA	16
3.1	Definições e Classificações de Características de Qualidade	16
3.2	Interação entre Características de Qualidade	17
3.3	Catálogos para apoiar Características de Qualidade	19
4	TRABALHOS RELACIONADOS	21
4.1	<i>Managing quality requirements: A systematic review</i>	21
4.2	<i>How to develop usability heuristics: A systematic literature review</i>	23
5	METODOLOGIA	26
5.1	Planejamento	26
5.1.1	<i>Definição do objetivo</i>	27
5.1.2	<i>Definição das questões de pesquisa</i>	27
5.1.3	<i>Definição da string de busca</i>	27
5.1.4	<i>Definição das bases</i>	28
5.1.5	<i>Definição dos critérios de seleção</i>	29
5.1.6	<i>Definição dos filtros</i>	29
5.1.7	<i>Teste da string</i>	29
5.2	Condução	30
5.2.1	<i>Busca nas bases de dados eletrônicas</i>	30
5.2.2	<i>Snowballing</i>	31
5.2.3	<i>Busca manual no WER</i>	32
6	RESULTADOS E DISCUSSÕES	34
6.1	RQ 1: Quantos artigos foram selecionados por base de dados?	34
6.2	RQ 2: Quais catálogos existem na literatura?	36
6.3	RQ 3: Como os catálogos são representados?	39
6.4	Ameaças à validade	41
7	CONCLUSÕES E TRABALHOS FUTUROS	44

REFERÊNCIAS	45
APÊNDICES	48
APÊNDICE A – Tabelas referentes à RQ 1	48
APÊNDICE B – Tabelas referentes à RQ 2	52
APÊNDICE C – Tabela referente à RQ 2.2	55
APÊNDICE D – <i>Links</i>	56
ANEXOS	56

1 INTRODUÇÃO

O mundo está cada vez mais informatizado, empresas utilizam sistemas para melhorar seus negócios, pessoas utilizam aplicativos para comunicação e até mesmo para monitorar atividades físicas e informações sobre sua saúde. Dessa forma, o avanço da tecnologia proporciona cada vez mais um ambiente acessível. Por isso, as empresas de desenvolvimento de software se preocupam em desenvolver sistemas que serão úteis para seus usuários, desde softwares para comunicação até jogos para diversão.

Para desenvolver softwares de qualidade, é necessário entender bem o objetivo do que está construindo. Para isso, a equipe de desenvolvimento elicita os requisitos do produto, pois conhecê-los é uma atividade básica para produzir um software (WIEGERS; BEATTY, 2013). Os requisitos podem ser funcionais ou não-funcionais.

Na prática, os requisitos funcionais (RFs), que definem as funcionalidades de um software, recebem mais atenção (MAIRIZA; ZOWGHI, 2010), porém os requisitos não-funcionais (RNFs), que caracterizam restrições do funcionamento/comportamento do sistema, são tão necessários quanto os funcionais para alcançar o sucesso de um projeto (MAIRIZA *et al.*, 2010). Um tipo de RNF é a Característica de Qualidade (WIEGERS; BEATTY, 2013), tais como Desempenho, Confiabilidade e Segurança.

As características de qualidade podem ser consideradas fatores críticos, pois o mal entendimento delas pode gerar sistemas que não satisfazem os clientes. Dessa forma, é importante realizar uma boa elicitação, análise, validação e especificação delas, o que não é fácil devido à diversos problemas, tais como: subjetividade das características (o entendimento de uma característica pode não ser o mesmo para todos envolvidos), variedade de implementação que muitas vezes é entendida apenas por um especialista e grande facilidade que esse tipo de requisito tem de interagir uns com os outros (MAIRIZA; ZOWGHI, 2010).

Essa interação pode ser positiva, que é conhecida como sinergia (onde há colaboração de uma característica em outra) ou negativa, que é conhecida como conflito (onde uma característica impacta negativamente em outra). Como exemplo, Segurança impacta negativamente no Desempenho, enquanto aumenta a Confiabilidade (WIEGERS; BEATTY, 2013).

Dessa forma, é importante que o engenheiro de requisitos saiba dessas interações para não se comprometer com o cliente em relação a características que são conflitantes (WIEGERS; BEATTY, 2013). Uma solução para ajudar os desenvolvedores nessa tarefa é o uso de catálogos de RNFs (CYSNEIROS, 2007).

Na literatura existem diferentes representações desses catálogos, eles podem ser feitos por meio de lista, matriz e gráfico. Chung *et al.* (2012) utilizam o Gráfico de Interdependência de Softgoals (do inglês, *Softgoal Interdependency Graphs* - SIGs) para representar características de qualidade e suas interdependências. Egyed e Grunbacher (2004) geraram uma lista reduzida e ponderada de possíveis conflitos e cooperações entre RNFs, Mairiza e Zowghi (2010) construíram um catálogo de conflitos entre requisitos não-funcionais em forma de matriz.

Vários catálogos têm sido propostos como os de Wieggers e Beatty (2013), Egyed e Grunbacher (2004), Afreen *et al.* (2016), Boehm (2015), Tabassum *et al.* (2014), Horikoshi *et al.* (2012), Coutinho *et al.* (2012), Khatter e Kalia (2011) com essas representações. Diferentes características de qualidade, soluções e interações são abordadas nesses catálogos e podem também ser específicas de domínios (e.g., aplicações web).

Existe uma necessidade de entender melhor como está o estado da arte sobre esse assunto. Até o momento, não foi encontrado um trabalho que reúna estudos sobre esses catálogos, de forma que apresente para o leitor um compilado sobre a área de catálogos de RNFs.

Este trabalho está dividido da seguinte forma: no Capítulo 2 são apresentados os objetivos gerais e específicos. No Capítulo 3 os conceitos básicos para o seu entendimento. Trabalhos relacionados são apresentados no Capítulo 4. A metodologia seguida para execução desta pesquisa é exposta no Capítulo 5. Os Capítulos 6 e 7 apresentam os resultados encontrados e as conclusões, respectivamente.

2 OBJETIVOS

2.1 Objetivo Geral

Este trabalho tem como objetivo geral realizar um estudo para compreender o estado atual da literatura sobre catálogos de características de qualidade.

2.2 Objetivos Específicos

Para fazer esse estudo da literatura, o método *mapeamento sistemático* da literatura será utilizado (PETERSEN *et al.*, 2008). Tal método tem como objetivo fornecer uma visão geral de uma área de pesquisa e identificar a quantidade, tipo e resultados encontrados nela (PETERSEN *et al.*, 2008). Assim, os objetivos específicos desse trabalho são:

1. Identificar os catálogos existentes;
2. Identificar as características abordadas nos catálogos; e
3. Analisar como os catálogos e as correlações são representadas.

3 FUNDAMENTAÇÃO TEÓRICA

Assegurar que os sistemas funcionem da melhor maneira possível é tão importante quanto oferecer as funcionalidades corretas. Para funcionar da melhor maneira, os sistemas devem considerar as características de qualidade durante o seu desenvolvimento. Esta seção apresenta os principais conceitos relacionados às características de qualidade. Primeiro, diferentes definições e classificações de características de qualidade são apresentadas. Depois, o conceito de interação entre características de qualidade é explicado. Após isso, catálogos de apoio às características de qualidade são explicados.

3.1 Definições e Classificações de Características de Qualidade

De acordo com a Iso/Iec (2005), característica de qualidade de software é a categoria de atributos de qualidade de software que se baseia na qualidade do software. Essas características de qualidade podem ser decompostas em vários níveis de subcaracterísticas e, em atributos de qualidade de software, conforme as Figuras 1 e 2.

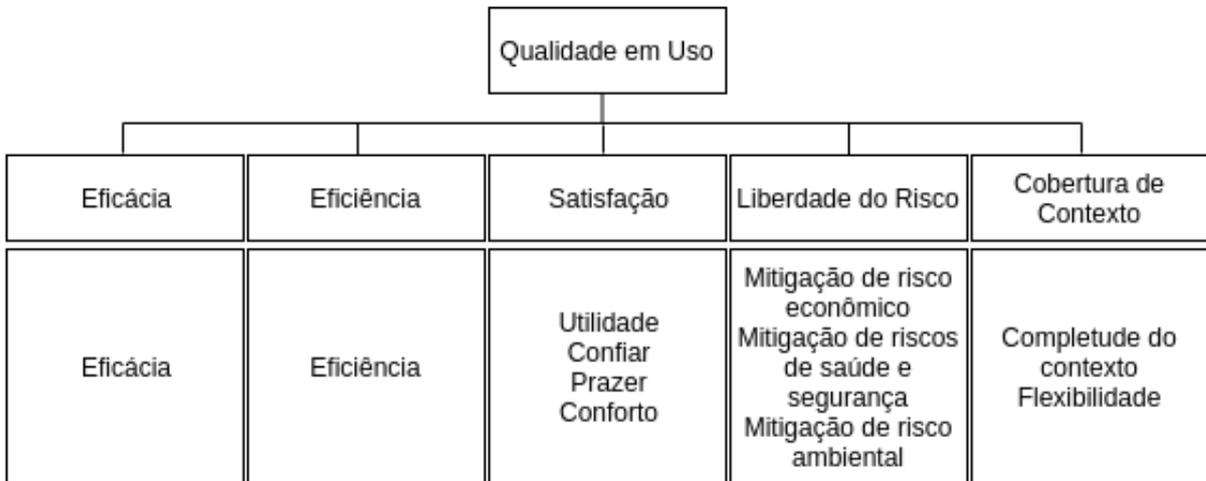
Wieggers e Beatty (2013) definem que atributo de qualidade é um tipo de requisito não-funcional que retrata um serviço ou uma característica de desempenho de um produto, e os distingue em atributos de qualidade externa e atributos de qualidade interna.

Alguns estudos consideram requisitos não-funcionais como sinônimo de atributos de qualidade e outros que, características de qualidade são um tipo de requisito não-funcional. Embora ainda não tenha uma definição formal e concisa, existe um esforço considerável na caracterização e classificação dos requisitos não-funcionais (CHUNG *et al.*, 2012).

Algumas diferentes definições podem ser encontradas na literatura. Segundo Thayer e Dorfman (1995), requisito não-funcional na engenharia de software, é um requisito do software que não descreve o que o software fará, mas como ele fará, por exemplo, requisitos de desempenho, de interface externa, restrições de *design* e atributos de qualidade. Para Kotonya e Sommerville (1998), no desenvolvimento de software, os requisitos não-funcionais são considerados como restrições impostas sobre o produto que está sendo desenvolvido. Para este trabalho será considerado que característica de qualidade, atributo de qualidade e requisito não-funcional, são sinônimos, e que eles podem se decompor em subníveis, ou seja, subcaracterísticas.

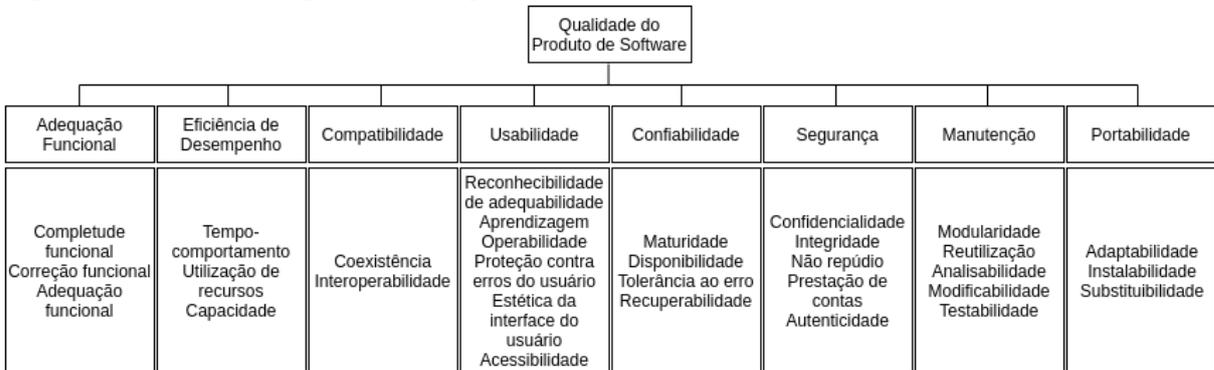
Assim como existem diferentes definições, também existem diferentes classificações para as características de qualidades. A Iso/Iec (2011) classifica as características em dois

Figura 1 – Modelo de qualidade em uso



Fonte: (ISO/IEC, 2011) com adaptações.

Figura 2 – Modelo de qualidade do produto de software



Fonte: (ISO/IEC, 2011) com adaptações.

modelos de qualidade. O primeiro é o modelo de qualidade em uso, o qual possui cinco características relacionadas aos resultados da interação com o sistema (ver Figura 1). O segundo é o modelo de qualidade interno e externo de produto, o qual categoriza propriedades de qualidade de produto de software em oito características (ver Figura 2).

Por outro lado, Wieggers e Beatty (2013) classificam as características em qualidade externa, quando descrevem características que são observadas enquanto o software está em execução e são mais significantes para o usuário (ver Tabela 1), e qualidade interna quando não é diretamente observado durante a execução e são mais significantes para os desenvolvedores e responsáveis pela manutenção (ver Tabela 2).

3.2 Interação entre Características de Qualidade

Certas combinações de características geram interações inevitáveis (WIEGERS; BEATTY, 2013), essas interações podem representar sinergia ou conflito, respectivamente, se

Tabela 1 – Características de qualidade externa

Característica	Descrição
Disponibilidade	É a capacidade dos serviços estarem disponíveis para o uso e totalmente operacional.
Instalabilidade	Quão fácil é instalar, desinstalar e reinstalar corretamente o aplicativo.
Integridade	Trata de prevenir a perda de informação e preservar a correção dos dados inseridos no sistema.
Interoperabilidade	Indica quão fácil o sistema pode trocar dados e serviços com outros sistemas de software e a facilidade com que ele pode se integrar com dispositivos de hardware externos.
Desempenho	Representa a capacidade de resposta do sistema a várias perguntas e ações dos usuários.
Confiabilidade	A probabilidade de o software executar sem falhar por um período de tempo específico.
Robustez	É o grau em que um sistema continua a funcionar corretamente quando confrontado com entradas inválidas, defeitos em componentes ou condições de operação inesperadas.
<i>Safety</i>	Trata da necessidade de evitar que um sistema cause ferimentos a pessoas ou danos à propriedade.
<i>Security</i>	Trata de bloquear o acesso não autorizado a funções ou dados do sistema, garantindo que o software esteja protegido contra ataques de <i>malware</i> , e assim por diante.
Usabilidade	Mede o esforço necessário para utilizar um sistema.

Fonte: Wieggers e Beatty (2013) com adaptações.

Tabela 2 – Características de qualidade interna

Característica	Descrição
Eficiência	É uma medida de quão bem o sistema utiliza a capacidade do processador, espaço em disco, memória ou largura de banda de comunicação.
Modificabilidade	A facilidade com que os projetos e o código do software podem ser entendidos, alterados e estendidos.
Portabilidade	O esforço necessário para migrar o software de um ambiente operacional para outro.
Reutilização	Indica o esforço relativo necessário para converter um componente de software para uso em outras aplicações.
Escalabilidade	A capacidade do aplicativo de crescer para acomodar mais usuários, dados, servidores, entre outros, sem comprometer o desempenho ou a correção.
Verificabilidade	Refere-se à como os componentes de software ou o produto integrado podem ser avaliados para demonstrar se o sistema funciona como esperado.

Fonte: Wieggers e Beatty (2013) com adaptações.

dois atributos se apoiam, ou seja, interferem positivamente, ou se dois atributos se contradizem, ou seja interferem negativamente.

Por exemplo, Wieggers e Beatty (2013) explicam que decisões de *design*, que aumentam a portabilidade de um componente de software, também tornam o software mais fácil de se conectar a outros componentes de software, mais fáceis de reutilizar e de testar.

Outro exemplo de Wieggers e Beatty (2013) é que aumentar a segurança no sistema,

provavelmente prejudicará o desempenho porque o sistema deverá passar por mais camadas de autenticação de usuários, criptografia e verificação de *malware*.

3.3 Catálogos para apoiar Características de Qualidade

Para identificarem as interações entre as características, os desenvolvedores devem fazer uso de catálogos (WIEGERS; BEATTY, 2013). Wieggers e Beatty (2013) apresentam uma matriz representando esses relacionamentos (Ver Figura 3).

Nessa matriz, um sinal de adição (+) em uma célula indica que a inclusão do atributo na linha correspondente geralmente tem um efeito positivo no atributo na coluna. Um sinal de menos (-) em uma célula significa que incluir o atributo nessa linha geralmente afeta negativamente o atributo na coluna. Uma célula vazia indica que o atributo na linha tem pouco efeito sobre o atributo na coluna. Entretanto, o efeito positivo ou negativo não é necessariamente recíproco, já que um atributo *A* qualquer pode possuir um efeito sobre um atributo *B* qualquer, e *B* simplesmente não possuir efeito algum sobre *A*.

Figura 3 – Matriz com interações entre características de qualidade

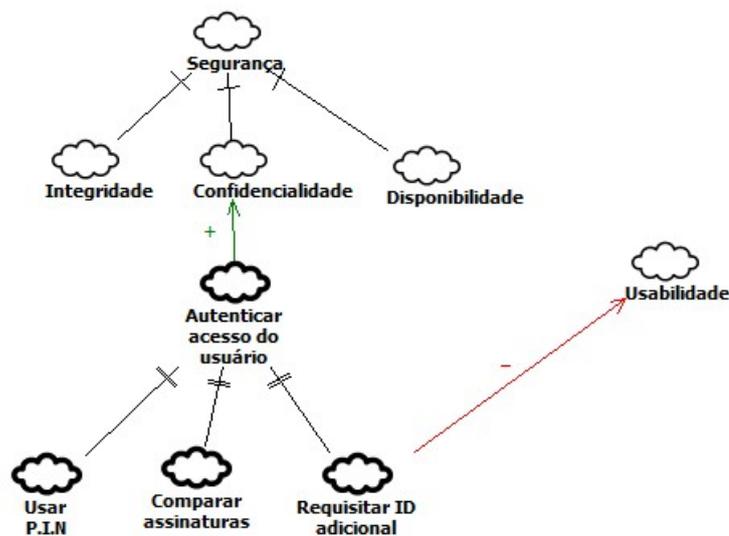
	Disponibilidade	Eficiência	Instalabilidade	Integridade	Interoperabilidade	Modificabilidade	Desempenho	Portabilidade	Confiabilidade	Reusabilidade	Robustez	Proteção	Escalabilidade	Segurança	Usabilidade	Verificabilidade	
Disponibilidade									+	+							
Eficiência	+				-	-	+	-			-		+		-		
Instalabilidade	+								+					+			
Integridade			-		-		-			-		+		+	-	-	
Interoperabilidade	+		-	-			-	+	+		+	-		-			
Modificabilidade	+		-				-		+	+			+			+	
Desempenho		+			-	-					-		-		-		
Portabilidade		-			+	-	-			+					-	-	+
Confiabilidade	+	-		+		+	-				+	+		+	+	+	
Reusabilidade		-	-	+	+		-	+						-		+	
Robustez	+	-	+	+	+		-		+			+	+	+	+		
Proteção		-		+	+		-				+			+	-	-	
Escalabilidade	+	+		+			+	+	+		+						
Segurança	+			+	+		-	-	+		+	+			-	-	
Usabilidade		-	+				-	-	+		+	+				-	
Verificabilidade	+		+	+		+			+	+	+	+		+	+		

Fonte: Rodrigues (2016).

Chung *et al.* (2012) utilizam o gráfico de interdependência de *softgoal* (do inglês,

Softgoal Interdependency Graphs - SIGs). Um SIG começa com uma característica de qualidade de alto nível e é desenhado como uma nuvem. Então essa característica é decomposta em sub-características mais específicas. Esses refinamentos devem ocorrer até chegarem em soluções que possam ser implementadas, que são chamadas de *softgoals* operacionalizadores e são desenhados como nuvens escuras. Um exemplo de SIG é apresentado na Figura 4, no qual Segurança foi refinada em três subcaracterísticas ou *softgoals* (Integridade, Confidencialidade e Disponibilidade). Ainda nesse exemplo, Confidencialidade foi refinada em um *softgoal* operacionalizador “Autenticar acesso do usuário”, este por sua vez foi mais uma vez refinado em três opções: 1) usar P.I.N; 2) Comparar Assinaturas; e 3) Requisitar ID adicional. Como é possível visualizar no gráfico, a última opção de *softgoal*, Requisitar ID adicional, tem um conflito com a característica Usabilidade, o qual é representado pelo sinal negativo.

Figura 4 – SIG de segurança



Fonte: Chung *et al.* (2012) com adaptações.

4 TRABALHOS RELACIONADOS

Até o momento, nenhum mapeamento sobre catálogos de características de qualidade foi encontrado. Portanto, nesta seção serão comentados dois trabalhos relacionados à mapeamento sistemático. O primeiro estudo é uma revisão que envolve características de qualidade, e o segundo estudo envolve heurísticas de usabilidade.

4.1 *Managing quality requirements: A systematic review*

Svensson *et al.* (2010), tem como objetivo neste estudo coletar e comparar evidências empíricas existentes sobre requisitos de qualidade (RQ), com o objetivo de fornecer aos pesquisadores direção para futuras pesquisas e profissionais com orientação na adoção da tecnologia RQ. Com foco em investigar artigos que se concentram em cinco áreas importantes para o gerenciamento de requisitos de qualidade, são eles: elicitación, métricas, dependências, estimativa de custos e priorização.

Para este estudo, Svensson *et al.* (2010) definiu as seguintes questões de pesquisa:

- RQ1: O que é atualmente conhecido de investigações empíricas sobre requisitos de qualidade em relação à elicitación, priorização, estimativa de custos, dependências e métricas?
- RQ2: Quais métodos de pesquisa empírica foram usados para avaliar os requisitos de qualidade?

Em uma tabela o estudo apresenta os termos utilizados para formar a string de busca separados em sete categorias, a seguinte cadeia de pesquisa foi usada: C1 AND C2 AND (C3 OR C4 OR C5 OR C6 OR C7). A estratégia de busca incluiu banco de dados eletrônicos e buscas manuais. As bases de dados são listadas a seguir:

- Bases de dados eletrônica
 - ACM Digital Library (portal.acm.org)
 - Compendex and Inspec (www.engineeringvillage2.com)
 - IEEE Xplore (ieeexplore.ieee.org)
 - Wiley Inter Science Journal Finder (www3.interscience.wiley.com)
 - ScienceDirect – Elsevier
 - SpringerLink
- Busca manual

- International Workshop on Software Product Management (IWSPM)
- Measuring Requirements for Project and Product Success (MeReP)
- Requirements Engineering: Foundation for Software Quality (REFSQ)

A busca retornou um total de 2.647 artigos, e a pesquisa teve quatro fases de filtro. A primeira fase filtrou somente trabalhos em inglês e deixou 1.560 estudos. Na segunda fase o primeiro autor leu os títulos dos trabalhos resultantes da primeira fase para determinar sua relevância para a revisão, ficando 727 estudos relevantes. Na terceira fase, o primeiro autor revisou todos os abstracts, e os outros dois autores revisaram 20% dos artigos excluídos, esse filtro deixou 229 estudos. A última fase deixou 18 estudos filtrando através de três critérios, onde o estudo deveria responder “sim” para o primeiro critério e “sim” para o segundo ou terceiro critério. Eles são descritos a seguir:

- SC1: O estudo é baseado em pesquisa empírica?
- SC2: As questões de pesquisa, objetivos do estudo e objetivos são bem definidos?
- SC3: O contexto estudado é bem definido?

Os 18 artigos selecionados foram avaliados de acordo com sete critérios de qualidade. O estudo apresenta uma tabela (tabela II) com a lista dos 18 estudos suas respectivas áreas e notas para cada critério utilizado, onde um “1” representa um “sim” e um “0” representa “não”.

Do resultado da primeira questão de pesquisa (O que é atualmente conhecido de investigações empíricas sobre requisitos de qualidade em relação à elicitação, priorização, estimativa de custos, dependências e métricas?), com relação à elicitação, o estudo mostrou que não há uma visão clara de como obter RQ. Porém existem várias sugestões de como o RQ deve ser eliciado. Já com relação à dependências, concluiu que as ferramentas ajudam os desenvolvedores e gerentes a lidar com as dependências de maneira mais eficiente. No que diz respeito à métricas, foram identificadas quatro escalas para quantificar os RQs, e são elas: Absoluto, Min-Max, Limite superior, Limite inferior. Para estimativa de custo apenas um artigo estava relacionado, e ele não apresenta como as estimativas de custo dos requisitos de qualidade são realizadas na indústria. Mas também conclui-se que as estimativas de custos são mais incertas nos estágios iniciais de uma nova tecnologia em comparação com a tecnologia que está disponível e atingiu certa maturidade do mercado. Com relação à priorização, a análise mostrou que nenhuma técnica de priorização focada unicamente no RQ foi avaliada empiricamente.

O resultado da segunda questão de pesquisa (Quais métodos de pesquisa empírica foram usados para avaliar os requisitos de qualidade?) identificou quatro tipos de estudos:

estudo de caso, survey, experimento e misto (survey e experimento), onde 12 trabalhos estavam relacionados a estudo de caso, indicando que o estudo de caso é o método de pesquisa mais utilizado.

A conclusão deste trabalho é que poucos estudos são replicados, por isso a possibilidade de tirar conclusões com base em variações é limitada (SVENSSON *et al.*, 2010). Observou-se que há a necessidade de aumentar o número e a qualidade dos estudos sobre requisitos de qualidade (SVENSSON *et al.*, 2010).

4.2 *How to develop usability heuristics: A systematic literature review*

Quiñones e Rusu (2017) realizaram uma revisão sistemática da literatura, a fim de conhecer quais processos são utilizados para desenvolver e especificar as heurísticas de usabilidade, e quais metodologias foram propostas para desenvolver essas heurísticas. O objetivo desse estudo também inclui: (i) identificar as diferentes abordagens usadas para gerar novos conjuntos de heurísticas de usabilidade, e (ii) determinar quantos estudos aplicam uma metodologia adequada para estabelecer as heurísticas (QUIÑONES; RUSU, 2017).

A partir do objetivo geral foram definidas as seguintes questões de pesquisa:

- RQ1. Como são desenvolvidas as heurísticas de usabilidade?
- RQ2. Existem metodologias para estabelecer heurísticas de usabilidade?
- RQ3. Se existem metodologias para estabelecer heurísticas de usabilidade, estas metodologias propõem um processo formal e sistemático para especificar e validar as heurísticas de usabilidade?

A string de busca foi dividida em cinco termos, são eles:

1. (“usability heuristics” OR “usability heuristic”) AND (methodology OR design OR "formal process"OR process) AND (“heuristic evaluation”)
2. (“usability heuristic” OR “usability heuristics” OR “usability guidelines”) AND (methodology OR design OR “formal process” OR “heuristic evaluation”)
3. (“usability heuristics” OR “usability heuristic”) AND (methodology OR design OR “formal process” OR process)
4. (usability) AND (heuristics) AND (“usability heuristics”) AND (process OR developing OR methodology OR “formal methodology” OR “formal process” OR “informal process” OR “developing usability heuristics” OR design OR “usability design” OR create)

5. (heuristic OR heuristics OR pattern) AND (usability) AND (methodology OR “design process” OR “heuristic evaluation”)

Os termos de busca estabelecidos foram aplicados nas bibliotecas digitais: Science Direct, AC M Digital Library, IEEE Xplore Digital Library, Springer Link, Scopus e Google Scholar. Também foram especificados critérios para determinar os artigos que seriam incluídos ou excluídos do escopo da revisão sistemática. Os trabalhos foram incluídos quando atenderam os seguintes critérios: Artigos de pesquisa; Estudos que propõem heurísticas de usabilidade para domínios específicos; Estudos que propõem uma abordagem, processo ou metodologia para estabelecer heurísticas de usabilidade; Estudos publicados entre janeiro de 2006 e abril de 2016; Artigos escritos em inglês, espanhol ou português.

Os critérios para exclusão dos trabalhos foram: Artigos que propõem heurísticas para outros aspectos (e.g., estética, automação e hiper-heurísticas); Estudos que não explicam como as heurísticas de usabilidade foram desenvolvidas; Teses que não foram publicadas; Artigos cujo foco principal não é a definição de heurísticas de usabilidade, como relatórios de estudos de caso de usabilidade ou testes de usabilidade.

Baseado nos critérios de inclusão e exclusão estabelecidos, um total de 73 artigos foram selecionados, assim, foram eliminados 87 estudos. Considerando os estudos incluídos no escopo da revisão, 68 deles são relacionados ao desenvolvimento de heurísticas de usabilidade voltadas para domínios específicos de aplicações, e os 7 restantes tratam de processos ou metodologias para o desenvolvimento das heurísticas de usabilidade.

Os resultados obtidos através da condução da revisão sistemática respondem às questões de pesquisa estabelecidas. Esses resultados são descritos a seguir.

Abordagens para o desenvolvimento das heurísticas de usabilidade (RQ1 e RQ2)

Os estudos utilizam diversas abordagens para desenvolver os conjuntos de heurísticas de usabilidade. Essas abordagens são baseadas em: heurísticas existentes; metodologias; revisões de literatura; relatórios de problemas de usabilidade; processos mistos; recomendações, princípios ou guidelines de design; entrevistas; e por fim, teorias. Dentre as abordagens aplicadas, as mais recorrentes são: Heurísticas existentes e Metodologias.

As metodologias existentes propõem um processo formal e sistemático para especificar e validar as heurísticas de usabilidade? (RQ3)

As metodologias identificadas apresentam processos novos, interessantes e úteis para estabelecer heurísticas de usabilidade. Contudo, elas não possuem uma especificação formal

para as suas etapas e atividades. Desse modo, ainda não existe uma metodologia que forneça um processo formal e sistemático para especificar e validar heurísticas de usabilidade.

É possível concluir também que para desenvolver um novo conjunto de heurísticas de usabilidade, se torna necessário a execução de quatro atividades (QUIÑONES; RUSU, 2017):

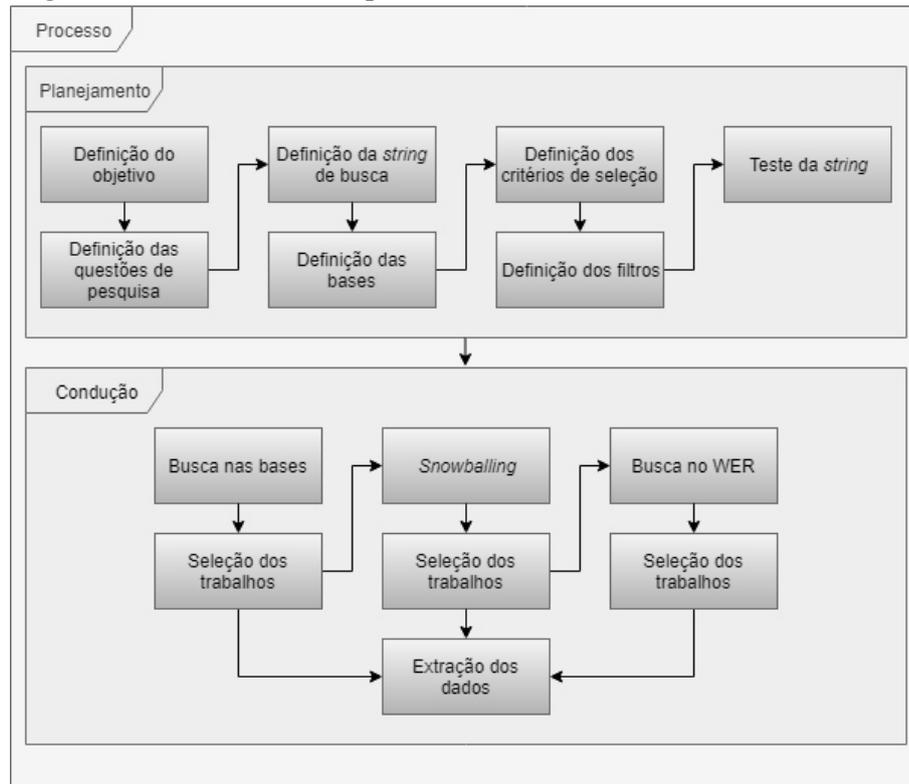
1. Determinar as características específicas do domínio de aplicações escolhido, a fim que essas características sejam avaliadas através do conjunto de heurísticas;
2. Identificar os conjuntos de heurísticas existentes que podem auxiliar na definição das novas heurísticas;
3. Especificar o conjunto de heurísticas considerando um template padrão para que as heurísticas propostas sejam bem definidas e fáceis de compreender;
4. Validar as heurísticas propostas com o intuito de avaliar se elas são capazes de identificar problemas de usabilidade e, principalmente, problemas específicos relacionados ao domínio de aplicação escolhido.

Em relação ao uso de uma metodologia adequada, conclui-se que a maioria dos estudos analisados não aplicaram um processo formal para especificar e validar as heurísticas de usabilidade (QUIÑONES; RUSU, 2017). Novamente, destaca-se que o uso desse tipo de processo é importante para criar conjuntos de heurísticas de usabilidade eficazes e bem projetados (QUIÑONES; RUSU, 2017).

5 METODOLOGIA

A Figura 5 apresenta o processo do mapeamento sistemático que serviu de guia para a execução deste trabalho, o qual está dividido em duas fases: planejamento e condução. Cada atividade dentro de uma fase é representada na figura e explicada no decorrer desta Seção.

Figura 5 – Processo do mapeamento sistemático



Fonte: Carvalho *et al.* (2017) com adaptações.

5.1 Planejamento

Nesta fase o objetivo do mapeamento é estabelecido, bem como suas questões de pesquisa, que ajudam no processo do mapeamento guiando a pesquisa até seu objetivo. No final do estudo espera-se que as questões de pesquisa sejam respondidas. A *string* para fazer a busca dos trabalhos é criada, e as bases de dados escolhidas. Após isto, os critérios de seleção dos estudos são definidos, eles são úteis para selecionar os estudos que são relevantes e ajudam a responder às questões de pesquisa. Os critérios são aplicados em filtros que também são definidos no planejamento. A última atividade desta etapa é realizar o teste da *string*, para validar se ela é suficiente. O protocolo do mapeamento é definido a seguir.

5.1.1 Definição do objetivo

O objetivo deste mapeamento é obter uma visão geral de como está a literatura acerca de catálogos de características de qualidade de software. Em geral, saber quais catálogos existem na literatura, quais requisitos são mais abordados e quais menos, a qual domínio pertencem, ou seja, se os catálogos são definidos sob o olhar de um domínio específico ou não, e quais os domínios que predominam a pesquisa sobre catálogos.

5.1.2 Definição das questões de pesquisa

Para ajudar a chegar no objetivo deste mapeamento, a seguir são apresentadas as questões de pesquisa definidas para este trabalho:

- RQ 1: Quantos artigos foram selecionados por base de dados?
 - RQ 1.1: Qual a quantidade de artigos publicados por ano?
 - RQ 1.2: Quais tipos de pesquisa os artigos abordam?
 - RQ 1.3: Quais os tipos de publicações encontrados?
 - RQ 1.4: Quais pesquisadores possuem mais artigos publicados?
- RQ 2: Quais catálogos existem na literatura?
 - RQ 2.1: Quais os tipos de catálogos encontrados?
 - RQ 2.2: Quais características e subcaracterísticas de qualidade são abordadas?
 - RQ 2.3: Quais domínios são abordados nos catálogos encontrados?
 - RQ 2.4: Qual a quantidade de catálogos publicados por ano?
 - RQ 2.5: Quais pesquisadores possuem mais catálogos publicados?
- RQ 3: Como os catálogos são representados? (tabela, gráfico, notação específica)
 - RQ 3.1: Como conflitos e sinergias são representados nos catálogos?

5.1.3 Definição da string de busca

Para a criação da *string* de busca foi utilizada a estratégia PICO (*Population, Intervention, Comparison, Outcomes* - População, Intervenção, Comparação, Resultados) de acordo com Petersen et al. (2015).

- População: a população pode se referir a papéis específicos da engenharia de software, uma área de aplicação ou um grupo industrial. No contexto deste trabalho a população são as características de qualidade.

- Intervenção: refere-se a uma metodologia, ferramenta, tecnologia, procedimento de software ou algo a ser investigado com relação à população. Neste trabalho a intervenção são as interações.
- Comparação: refere-se à qualquer comparação entre os estudos primários. Como este estudo é focado na caracterização de uma área, nenhuma comparação foi definida.
- Resultados: são os resultados esperados. Para este estudo os resultados são catálogos de interação entre as características de qualidade.

Porém para este estudo utilizamos apenas o PI. A decisão tem como base Petersen *et al.* (2015) que explica que para estudo de mapeamento sistemático o uso dos outros pilares restringem muito a pesquisa e pode acabar excluindo artigos da área desejada.

Utilizando a estratégia PICO e se inspirando no trabalho de Santos *et al.* (2017), a Tabela 3 foi criada para ajudar na criação da *string* de busca.

Tabela 3 – *String* de busca usando a estratégia PICO

População	quality characteristic, non-functional requirement, nfr, quality attribute, non-functional property, extra-functional requirement, non-behavioural requirement, quality requirement, quality factor
Intervenção	catalog, catalogue, sig, softgoal interdependency graph

Fonte: Elaborado pela autora.

Dessa forma, a *string* resultante é formada pela concatenação das palavras-chave, utilizando os operadores booleanos *AND* e *OR*, apresentada abaixo:

("quality characteristic"OR "non-functional requirement"OR nfr OR "quality attribute"OR
 "non-functional property"OR "extra-functional requirement"OR "non-behavioural
 requirement"OR "quality requirement"OR "quality factor")
 AND
 (catalog OR catalogue OR sig OR "softgoal interdependency graph")

5.1.4 Definição das bases

As bases de dados Web of Science e Scopus foram escolhidas pois, de acordo com Santos *et al.* (2017), elas apresentam uma boa cobertura e estabilidade, além de abrangerem outras bases, como a IEEE. Outra base de dados escolhida foi o *Workshop on Requirements Engineering* (WER), por três razões: ele representa um evento importante na área onde os

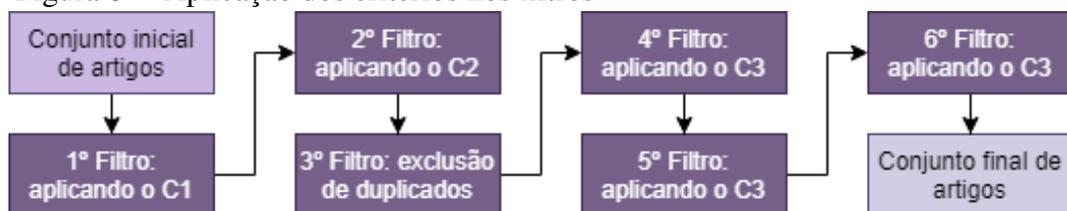
pesquisadores geralmente publicam catálogos de NFRs, as publicações dessa base não são todas indexadas nos bancos de dados que escolhemos anteriormente, e fornece um mecanismo de busca próprio.

5.1.5 Definição dos critérios de seleção

Os critérios de seleção de artigos para este estudo são três. O primeiro critério (C1) é que seja um trabalho escrito na língua inglesa. O segundo critério (C2) é que os artigos sejam da área de computação e engenharia, como o foco é características de qualidade, não restringir uma área de pesquisa resultaria em artigos com contexto e objetivos completamente diferentes do esperado. O último critério (C3) é que o artigo seja relevante para o mapeamento, entende-se por relevante um estudo que esteja diretamente relacionado ao tema abordado.

5.1.6 Definição dos filtros

Figura 6 – Aplicação dos critérios nos filtros



Fonte: Elaborado pela autora.

Como mostra a Figura 6, os dois primeiros filtros são realizados nas próprias bases de dados, o primeiro leva em consideração o critério de ser um estudo escrito na língua inglesa e o segundo filtro leva em consideração o segundo critério que é pertencer à área da computação ou engenharia. O terceiro filtro é a remoção de artigos duplicados, este filtro e os próximos são realizados com o suporte da ferramenta Parsifal. O quarto filtro é a leitura de títulos e *abstracts* buscando por estudos relevantes para o mapeamento. O quinto filtro é feito por meio da leitura da introdução e conclusão, e se o artigo apresenta algum catálogo de características. No último filtro é realizada a leitura completa do artigo.

5.1.7 Teste da string

A última atividade desta fase é realizar o teste da *string*, para validar se ela é suficiente. O teste é feito nas bases de dados Web of Science e Scopus, onde a *string* deve

retornar dentre todos os artigos alcançados, os artigos de controle:

- *Management of performance requirements for information systems*
- *Constructing a Catalogue of Conflicts among Non-functional Requirements*
- *Evaluating the Effectiveness of Using Catalogues to Elicit Non-Functional Requirements*

5.2 Condução

A busca aconteceu em três momentos diferentes, primeiramente a busca nas bases de dados eletrônica é realizada e é explicada na Seção 5.2.1. O resultado dessa busca é usado como entrada para a busca manual em forma de *Snowballing*, que é apresentada na Seção 5.2.2. A última busca, também manual, foi na base do *Workshop on Requirements Engineering (WER)*, mais detalhes na Seção 5.2.3.

5.2.1 Busca nas bases de dados eletrônicas

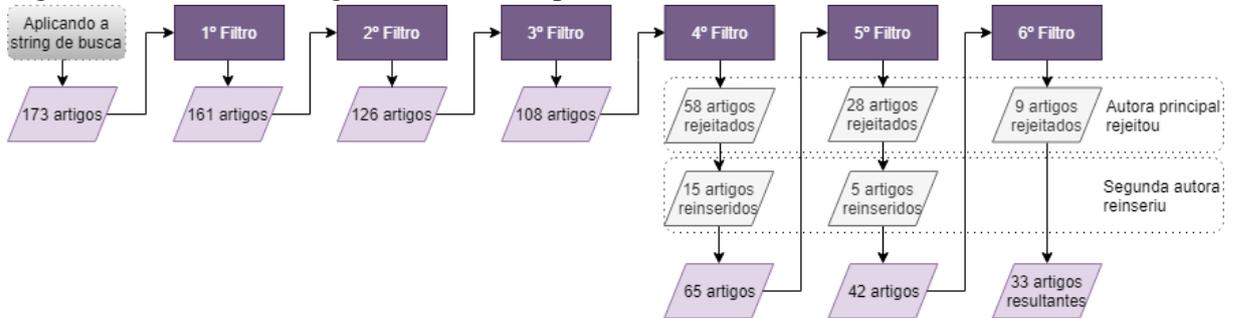
A Figura 7 ilustra os filtros que foram realizados e a quantidade de artigos. Quando a *string* foi lançada nas bases de dados, foram encontrados 143 artigos na scopus, dentre eles dois artigos de controle (*Constructing a Catalogue of Conflicts among Non-functional Requirements* e *Evaluating the Effectiveness of Using Catalogues to Elicit Non-Functional Requirements*). E na web of science, 30 artigos, dentre eles um artigo de controle (*Management of performance requirements for information systems*). Nas próprias bases foram realizados dois filtros, o primeiro aplicado deixou 132 artigos da scopus e 29 artigos da web of science para o filtro seguinte. O segundo resulta em 105 e 21 artigos da scopus e web of science, respectivamente, para o próximo passo.

A ferramenta parsifal, utilizada para ajudar o processo de seleção dos artigos permite que o bibtex dos artigos seja adicionado, então boa parte das informações dos artigos estava disponível, como título, autores, ano e lugar de publicação, além do abstract. Com o apoio desta ferramenta, os filtros seguintes foram fáceis de realizar, ela mesma, por exemplo, já identifica estudos duplicados, e adiciona um rótulo. Neste filtro foram encontrados 18 artigos duplicados na scopus.

Os próximos filtros são feitos por meio da leitura de partes específicas dos artigos, buscando indícios de quais artigos são relevantes para este estudo. Os dois filtros seguintes são

feitos da mesma forma, cada filtro é dividido em duas partes, na primeira, uma autora faz a leitura dos artigos e rotula eles em aceito ou rejeitado. Uma segunda autora, realiza a leitura dos artigos que foram rejeitados pela primeira autora mudando o rótulo dos que acha ser relevante.

Figura 7 – Filtro dos artigos das bases scopus e web of science



Fonte: Elaborado pela autora.

No quarto filtro, que é a leitura dos títulos e *abstracts*, a primeira autora rotulou como aceito 36 artigos da scopus e 14 da web of science. E com rejeitado 51 artigos da scopus e 7 da web of science. Dos artigos rejeitados, 15 voltaram para o *status* aceito, sendo 11 da scopus e 4 da web of science, após a análise da segunda autora.

No quinto filtro por meio da leitura da introdução e conclusão e da busca por catálogos nos artigos, a primeira autora aceitou 28 artigos da scopus e 9 da web of science. E rejeitou 19 artigos da scopus e 9 da web of science. Três estudos foram excluídos sem uma segunda avaliação, pois um era tutorial, o segundo era um artigo em português e o terceiro era duplicado. A segunda autora aceitou 5 artigos da web of science.

O último filtro tem 6 artigos da scopus e 3 da web of science rejeitados. Para a extração dos dados ficam 22 artigos da scopus e 11 da web of science.

5.2.2 *Snowballing*

O conjunto de artigos restantes da busca anterior foi definido como o início para o *Snowballing* para trás, definido por Wohlin *et al.* (2013) para complementar o conjunto de artigos encontrados. O procedimento inverso consiste em verificar a lista de referências de um conjunto de documentos.

O procedimento de Wohlin *et al.* (2013) que consistia em quatro filtros, foi adaptado. Esses filtros são diferentes dos usados anteriormente, já que a seleção por meio de *Snowballing* para trás é um pouco diferente. Primeiro, começou com a revisão manual de todas as referências de cada artigo aceito na busca anterior, então, para cada referência, foi aplicado os seguintes

filtros:

1. Aplicando todos os critérios de seleção na referência. Nesse caso, sendo mais específico, pois era apenas leitura de títulos, autores, local e ano. Assim, foi aceito artigos que apresentavam qualquer palavra-chave da *string* de pesquisa.
2. Aplicando todos os critérios de seleção na leitura dos *abstracts*.
3. Aplicando os critérios de seleção nas partes mais relevantes dos artigos (introdução, conclusão e imagens de catálogos).
4. Aplicando os critérios de seleção na leitura completa do artigo.

Figura 8 – Filtro dos artigos do *snowballing*



Fonte: Elaborado pela autora.

Como ilustra a Figura 8, a busca iniciou com 847 referências e finalizou com 15 trabalhos selecionados para a extração de dados.

5.2.3 Busca manual no WER

O Workshop de Engenharia de Requisitos (WER) existe desde 1998 e tem sido um avanço para a comunidade ibero-americana de pesquisadores. Este workshop fornece um mecanismo de busca do Google que explora documentos em todas as suas edições. Portanto, para realizar a busca em cada edição foi utilizada a mesma *string* e filtros usados na busca nas bases de dados.

Figura 9 – Filtro dos artigos do WER



Fonte: Elaborado pela autora.

Como a Figura 9 apresenta, a busca iniciou com 28 trabalhos, os filtros foram aplicados usando os critérios já citados, ao final 5 trabalhos foram para extração. A leitura completa dos artigos que saíram do quinto filtro, não excluiu nenhum trabalho.

Para um melhor entendimento do processo da condução e seus resultados, a Figura 10 demonstra os filtros que aconteceram durante a condução em todas as bases de dados.

Figura 10 – Filtro geral



Fonte: Elaborado pela autora.

6 RESULTADOS E DISCUSSÕES

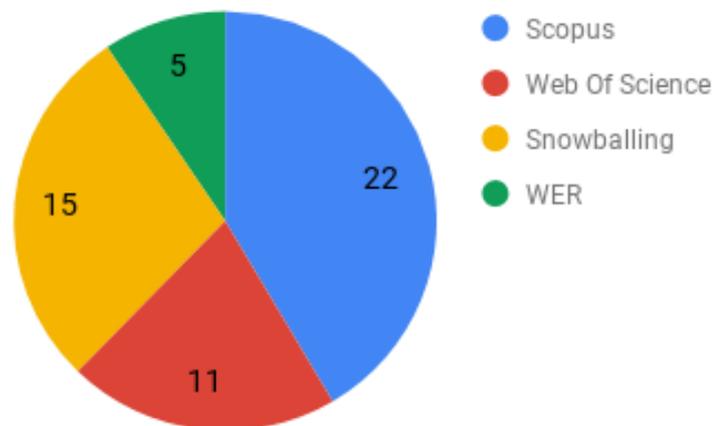
Um formulário de extração foi elaborado e está organizado em cinco partes. A primeira é sobre os artigos que foram aceitos durante a fase de condução: título, autores, ano, fonte (Scopus, Web of Science, *Snowballing* ou WER), local, tipo de publicação (ex. Conferência, revista), principal contribuição do artigo e tipo de pesquisa, seguindo a classificação de Wieringa *et al.* (2006).

As outras partes estão relacionadas aos dados para responder às questões de pesquisa sobre os catálogos. Foram definidos os seguintes dados: ID do catálogo, tipo do catálogo (de acordo com as informações que o catálogo apresenta), RNFs considerados no catálogo, subcaracterísticas, métodos, quantidade de correlações negativas e positivas, tipo de correlação (INTRA-RNF ou INTER-RNF), nível de correlação e área de domínio pela qual o catálogo foi desenvolvido. Também foi definido um dado chamado “Autoria Original” porque muitos estudos não propuseram por si só um catálogo, mas usaram um catálogo de outro estudo. E assim, obter as informações sobre a origem do catálogo. A forma de representação do catálogo e a forma de representação para correlações.

Todas as informações extraídas foram gerenciadas em um editor de planilhas, os links para os dados extraídos estão no Apêndice D. A seguir são apresentados os resultados e as análises dos dados.

6.1 RQ 1: Quantos artigos foram selecionados por base de dados?

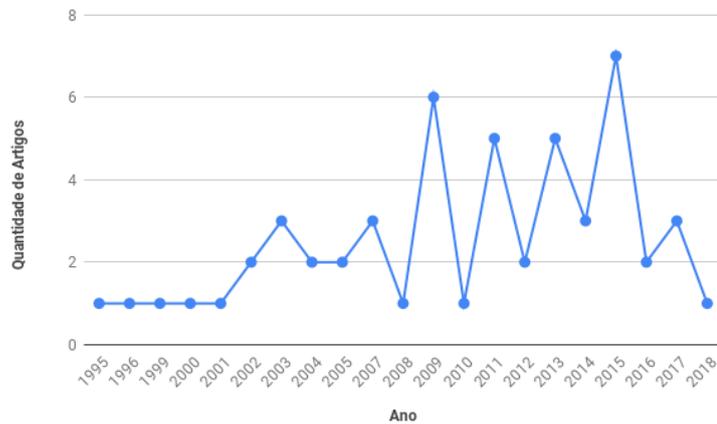
Figura 11 – Quantidade de artigos por base



Fonte: Elaborado pela autora.

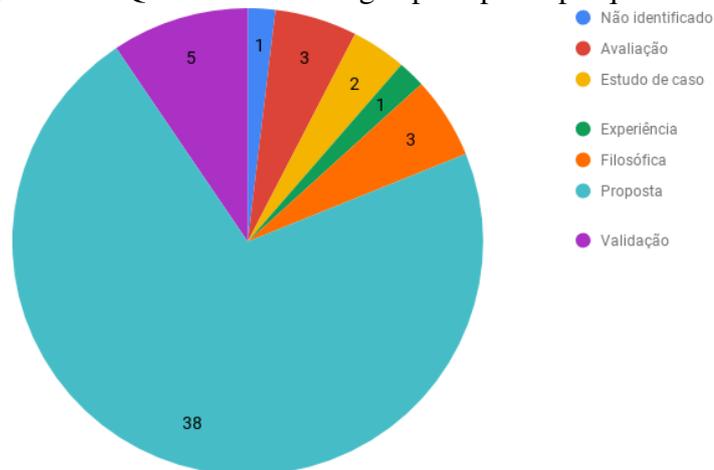
Após a fase de seleção de trabalhos, 53 artigos foram escolhidos (Figura 11), onde a maior quantidade de artigos foi da base de dados Scopus. No Apêndice A está a lista dos artigos, com a identificação do artigo, sua referência e os catálogos que foram encontrados a partir dele. Dos selecionados, algumas informações básicas foram extraídas, como o ano de publicação, que é apresentado na Figura 12, permitindo observar o progresso da área de pesquisa durante os anos. O artigo mais antigo selecionado foi de 1995, mesmo assim poucos artigos foram publicados por ano, já que os dois maiores picos aconteceram em 2009 e 2015, com seis e sete artigos publicados, respectivamente. E nos anos seguintes, 2010 e 2016, há uma queda de mais da metade da quantidade de artigos publicados.

Figura 12 – Quantidade de artigos por ano



Fonte: Elaborado pela autora.

Figura 13 – Quantidade de artigos por tipo de pesquisa



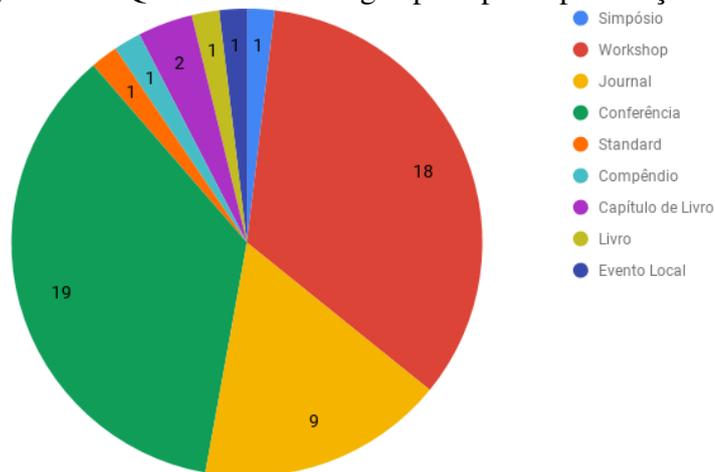
Fonte: Elaborado pela autora.

Outra informação coletada foi o tipo de pesquisa dos artigos, o tipo mais utilizado nos trabalhos, como mostra a Figura 13, foi o de proposta. O menos frequente foi o de experimento,

houve ainda um artigo que o tipo de pesquisa não foi identificado.

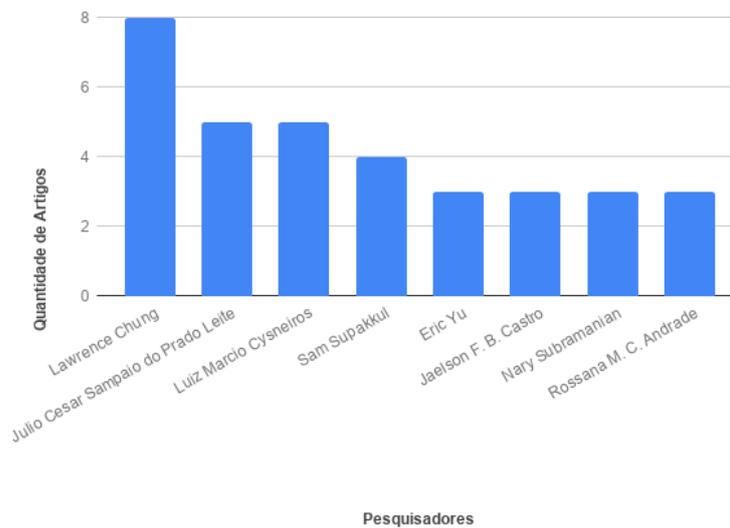
Na Figura 14, conferência e workshop são os tipos de publicação mais frequentes. E a Figura 15 mostra os pesquisadores com mais artigos publicados. Lawrence Chung possui a maior quantidade de artigos publicados.

Figura 14 – Quantidade de artigos por tipo de publicação



Fonte: Elaborado pela autora.

Figura 15 – Pesquisadores com mais artigos publicados



Fonte: Elaborado pela autora.

6.2 RQ 2: Quais catálogos existem na literatura?

No total foram extraídos 102 catálogos de características diferentes, a lista desses catálogos está no Apêndice B contendo informações, como o identificador do catálogo e os

autores, caso forem dos artigos já selecionados estará o identificador do artigo, caso não, estará a referência do artigo. As imagens dos catálogos estão numa pasta, no Apêndice D está o *link* para acesso.

As informações encontradas nos catálogos foram: subcaracterísticas, métodos e correlações. Nem todos os catálogos possuem todas as informações citadas, por isso neste trabalho, 7 tipos de catálogos foram definidos e são expostos na Tabela 4.

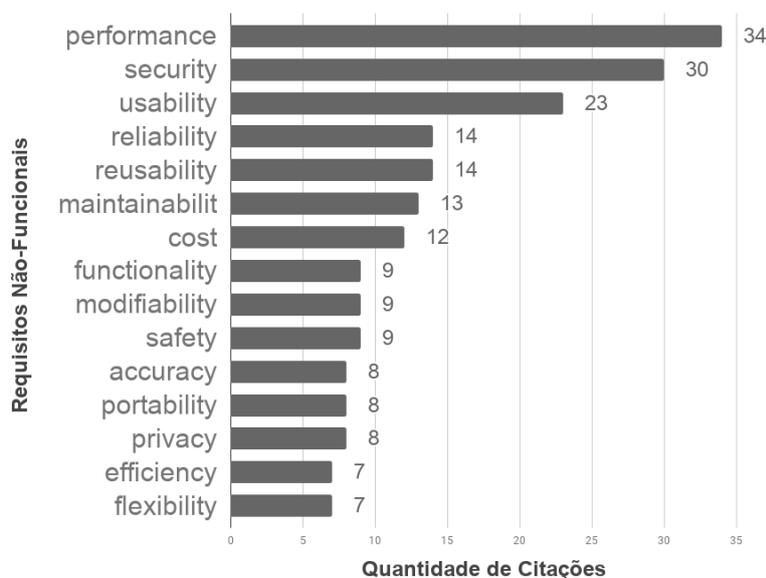
Tabela 4 – Tipos dos catálogos

Id	Tipo de Informação	Quantidade de Catálogos
T1	Subcaracterísticas	28
T2	Subcaracterísticas e métodos	22
T3	Subcaracterísticas, métodos e correlações	30
T4	Métodos	7
T5	Métodos e correlações	3
T6	Correlações	5
T7	Subcaracterísticas e correlações	7

Fonte: Elaborado pela autora.

A Tabela 4 mostra que na sua maioria os catálogos apresentam três tipos de informação: Subcaracterísticas, Métodos e Correlações. Seguido por catálogos que mostram apenas subcaracterísticas, e catálogos que apresentam subcaracterísticas e métodos.

Figura 16 – Requisitos mais citados



Fonte: Elaborado pela autora.

Também houve extração dos RNFs suportados por esses catálogos. No total, 348 RNFs foram extraídos. No entanto, esse número inclui RNFs duplicados, o que significa que

eles tinham o mesmo nome. Em seguida, foi analisado RNFs duplicados e sinônimos de RNFs, após essa análise, observa-se que esses catálogos suportam 86 RNFs diferentes. O Apêndice C apresenta todos os RNFs e a Figura 16 os requisitos não-funcionais mais citados. Desempenho foi o RNF mais citado (34), seguido de Segurança (29), Usabilidade (23) e Confiabilidade (14). Em relação às subcaracterísticas e métodos, foram encontrados 1269 e 1113, respectivamente.

Na Tabela 5 são descritos a área, domínio e parte do sistema dos catálogos e suas respectivas quantidades. A área geral possui o maior índice de catálogos (29), e 13 catálogos estão sob o domínio da saúde.

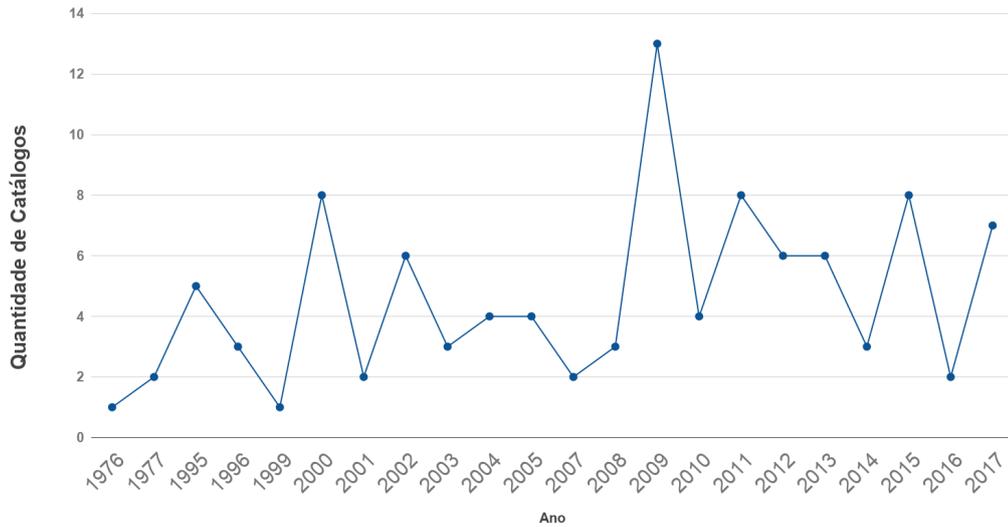
Tabela 5 – Domínio dos catálogos

Área	Qtd	Domínio	Qtd	Parte do Sistema	Qtd
General	29	Electoral Systems	1	Requirements Scenarios Ar- tifacts	1
Ubiquitous/Pervasive	8	Health	13	Middleware	8
Mobile Applications	10	Banking	1	Architecture	1
Context-Aware Adaptive	6	Accessibility	1	Source Code	2
Software Platform	5	Deicing services Manage- ment	3		
Data warehouse	5	Government Tax Credits	1		
Information System	4	KWIC (Keyword in Context) system	2		
Web	3	Museum integration	1		
Software Product Lines	2	integrate cultural heritage re- positories	1		
Micro Businesses	2	Process Improvement	1		
Embedded System	2	Learning	2		
Cloud Services	1				
Scientific Workflows	1				
Self-adaptive systems	1				
trustworthy systems	1				
image processing applicati- ons	1				
information visualization	1				
COTS-based System	1				
Legacy systems	1				
Multi-agent systems	1				

Fonte: Elaborado pela autora.

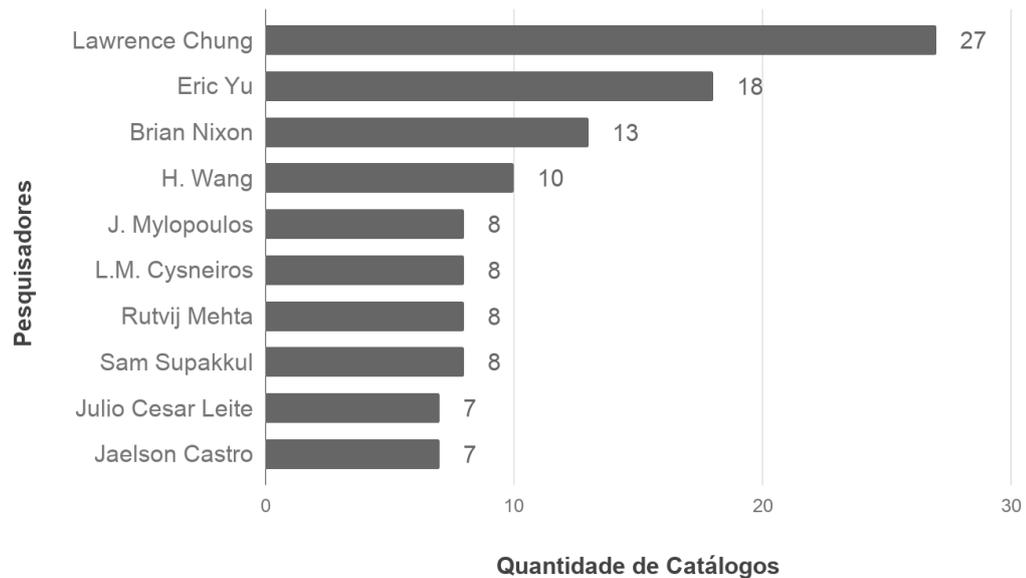
As Figuras 17 e 18 apresentam a quantidade de catálogos publicados por ano e os pesquisadores com mais catálogos publicados, respectivamente. Em 2009 houve o maior índice de catálogos publicados como mostra a Figura 17. E o pesquisador Lawrence Chung lidera novamente com 27 catálogos publicados (Figura 18) a lista de pesquisadores com mais catálogos publicados, assim como lidera na Figura 15.

Figura 17 – Quantidade de catálogos por ano



Fonte: Elaborado pela autora.

Figura 18 – Quantidade de catálogos por pesquisador



Fonte: Elaborado pela autora.

6.3 RQ 3: Como os catálogos são representados?

Como mostra a Tabela 6, os tipos de representação do conhecimento sobre RNFs encontrados foram: matriz, notação de i^* , tabela, estrutura hierárquica, SIG, adaptações de SIG, lista, *template* e *pattern*. As representações mais comuns dos catálogos são SIG, estrutura hierárquica e matriz. Para melhor descrever essas representações, foi analisado quais tipos de catálogos de RNFs são mais armazenados em cada um deles. O Quadro 1 apresenta o cruzamento de dados entre o tipo de catálogo e o tipo de representação. Portanto, pode-se ver que SIGs e

estruturas hierárquicas são os mais utilizados.

Tabela 6 – Tipos de representação dos catálogos

Tipo de Representação	Quantidade de Catálogos
Matriz	14
i*	7
Tabela	4
Estrutura hierárquica	22
SIG	50
SIG adaptado	2
Lista	1
Template	2
Pattern	1

Fonte: Elaborado pela autora.

Quadro 1 – Tipo de catálogo vs tipo de representação

	T1	T2	T3	T4	T5	T6	T7
Matriz			3		2	5	4
i*	4	1	2				
Tabela	1			3			
Estrutura hierárquica	16	1		5			
SIG	5	17	25				3
SIG adaptado		2					
Lista	1						
Template				1	1		
Pattern		1					

Fonte: Elaborado pela autora

Assim como citado anteriormente, existem 7 tipos de catálogos (Tabela 4), e apenas os catálogos do Tipo T3, T5, T6 e T7 apresentam correlações. De acordo com Mehta *et al.* (2012), as correlações podem ser INTER-RNFs, o que significa que há uma interação entre pelo menos dois RNFs de nível superior e INTRA-RNF, o que significa que há interação entre os componentes de um único RNF.

De acordo com Mehta *et al.* (2012) as correlações como INTER-RNFs podem ser mostradas através de quatro níveis:

- a) correlação entre os RNFs de nível superior, por exemplo, usabilidade prejudica a segurança;
- b) correlação entre subcaracterísticas, por exemplo, confidencialidade (sub-RNF de segurança) prejudica a Eficiência (sub-RNF de usabilidade);
- c) correlação entre o método e o RNF de nível superior, por exemplo, criptografia prejudica o desempenho; e

- d) correlação entre o método e a subcaracterística, por exemplo, criptografia prejudica o Tempo de Resposta (sub-RNF de desempenho).

As correlações como INTRA-RNF podem ser mostradas através de três níveis, de acordo com Mehta *et al.* (2012):

- a) correlação entre as subcaracterísticas do mesmo RNF;
- b) correlação entre um método de uma subcaracterística e outra subcaracterística do mesmo RNF; e
- c) correlação entre métodos.

Com isso observa-se que o Quadro 2 mostra a quantidade de catálogos por tipo e nível de correlação.

Quadro 2 – Tipo de correlação por nível de correlação

Nível	Inter-NFR	Intra-NFR
Entre características	6	
Entre subcaracterísticas	5	9
Entre métodos	4	2
Entre métodos e características	13	
Entre métodos e subcaracterísticas	9	12
Entre características e subcaracterísticas	3	

Fonte: Elaborado pela autora

As correlações encontradas podem ser de conflito ou sinergia, elas são representadas por símbolos, termos ou ambos.

- Símbolos
 - +, -, ++, --, -?, 0, X, *, - -*
- Termos
 - help, hurt
- Ambos
 - help+, hurt-, Make (++) , help (+), hurt (-), break (-)

6.4 Ameaças à validade

Seguindo a sugestão de Petersen *et al.* (2015), que também é um estudo sistemático, os seguintes tipos de validade devem ser considerados:

- a) Validade descritiva: significa a extensão em que as observações são descritas com precisão. Esse tipo de ameaça geralmente tem mais risco em estudos qualitativos. No caso deste trabalho, não houve questões de pesquisa para extrair

dados qualitativos, portanto, essa ameaça é considerada sob controle.

- b) Validade teórica: é determinada pela capacidade de capturar o que se pretende capturar. Portanto, a identificação do estudo, a seleção do estudo e a extração de dados são ameaças importantes a serem consideradas.

No caso deste estudo, inicialmente optou-se por usar dois bancos de dados (Scopus e Web of Science), mas para reduzir a ameaça em relação à identificação do estudo, foi realizado o *snowballing* para trás para os artigos selecionados da Scopus e da Web of Science. Além disso, foi adicionado um banco de dados do *Workshop on Requirements Engineering* (WER) porque ele representa um evento importante na área onde os pesquisadores da área de catálogos geralmente publicam seus estudos, porém nem todas as publicações são indexadas nos bancos de dados. No entanto, vale ressaltar que apenas um pequeno número (5) de novos estudos foi obtido a partir do WER, indicando que as conclusões gerais deste mapeamento não mudariam.

Quanto à seleção do estudo, artigos válidos poderiam ter sido rejeitados. Para evitar essa ameaça, durante a seleção de estudos da Web of Science e da Scopus, um especialista que trabalha com catálogos de NFRs revisou essa seleção toda vez que um artigo foi rejeitado. Portanto, um artigo que foi rejeitado pode ser considerado novamente após a revisão.

Além disso, sempre que um artigo excluído retornava ao conjunto de artigos selecionados, uma explicação era dada ao autor que estava fazendo a seleção para obter um entendimento comum dos estudos.

As sugestões da literatura dizem que também é apropriado ter um pesquisador extraindo dados e outro revisando a extração (PETERSEN *et al.*, 2015). Portanto, extrações foram realizadas por um autor e outro revisado todas as extrações em estudos da Web of Science e Scopus.

- c) Generalização: é determinada pelo grau de generalização dos resultados. Há uma distinção entre generalização externa (generalização entre grupos ou organizações) e generalização interna (generalização dentro de um grupo). Uma possível ameaça é que a interpretação dos dados pode ser diferente para diferentes pesquisadores. Para atenuar essa ameaça, a maior parte da extração foi realizada em pares. Assim, considera-se que a generalização interna não é uma grande

ameaça. Em relação à generalização externa, procurou-se atenuar essa ameaça realizando esse trabalho com pesquisadores de três organizações e dois países, aumentando as possibilidades de generalização.

- d) Validade Interpretativa: este tipo de validade preocupa-se se as conclusões foram baseadas nos dados, sejam objetivos ou subjetivos. Neste trabalho, a maioria das extrações foi revisada por um especialista. Além disso, a análise dos dados não foi conduzida informalmente, e sim seguindo uma metodologia. A análise quantitativa seguiu as sugestões de Wholin para apresentar os dados em gráficos.
- e) Repetibilidade: requer relatórios detalhados do processo de pesquisa. Para atingir este tipo de validade, seguiu-se as diretrizes da literatura para realizar várias etapas desta pesquisa. Para realizar o estudo sistemático de mapeamento em geral, seguiu-se as instruções propostas por Petersen *et al.* (2015), e também considerou-se outros processos sistemáticos de mapeamento, como os de Santos *et al.* (2017), Montagud *et al.* (2012), Neto *et al.* (2011) e Carvalho *et al.* (2017). Além disso, seguiu-se as diretrizes propostas por Jalali e Wohlin (2012) e Badampudi *et al.* (2015) para a realização do *snowballing*.

7 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou um mapeamento sistemático dos catálogos dos RNFs. Através do método de mapeamento sistemático, juntamente com a análise quantitativa, foi possível obter muitas informações sobre os catálogos úteis não apenas para pesquisadores, mas também para desenvolvedores preocupados com a satisfação de RNFs em seus sistemas. Assim, as contribuições deste trabalho têm implicações tanto para a indústria quanto para a academia.

Para a indústria, este trabalho gerou um banco de dados contendo mais de 1000 subcaracterísticas, 1113 métodos de implementação, 473 correlações positivas, 395 correlações negativas para 86 RNFs. Acredita-se que esta base pode ser útil para os desenvolvedores pesquisarem várias maneiras de implementar os RNFs e, assim, reutilizar o conhecimento obtido com essa pesquisa. Para a academia, este trabalho gerou oportunidades de pesquisa que podem melhorar substancialmente a área de catálogos de RNFs, como por exemplo, descobrir quais catálogos são melhores dentro de um contexto específico.

Como trabalhos futuros, planeja-se criar um site para armazenar os dados e deixá-los mais fáceis de serem acessados, realizar um estudo na indústria para entender o uso do catálogo de RNFs em cenários reais. Então, os atuais desafios e questões que os praticantes enfrentaram em seu trabalho poderiam ser investigados. Além disso, incentiva-se a investigação adicional e possível expansão do banco de dados gerado por este trabalho.

REFERÊNCIAS

- AFREEN, N.; KHATOON, A.; SADIQ, M. A taxonomy of software's non-functional requirements. In: SPRINGER. **Proceedings of the second international conference on computer and communication technologies**. [S.l.], 2016. p. 47–53.
- BADAMPUDI, D.; WOHLIN, C.; PETERSEN, K. Experiences from using snowballing and database searches in systematic literature studies. In: ACM. **Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering**. [S.l.], 2015. p. 17.
- BOEHM, B. Architecture-based quality attribute synergies and conflicts. In: IEEE PRESS. **Proceedings of the Second International Workshop on Software Architecture and Metrics**. [S.l.], 2015. p. 29–34.
- CARVALHO, R. M.; ANDRADE, R. M. de C.; OLIVEIRA, K. M. de; SANTOS, I. de S.; BEZERRA, C. I. M. Quality characteristics and measures for human–computer interaction evaluation in ubiquitous systems. **Software Quality Journal**, Springer, v. 25, n. 3, p. 743–795, 2017.
- CHUNG, L.; NIXON, B. A.; YU, E.; MYLOPOULOS, J. **Non-functional requirements in software engineering**. [S.l.]: Springer Science & Business Media, 2012. v. 5.
- COUTINHO, E.; MOREIRA, C.; ANDRADE, R.; SOUZA, J. N. de. Conflict of requirements: a pattern for the treatment of conflicting non-functional requirements. In: ACM. **Proceedings of the 9th Latin-American Conference on Pattern Languages of Programming**. [S.l.], 2012. p. 4.
- CYSNEIROS, L. M. Evaluating the effectiveness of using catalogues to elicit non-functional requirements. In: **WER**. [S.l.: s.n.], 2007. p. 107–115.
- EGYED, A.; GRUNBACHER, P. Identifying requirements conflicts and cooperation: How quality attributes and automated traceability can help. **IEEE software**, IEEE, v. 21, n. 6, p. 50–58, 2004.
- HORIKOSHI, H.; NAKAGAWA, H.; TAHARA, Y.; OHSUGA, A. Dynamic reconfiguration in self-adaptive systems considering non-functional properties. In: ACM. **Proceedings of the 27th Annual ACM Symposium on Applied Computing**. [S.l.], 2012. p. 1144–1150.
- ISO/IEC, I. Software engineering–software product quality requirements and evaluation (square)–guide to square. **ISO Standard**, v. 25000, p. 2005, 2005.
- ISO/IEC, I. Iec 25010: 2011. **Systems and software engineering-Systems and software Quality Requirements and Evaluation (SQuARE)-System and software quality models**, 2011.
- JALALI, S.; WOHLIN, C. Systematic literature studies: database searches vs. backward snowballing. In: IEEE. **Proceedings of the 2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement**. [S.l.], 2012. p. 29–38.
- KHATTER, K.; KALIA, A. An integrated approach to capture semantics of requirement conflicts. In: SPRINGER. **International Conference on Computational Intelligence and Information Technology**. [S.l.], 2011. p. 826–831.

KOTONYA, G.; SOMMERVILLE, I. **Requirements engineering: processes and techniques**. [S.l.]: Wiley Publishing, 1998.

MAIRIZA, D.; ZOWGHI, D. Constructing a catalogue of conflicts among non-functional requirements. In: SPRINGER. **International Conference on Evaluation of Novel Approaches to Software Engineering**. [S.l.], 2010. p. 31–44.

MAIRIZA, D.; ZOWGHI, D.; NURMULIANI, N. An investigation into the notion of non-functional requirements. In: ACM. **Proceedings of the 2010 ACM Symposium on Applied Computing**. [S.l.], 2010. p. 311–317.

MEHTA, R.; WANG, H.; CHUNG, L. Dealing with nfrs for smart-phone applications: A goal-oriented approach. In: **Software Engineering Research, Management and Applications 2012**. [S.l.]: Springer, 2012. p. 113–125.

MONTAGUD, S.; ABRAHÃO, S.; INSFRAN, E. A systematic review of quality attributes and measures for software product lines. **Software Quality Journal**, Springer, v. 20, n. 3-4, p. 425–486, 2012.

NETO, P. A. d. M. S.; MACHADO, I. do C.; MCGREGOR, J. D.; ALMEIDA, E. S. D.; MEIRA, S. R. de L. A systematic mapping study of software product lines testing. **Information and Software Technology**, Elsevier, v. 53, n. 5, p. 407–423, 2011.

PETERSEN, K.; FELDT, R.; MUJTABA, S.; MATTSSON, M. Systematic mapping studies in software engineering. In: **Ease**. [S.l.: s.n.], 2008. v. 8, p. 68–77.

PETERSEN, K.; VAKKALANKA, S.; KUZNIARZ, L. Guidelines for conducting systematic mapping studies in software engineering: An update. **Information and Software Technology**, Elsevier, v. 64, p. 1–18, 2015.

QUIÑONES, D.; RUSU, C. How to develop usability heuristics: A systematic literature review. **Computer Standards & Interfaces**, Elsevier, v. 53, p. 89–122, 2017.

RODRIGUES, A. S. Uma abordagem para definição de guias de elicitação de requisitos não-funcionais. Tese (Doutorado) - Universidade de Fortaleza. Programa de Informática Aplicada, Fortaleza,, 2016.

SANTOS, I. de S.; ANDRADE, R. M. de C.; ROCHA, L. S.; MATALONGA, S.; OLIVEIRA, K. M. de; TRAVASSOS, G. H. Test case design for context-aware applications: Are we there yet? **Information and Software Technology**, Elsevier, v. 88, p. 1–16, 2017.

SVENSSON, R. B.; HOST, M.; REGNELL, B. Managing quality requirements: A systematic review. In: IEEE. **2010 36th EUROMICRO Conference on Software Engineering and Advanced Applications**. [S.l.], 2010. p. 261–268.

TABASSUM, M. R.; SIDDIK, M. S.; SHOYAIB, M.; KHALED, S. M. Determining interdependency among non-functional requirements to reduce conflict. In: IEEE. **2014 International Conference on Informatics, Electronics & Vision (ICIEV)**. [S.l.], 2014. p. 1–6.

THAYER, R. H.; DORFMAN, M. **System and software requirements engineering**. [S.l.]: IEEE Computer Society Press, 1995.

WIEGERS, K.; BEATTY, J. **Software requirements**. [S.l.]: Pearson Education, 2013.

WIERINGA, R.; MAIDEN, N.; MEAD, N.; ROLLAND, C. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. **Requirements engineering**, Springer, v. 11, n. 1, p. 102–107, 2006.

WOHLIN, C.; RUNESON, P.; NETO, P. A. d. M. S.; ENGSTRÖM, E.; MACHADO, I. do C.; ALMEIDA, E. S. D. On the reliability of mapping studies in software engineering. **Journal of Systems and Software**, Elsevier, v. 86, n. 10, p. 2594–2610, 2013.

APÊNDICE A – TABELAS REFERENTES À RQ 1

Tabela 7 – Lista de artigos - parte 1

Id	Artigo Aceito	Id Catálogo
1	Carvalho, R. M., Andrade, R. M., & de Oliveira, K. M. (2018, May). Towards a catalog of conflicts for HCI quality characteristics in UbiComp and IoT applications: Process and first results. In 2018 12th International Conference on Research Challenges in Information Science (RCIS) (pp. 1-6). IEEE.	1
2	Sadi, M. H., & Yu, E. (2017, June). Accommodating Openness Requirements in Software Platforms: A Goal-Oriented Approach. In International Conference on Advanced Information Systems Engineering (pp. 44-59). Springer, Cham.	2,3,4,5 e 6
3	Uchôa, A. G., Lima, L. P., Bezerra, C. I., Monteiro, J. M., & Andrade, R. M. (2017, May). DyMMer-NFP: Modeling Non-functional Properties and Multiple Context Adaptation Scenarios in Software Product Lines. In International Conference on Software Reuse (pp. 175-183). Springer, Cham.	7
4	Totiya, S., & Senivongse, T. (2017). Framework to Support Cloud Service Selection Based on Service Measurement Index. In Proceedings of the World Congress on Engineering and Computer Science (Vol. 1).	8
5	Soad, G. W., Duarte Filho, N. F., & Barbosa, E. F. (2016, October). Quality evaluation of mobile learning applications. In 2016 IEEE Frontiers in Education Conference (FIE) (pp. 1-8). IEEE.	9
6	Villa, L., Cabezas, I., Lopez, M., & Casas, O. (2016, July). Towards a sustainable architectural design by an adaptation of the architectural driven design method. In International Conference on Computational Science and Its Applications(pp. 71-86). Springer, Cham.	10
7	Sarmiento, E., do Prado Leite, J. C. S., & Almentero, E. (2015, August). Using correctness, consistency, and completeness patterns for automated scenarios verification. In 2015 IEEE Fifth International Workshop on Requirements Patterns (RePa)(pp. 47-54). IEEE.	11
8	Hu, H., Ma, Q., Zhang, T., Tan, Y., Xiang, H., Fu, C., & Feng, Y. (2015). Semantic modelling and automated reasoning of non-functional requirement conflicts in the context of softgoal interdependencies. IET Software, 9(6), 145-156.	12
9	de Castro Leal, A. L., Braga, J. L., & da Cruz, S. M. S. (2015, August). Cataloguing provenance-awareness with patterns. In 2015 IEEE Fifth International Workshop on Requirements Patterns (RePa) (pp. 9-16). IEEE.	13
10	Zinovatna, O., & Cysneiros, L. M. (2015, August). Reusing knowledge on delivering privacy and transparency together. In 2015 IEEE Fifth International Workshop on Requirements Patterns (RePa) (pp. 17-24). IEEE.	14, 15, 16 e 17
11	Cunha, H., & do Prado Leite, J. C. S. (2014, August). Reusing non-functional patterns in i modeling. In 2014 IEEE 4th International Workshop on Requirements Patterns (RePa) (pp. 25-32). IEEE.	18
12	Cunha, H., do Prado Leite, J. C. S., Duboc, L., & Werneck, V. (2013, July). The challenges of representing transparency as patterns. In 2013 3rd International Workshop on Requirements Patterns (RePa) (pp. 25-30). IEEE.	19

Fonte: Elaborado pela autora.

Tabela 8 – Lista de artigos - parte 2

Id	Artigo Aceito	Id Catálogo
13	Cardoso, E., Almeida, J. P. A., Guizzardi, R. S., & Guizzardi, G. (2011). A method for eliciting goals for business process models based on non-functional requirements catalogues. <i>International Journal of Information System Modeling and Design (IJISMD)</i> , 2(2), 1-18.	20
14	Subramanian, N., Drager, S., & McKeever, W. (2014). Designing trustworthy software systems using the NFR approach. In <i>Emerging Trends in ICT Security</i> (pp. 203-225). Morgan Kaufmann.	21
15	Serrano, M., & Serrano, M. (2013). Ubiquitous, Pervasive and Mobile Computing: A Reusable-Models-based Non-Functional Catalogue. In <i>ER@ BR</i> .	22
16	Macasaet, R. J., Noguera, M., Rodríguez, M. L., Garrido, J. L., Supakkul, S., & Chung, L. (2013, May). A requirements-based approach for representing micro-business patterns. In <i>IEEE 7th International Conference on Research Challenges in Information Science (RCIS)</i> (pp. 1-12). IEEE.	23 e 24
17	Zhu, M. X., Luo, X. X., Chen, X. H., & Wu, D. D. (2012). A non-functional requirements tradeoff model in trustworthy software. <i>Information Sciences</i> , 191, 61-75.	25
18	Mehta, R., Wang, H., & Chung, L. (2012). Dealing with NFRs for smart-phone applications: A goal-oriented approach. In <i>Software Engineering Research, Management and Applications 2012</i> (pp. 113-125). Springer, Berlin, Heidelberg.	26, 27, 28, 29 e 30
19	Mairiza, D., & Zowghi, D. (2010, July). Constructing a catalogue of conflicts among non-functional requirements. In <i>International Conference on Evaluation of Novel Approaches to Software Engineering</i> (pp. 31-44). Springer, Berlin, Heidelberg.	31
20	Wang, H., Mehta, R., Supakkul, S., & Chung, L. (2011, September). Rule-based context-aware adaptation using a goal-oriented ontology. In <i>Proceedings of the 2011 international workshop on Situation activity & goal awareness</i> (pp. 67-76). ACM.	32, 33 e 34
21	Hill, T., Supakkul, S., & Chung, L. (2010, September). Run-time monitoring of system performance: A goal-oriented and system architecture simulation approach. In <i>2010 First International Workshop on Requirements@ Run. Time</i> (pp. 31-40). IEEE.	35
22	Yrjönen, A., & Merilinna, J. (2009, October). Extending the NFR Framework with Measurable NonFunctional Requirements. In <i>NF-PinDSML@ MoDELS</i> .	36
23	Cysneiros, L. M., Breitman, K. K., Lopez, C., & Astudillo, H. (2008, October). Querying Software Interdependence Graphs. In <i>2008 32nd Annual IEEE Software Engineering Workshop</i> (pp. 108-112). IEEE.	37
24	Burgess, C., Krishna, A., & Jiang, L. (2009, August). Towards optimizing non-functional requirements. In <i>2009 Ninth International Conference on Quality Software</i> (pp. 269-277). IEEE.	38
25	López, C., Inostroza, P., Cysneiros, L. M., & Astudillo, H. (2009). Visualization and comparison of architecture rationale with semantic web technologies. <i>Journal of Systems and Software</i> , 82(8), 1198-1210.	39, 40 e 41
26	Hill, T., Supakkul, S., & Chung, L. (2009, November). Confirming and reconfirming architectural decisions on scalability: a goal-driven simulation approach. In <i>OTM Confederated International Conferences"On the Move to Meaningful Internet Systems"</i> (pp. 327-336). Springer, Berlin, Heidelberg.	42 e 43

Fonte: Elaborado pela autora.

Tabela 9 – Lista de artigos - parte 3

Id	Artigo Aceito	Id Catálogo
27	Kumar, P., Subramanian, N., & Zhang, K. (2008, September). Evaluation of Information Visualization Tools Using the NFR Approach. In International Conference on Advances in Visual Information Systems (pp. 44-55). Springer, Berlin, Heidelberg.	44
28	Cysneiros, L. M. (2007, May). Evaluating the Effectiveness of Using Catalogues to Elicit Non-Functional Requirements. In WER (pp. 107-115).	45, 46, 47, 48, 49, 50, 51, 52 e 53
29	Vega, J. P. C., Franch, X., & Quer, C. (2007, February). Towards a unified catalogue of non-technical quality attributes to support COTS-based systems lifecycle activities. In 2007 Sixth International IEEE Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems (ICCBSS'07) (pp. 21-32). IEEE.	54
30	Cleland-Huang, J., Settimi, R., BenKhadra, O., Berezanskaya, E., & Christina, S. (2005, May). Goal-centric traceability for managing non-functional requirements. In Proceedings. 27th International Conference on Software Engineering, 2005. ICSE 2005. (pp. 362-371). IEEE.	55 e 56
31	Subramanian, N., & Chung, L. (2003). SA3—a tool for supporting adaptable software architecture generation for embedded systems. <i>Computer Standards & Interfaces</i> , 25(3), 283-290.	57
32	Tahvildari, L., & Kontogiannis, K. (2002). A software transformation framework for quality-driven object-oriented re-engineering. In International Conference on Software Maintenance, 2002. Proceedings. (pp. 596-605). IEEE.	58
33	Nixon, B. A. (2000). Management of performance requirements for information systems. <i>IEEE Transactions on Software Engineering</i> , 26(12), 1122-1146.	59 e 60
34	Egyed, A., & Grunbacher, P. (2004). Identifying requirements conflicts and cooperation: How quality attributes and automated traceability can help. <i>IEEE software</i> , 21(6), 50-58.	61 e 62
35	Berander, P., Damm, L. O., Eriksson, J., Gorschek, T., Henningsson, K., Jönsson, P., ... & Tomaszewski, P. (2005). Software quality attributes and trade-offs. Blekinge Institute of Technology.	63, 64 e 65
36	Maia, M. E., Rocha, L. S., & Andrade, R. (2009, July). Requirements and challenges for building service-oriented pervasive middleware. In Proceedings of the 2009 international conference on Pervasive services (pp. 93-102). ACM.	66, 67, 68, 69, 70 e 71
37	Feitosa, D., Ampatzoglou, A., Avgeriou, P., & Nakagawa, E. Y. (2015, May). Investigating quality trade-offs in open source critical embedded systems. In 2015 11th International ACM SIGSOFT Conference on Quality of Software Architectures (QoSA) (pp. 113-122). IEEE.	72 e 73
38	García-Mireles, G. A., de la Rubia, M. Á. M., García, F., & Piattini, M. (2015, April). Identifying quality characteristic interactions during software development. In 2015 International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE) (pp. 78-89). IEEE.	74
39	ISO/IEC 25010 - Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuARE) - System and Software Quality Models.	75 e 76

Fonte: Elaborado pela autora.

Tabela 10 – Lista de artigos - parte 4

Id	Artigo Aceito	Id Catálogo
40	Hammani, F. Z. (2014, May). Survey of non-functional requirements modeling and verification of software product lines. In 2014 IEEE Eighth International Conference on Research Challenges in Information Science (RCIS) (pp. 1-6). IEEE.	77
41	Duarte Filho, N. F., & Barbosa, E. F. (2013). A requirements catalog for mobile learning environments. In Proceedings.	78
42	Serrano, M., & do Prado Leite, J. C. S. (2011, August). Capturing transparency-related requirements patterns through argumentation. In 2011 First International Workshop On Requirements Patterns (pp. 32-41). IEEE.	79
43	Sadana, V., & Liu, X. F. (2007, July). Analysis of conflicts among non-functional requirements using integrated analysis of functional and non-functional requirements. In 31st annual international computer software and applications conference (COMPSAC 2007) (Vol. 1, pp. 215-218). IEEE.	80
44	Pashazadeh, S. (2011, October). Modeling non functional requirements in designing middleware for pervasive healthcare system. In 2011 5th International Conference on Application of Information and Communication Technologies (AICT) (pp. 1-5). IEEE.	81
45	Boehm, B., & In, H. (1996). Aids for identifying conflicts among quality requirements. IEEE Software, 13(2), 25-35.	82, 83 e 84
46	Chung, L., Nixon, B. A., & Yu, E. (1995, April). Using non-functional requirements to systematically select among alternatives in architectural design. In Proc. 1st Int. Workshop on Architectures for Software Systems, Seattle, Washington(pp. 31-43).	85, 86, 87, 88 e 89
47	Mylopoulos, J., Chung, L., Liao, S., Wang, H., & Yu, E. (2001). Exploring alternatives during requirements analysis. IEEE Software, 18(1), 92-96.	90 e 91
48	Cysneiros, L. M., & do Prado Leite, J. S. (1999). Integrating non-functional requirements into data modeling. In Proceedings IEEE International Symposium on Requirements Engineering (Cat. No. PR00188) (pp. 162-171). IEEE.	92
49	Paim, F. R. S., & Castro, J. (2002). Enhancing Data Warehouse Design with the NFR Framework. Wer, 2, 40-57.	93, 94, 95, 96 e 97
50	Andreopoulos, B. (2004, December). Satisficing the Conflicting Software Qualities of Maintainability and Performance at the Source Code Level. In WER (pp. 176-188).	98 e 99
51	de Sousa, G. M. C., & Castro, J. (2003). Towards a Goal-Oriented Requirements Methodology Based on the Separation of Concerns Principle. In WER (pp. 223-239).	100
52	Silva, C. T., Castro, J., & Tedesco, P. A. (2003). Requirements for Multi-Agent Systems. WER, 2003, 198-212.	101
53	Carvalho Vega, J. P., Franch Gutiérrez, J., & Quer, C. (2015). Improving quality model construction through knowledge reuse. In Anais do WER15-Workshop em Engenharia de Requisitos.	102

Fonte: Elaborado pela autora.

APÊNDICE B – TABELAS REFERENTES À RQ 2

Tabela 11 – Lista de catálogos - parte 1

ID Cat	Autores do Catálogo (Id do Artigo ou Referência)
1	Wiegers, K., & Beatty, J. (2013). Software requirements. Pearson Education.
2	2
3	2
4	2
5	2
6	2
7	3
8	4
9	5
10	6
11	7
12	Cleland-Huang, J., Marrero, W., Berenbach, B.: ‘Goal-centric traceability: using virtual plumb lines to maintain critical systemic qualities’, IEEE Trans. Softw. Eng., 2008, 34, (5), pp. 685–699.
13	9
14	10
15	Leite, J.C.. S., Cappelli, C. (2010). Software transparency. Business & Information Systems Engineering:, 2(3), 127-139.
16	Cappelli, C., Cunha, H., Gonzalez-Baixauli, B., & do Prado Leite, J. C. S. (2010, March). Transparency versus security: early analysis of antagonistic requirements. In Proceedings of the 2010 ACM symposium on applied computing (pp. 298-305). ACM.
17	10
18	11
19	12
20	13
21	14
22	15
23	16
24	16
25	17
26	18
27	18
28	18
29	18
30	18
31	19
32	20
33	20
34	20

Fonte: Elaborado pela autora.

Tabela 12 – Lista de catálogos - parte 2

Id Cat	Autores do Catálogo (Id do Artigo ou Referência)
35	21
36	22
37	23
38	24
39	25
40	25
41	25
42	26
43	26
44	27
45	Cysneiros, L. M., Werneck, V. M., & Kushniruk, A. (2005, August). Reusable knowledge for satisficing usability requirements. In null (pp. 463-464). IEEE.
46	< http://www.math.yorku.ca/~cysneiro/nfrs/Traceability.htm >
47	< http://www.math.yorku.ca/~cysneiro/nfrs/Traceability.htm >
48	Chung, L., Nixon, B., Yu, E. and Mylopoulos, J. “Non-Functional Requirements in Software Engineering” Kluwer Academic Publishers 2000.
49	Chung, L., Nixon, B., Yu, E. and Mylopoulos, J. “Non-Functional Requirements in Software Engineering” Kluwer Academic Publishers 2000.
50	Chung, L., Nixon, B., Yu, E. and Mylopoulos, J. “Non-Functional Requirements in Software Engineering” Kluwer Academic Publishers 2000.
51	Chung, L., Nixon, B., Yu, E. and Mylopoulos, J. “Non-Functional Requirements in Software Engineering” Kluwer Academic Publishers 2000.
52	Chung, L., Nixon, B., Yu, E. and Mylopoulos, J. “Non-Functional Requirements in Software Engineering” Kluwer Academic Publishers 2000.
53	Chung, L., Nixon, B., Yu, E. and Mylopoulos, J. “Non-Functional Requirements in Software Engineering” Kluwer Academic Publishers 2000.
54	Vega, J. P. C., Franch, X., & Quer, C. (2007, February). Towards a unified catalogue of non-technical quality attributes to support COTS-based systems lifecycle activities. In 2007 Sixth International IEEE Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems (ICCBSS'07) (pp. 21-32). IEEE.
55	30
56	30
57	31
58	32
59	33
60	33
61	34
62	34
63	McCall, J. A., Richards, P. K., & Walters, G. F. (1977). Factors in software quality. volume i. concepts and definitions of software quality. GENERAL ELECTRIC CO SUNNYVALE CA.
64	McCall, J. A., Richards, P. K., & Walters, G. F. (1977). Factors in software quality. volume i. concepts and definitions of software quality. GENERAL ELECTRIC CO SUNNYVALE CA.
65	Boehm, Barry W., Brown, J. R., and Lipow, M.: Quantitative evaluation of software quality, International Conference on Software Engineering, Proceedings of the 2nd international conference on Software engineering, 1976.
66	36
67	36
68	36
69	36

Fonte: Elaborado pela autora.

Tabela 13 – Lista de catálogos - parte 3

Id Cat	Autores do Catálogo (Id do Artigo ou Referência)
70	36
71	36
72	37
73	37
74	38
75	39
76	39
77	40
78	41
79	42
80	43
81	44
82	45
83	45
84	45
85	46
86	46
87	46
88	46
89	46
90	47
91	47
92	48
93	49
94	49
95	49
96	49
97	49
98	50
99	50
100	51
101	52
102	53

Fonte: Elaborado pela autora.

APÊNDICE C – TABELA REFERENTE À RQ 2.2

Tabela 14 – Características de qualidade e a quantidade de citações

Característica	Quantidade	Característica	Quantidade
performance	34	security	30
usability	23	reliability	14
reusability	14	maintainability	13
cost	12	functionality	9
modifiability	9	safety	9
accuracy	8	portability	8
privacy	8	efficiency	7
flexibility	7	interoperability	7
extensibility	6	ubiquity	6
availability	5	correctness	5
assurance	4	confidentiality	4
effectiveness	4	integrity	4
transparency	4	evolvability	3
multidimensionality	3	openess	3
recoverability	3	satisfaction	3
scalability	3	testability	3
traceability	3	trust	3
understandability	3	autonomy	2
communication	2	comprehensibility	2
dependability	2	mobility	2
operational restrictions	2	pedagogical	2
quality	2	quietness	2
robustness	2	sociability	2
support	2	user-friendliness	2
accessibility	1	accountability	1
adaptability	1	agility	1
as-is utility	1	awareness	1
compatibility	1	completeness	1
context coverage	1	delivery time	1
development time	1	empathy	1
fault-tolerance	1	financial	1
freedom from risk	1	installability	1
integrability	1	long battery life	1
low risk	1	maturity	1
message exchange	1	ownership	1
pervasiveness	1	product	1
provenance	1	public service	1
qos	1	resource behavior	1
responsiveness	1	service coordination	1
service description	1	service discovery	1
service level	1	supplier	1
sustainability	1	time behavior	1
verifiability	1	visualizability	1

Fonte: Elaborado pela autora.

APÊNDICE D – *LINKS*

Este apêndice apresenta os *links* para acesso aos dados do estudo:

- Planilha de extração dos dados
- Planilha de análise dos dados
- Pasta com catálogos encontrados