



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CENTRO DE CIÊNCIAS E TECNOLOGIA (CCT)**  
**DEPARTAMENTO DE COMPUTAÇÃO**  
**MESTRADO E DOUTORADO EM CIÊNCIA DA COMPUTAÇÃO (MDCC)**  
**MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO**

**AMANDA OLIVEIRA DE SOUSA**

**MEDIDAS PARA AVALIAÇÃO DA CONFIABILIDADE DE SISTEMAS**  
**AUTOADAPTATIVOS**

**FORTALEZA**

**2019**

AMANDA OLIVEIRA DE SOUSA

MEDIDAS PARA AVALIAÇÃO DA CONFIABILIDADE DE SISTEMAS  
AUTOADAPTATIVOS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Mestrado e Doutorado em Ciência da Computação (MDCC) do Centro de Ciências e Tecnologia (CCT) da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciências da Computação. Área de Concentração: Qualidade de Software.

Orientadora: Prof. Dr. Rossana Maria de Castro Andrade.

Coorientadora: Prof. Dr. Carla Ilane Moreira Bezerra.

Coorientador: Prof. Dr. José Maria da Silva Monteiro Filho.

FORTALEZA

2019

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

---

S696m Sousa, Amanda Oliveira de.  
Medidas para avaliação da confiabilidade de sistemas autoadaptativos / Amanda Oliveira de Sousa. – 2019.  
103 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2019.

Orientação: Profa. Dra. Rossana Maria de Castro Andrade.

Coorientação: Profa. Dra. Carla Ilane Moreira Bezerra.

1. Sistemas autoadaptativos. 2. Reconfigurações Dinâmicas. 3. Avaliação de Qualidade. 4. Confiabilidade. 5. Medidas de Qualidade. I. Título.

CDD 005

---

AMANDA OLIVEIRA DE SOUSA

MEDIDAS PARA AVALIAÇÃO DA CONFIABILIDADE DE SISTEMAS  
AUTOADAPTATIVOS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Mestrado e Doutorado em Ciência da Computação (MDCC) do Centro de Ciências e Tecnologia (CCT) da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciências da Computação. Área de Concentração: Qualidade de Software.

Aprovada em: 26/11/2019.

BANCA EXAMINADORA

---

Prof. Dr. Rossana Maria de Castro Andrade (Orientadora)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Carla Ilane Moreira Bezerra (Coorientadora)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. José Maria da Silva Monteiro Filho  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Gibeon Soares de Aquino Júnior  
Universidade Federal do Rio Grande do Norte (UFRN)

---

Prof. Dr. Rafael Capilla Sevilla  
Universidad Rey Juan Carlos (URJC)

---

Prof. Dr. Márcio Espíndola Freire Maia  
Universidade Federal do Ceará (UFC)

À minha família, por sua capacidade de acreditar  
e investir em mim.

## **AGRADECIMENTOS**

À Prof<sup>a</sup>. Dr<sup>a</sup>. Rossana Maria de Castro Andrade por me orientar em minha dissertação de mestrado e por me ajudar a me desenvolver como pesquisadora.

À Prof<sup>a</sup>, Dr<sup>a</sup> Carla Ilane Moreira Bezerra por me coorientar em minha dissertação de mestrado, pela sua paciência e seus incentivos que me ajudaram a ir mais longe.

Ao Prof. Dr. José Maria da Silva Monteiro Filho por me coorientar em minha dissertação de mestrado e por toda a sua positividade sempre.

Aos meus colegas do laboratório GREat, com quem compartilhei ideais e troquei experiências. Vocês me ajudaram a crescer como pesquisadora e como pessoa, obrigada.

Ao meu companheiro de todas as horas Rafael Gaspar, pelo apoio, paciência, amor e cuidado comigo durante o desenvolvimento deste trabalho.

Aos meus pais e irmãos que nos momentos de minha ausência dedicados ao estudo superior, sempre me apoiaram, me incentivaram e me impulsionaram a lutar pelo meu futuro.

Agradeço a todos os professores por me proporcionarem o conhecimento não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação profissional, por tanto que se dedicaram a mim, não somente por terem me ensinado, mas por terem me feito aprender.

“A qualidade nunca se obtém por acaso, ela é sempre o resultado do esforço inteligente.”  
(JOHN RUSKIN)

## RESUMO

Sistemas autoadaptativos (SAS) são capazes de alterar seu próprio comportamento em tempo de execução e estão associados a domínios como espaços inteligentes, sistemas de aviação e *healthcare*, apresentando operações dinâmicas e complexas, uso de sensores e comunicação com outros dispositivos. Dessa forma, para prover serviços de forma adequada requerem altos níveis de qualidade. Uma forma de ajudar a garantir a qualidade dos SAS é realizando avaliações de qualidade que são em sua maioria baseadas em medidas de software que, por sua vez, buscam quantificar atributos do sistema em diversas perspectivas. No entanto, para avaliar sistemas autoadaptativos, as medidas devem conter elementos adequados as especificidades dos SAS, como reconfiguração dinâmica e comunicação com sensores, atuadores e outros sistemas. Visando identificar características, atributos e medidas de qualidade importantes para SAS, primeiro, neste trabalho, foi realizado um mapeamento sistemático. A característica confiabilidade foi então identificada como uma das características mais recorrentes nos estudos selecionados. Entretanto, com poucas medidas e sub-características pobremente exploradas. Além disso, vale ressaltar que a confiabilidade tem grande impacto no funcionamento dos sistemas autoadaptativos, uma vez que SAS com baixos níveis de confiabilidade podem apresentar problemas de desempenho, operações de adaptação incorretas e má qualidade no fornecimento de serviços. Sendo assim, visando prover suporte à avaliação da confiabilidade dos SAS e, conseqüentemente, contribuir com a melhoria da garantia da qualidade desses sistemas, este trabalho tem o objetivo de propor um conjunto de medidas de software direcionadas a confiabilidade de SAS e avaliá-las quanto a sua aplicabilidade e seus benefícios.

**Palavras-chave:** sistemas autoadaptativos; reconfigurações dinâmicas; avaliação de qualidade; confiabilidade; medidas de qualidade.

## ABSTRACT

Self-adaptive systems (SAS) are capable of changing their behavior at runtime and are associated with domains such as intelligent spaces, aviation systems, and *healthcare*. These systems present dynamic and complex operations, and to provide adequate services, they require high levels of quality. One way to help ensure the quality of these systems is to perform quality assessments that are mostly based on software measures that, in turn, seek to quantify system attributes from various perspectives. However, in order to evaluate the quality of self-adaptive systems, the measures must contain elements appropriate to the specificities of SAS, such as dynamic reconfiguration and communication with sensors, actuators, and other systems. Aiming to identify characteristics, attributes, and quality measures important for SAS, a systematic mapping was first performed. In this work, the reliability characteristic was then identified as one of the most recurrent characteristics in the selected studies and according to the studies selected, reliability has a significant impact on the functioning of self-adaptive systems. The systematic mapping also showed that reliability characteristic had few measures, and its sub characteristics are poorly explored. Then, this work proposes a set of software measures aimed at the reliability of SAS and evaluate them related to their applicability and benefits. At the end, this work aims to provide support to the evaluation of SAS reliability and, consequently, contribute to the improvement of the quality assurance of these systems.

**Keywords:** self-adaptive systems; dynamic reconfigurations; quality evaluation; reliability; quality measures.

## LISTA DE FIGURAS

Figura 1 – Metodologia de Desenvolvimento da Dissertação. . . . .	17
Figura 2 – <i>Loop</i> de controle. Adaptado de (CHENG <i>et al.</i> , 2009a) . . . . .	21
Figura 3 – Propriedades de sistemas <i>self</i> -.*. Adaptado de (SALEHIE; TAHVILDARI, 2009) . . . . .	22
Figura 4 – Modelo de Qualidade do Produto. Adaptado de (ISO, 2011a). . . . .	26
Figura 5 – Modelo de Referência para a Medição da Qualidade do Produto. Adaptado de (ISO, 2011a). . . . .	31
Figura 6 – Paradigma GQM. Adaptado de Solingen e Berghout (1999) . . . . .	32
Figura 7 – Processo de mapeamento sistemático adotado. Adaptado de Kitchenham e Charters (2007) . . . . .	37
Figura 8 – Processo de Seleção do Mapeamento Sistemático . . . . .	42
Figura 9 – Características capturadas no mapeamento sistemático por número de ocorrência	45
Figura 10 – Número de medidas por características e tipos de estudo. . . . .	47
Figura 11 – Número de estudos por tipo de contribuição e área . . . . .	50
Figura 12 – Formação dos Especialistas . . . . .	53
Figura 13 – Grau de conhecimento dos participantes da definição de medidas por tópico	55
Figura 14 – Planilha de Abstração. Adaptada de (SOLINGEN; BERGHOUT, 1999). . .	55
Figura 15 – Processo de Preenchimento das Planilhas de Abstração . . . . .	56
Figura 16 – Grau de conhecimento dos participantes da revisão por tópico . . . . .	58
Figura 17 – Atividades da Prova de Conceito . . . . .	70
Figura 18 – Tela inicial do TAS ( <i>Tele Assistance System</i> ) . . . . .	71

## LISTA DE TABELAS

Tabela 1 – Formulário de Extração. . . . .	40
Tabela 2 – Estudos por base de dados . . . . .	40
Tabela 3 – Conjunto Final de Estudos Seleccionados . . . . .	43
Tabela 4 – Número de sub-características e atributos de qualidade por característica . . . . .	45
Tabela 5 – Classificação de estudos quanto ao tipo de avaliação. Adaptado de Garcés <i>et al.</i> (2017) . . . . .	46
Tabela 6 – Classificação por tipos de pesquisa . . . . .	49
Tabela 7 – Definição do Objetivo (SOLINGEN; BERGHOUT, 1999) . . . . .	52
Tabela 8 – Classificação das medidas extraídas no mapeamento sistemático . . . . .	54
Tabela 9 – Distribuição da quantidade de medidas definidas por sub-característica . . . . .	57
Tabela 10 – <i>Checklist</i> de avaliação de medidas. Adaptado de Barcellos <i>et al.</i> (2010) . . . . .	59
Tabela 11 – Medidas de Maturidade . . . . .	63
Tabela 12 – Medidas de Disponibilidade . . . . .	64
Tabela 13 – Medidas de Recuperabilidade . . . . .	65
Tabela 14 – Medidas de Tolerância a Falhas . . . . .	66
Tabela 15 – Medida de Robustez . . . . .	67
Tabela 16 – Resultados da coleta e cálculo das medidas de maturidade . . . . .	75
Tabela 17 – Resultados da coleta e cálculo das medidas de disponibilidade . . . . .	76
Tabela 18 – Resultados da coleta e cálculo das medidas de recuperabilidade . . . . .	76
Tabela 19 – Resultados da coleta e cálculo das medidas de tolerância a falhas . . . . .	78
Tabela 20 – Resultados da coleta e cálculo das medidas de robustez . . . . .	78

## LISTA DE ABREVIATURAS E SIGLAS

CECILIA	biblioteCa de mEdidas de ConfIabiLidade para sIstemas Autoadaptativos
GQM	<i>Goal Question Metric</i>
GREat	Grupo de Redes de Computadores, Engenharia de Software e Sistemas
IDE	<i>Integrated Development Environment</i>
IoT	<i>Internet of Things</i>
ISO	Organização Internacional de Normalização
LPSDs	Linhas de Produto de Software Dinâmicas
MS	Mapeamento Sistemático
RAM	<i>Random Access Memory</i>
SAS	<i>Self-Adaptive Systems</i>
SQuaRE	<i>Software Product Quality Requirement and Evaluation</i>
TAS	<i>Tele Assistance System</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
<b>1.1</b>	<b>Contextualização</b>	<b>14</b>
<b>1.2</b>	<b>Motivação</b>	<b>16</b>
<b>1.3</b>	<b>Objetivos e Metodologia</b>	<b>17</b>
<b>1.4</b>	<b>Estrutura da Dissertação</b>	<b>19</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>20</b>
<b>2.1</b>	<b>Sistemas Autoadaptativos</b>	<b>20</b>
<b>2.2</b>	<b>Qualidade de Software</b>	<b>24</b>
<b>2.2.1</b>	<i>Modelo de Qualidade SQuaRE</i>	<b>24</b>
<b>2.2.2</b>	<i>Característica de Confiabilidade</i>	<b>29</b>
<b>2.2.3</b>	<i>Medição de Software</i>	<b>30</b>
<b>2.2.4</b>	<i>Método Goal Question Métric - GQM</i>	<b>31</b>
<b>2.3</b>	<b>Avaliação da Qualidade em Sistemas Autoadaptativos</b>	<b>33</b>
<b>2.4</b>	<b>Considerações Finais</b>	<b>34</b>
<b>3</b>	<b>QUALIDADE EM SAS: MAPEAMENTO SISTEMÁTICO</b>	<b>36</b>
<b>3.1</b>	<b>Metodologia do Mapeamento Sistemático</b>	<b>36</b>
<b>3.2</b>	<b>Planejamento</b>	<b>37</b>
<b>3.3</b>	<b>Execução</b>	<b>40</b>
<b>3.4</b>	<b>Análise</b>	<b>41</b>
<b>3.4.1</b>	<i>QP1. Quais atributos de qualidade tem sido utilizados para avaliar sistemas autoadaptativos?</i>	<b>44</b>
<b>3.4.2</b>	<i>QP2. Quais medidas de qualidade tem sido utilizadas para avaliar sistemas autoadaptativos?</i>	<b>46</b>
<b>3.4.3</b>	<i>QP3. Quais soluções têm sido propostas para avaliar sistemas autoadaptativos?</i>	<b>48</b>
<b>3.5</b>	<b>Discussão</b>	<b>50</b>
<b>3.6</b>	<b>Considerações Finais</b>	<b>51</b>
<b>4</b>	<b>MEDIDAS DE CONFIABILIDADE PARA SAS</b>	<b>52</b>
<b>4.1</b>	<b>Definição de Medidas de Confiabilidade</b>	<b>52</b>
<b>4.1.1</b>	<i>Definição do Objetivo</i>	<b>52</b>

4.1.2	<i>Classificação</i> . . . . .	52
4.1.3	<i>Definição de Medidas com a Utilização do Método GQM</i> . . . . .	53
4.1.3.1	<i>Perfil dos Participantes</i> . . . . .	53
4.1.3.2	<i>Preenchimento da Planilha de Abstração</i> . . . . .	55
4.2	<b>Revisão do Conjunto Inicial de Medidas de Confiabilidade</b> . . . . .	57
4.3	<b>Documentação do Conjunto Final de Medidas de Confiabilidade</b> . . . . .	59
4.4	<b>Considerações Finais</b> . . . . .	68
5	<b>AVALIAÇÃO DAS MEDIDAS DE CONFIABILIDADE</b> . . . . .	69
5.1	<b>Prova de Conceito</b> . . . . .	69
5.1.1	<i>Biblioteca CECILIA</i> . . . . .	70
5.1.2	<i>Planejamento</i> . . . . .	70
5.1.3	<i>Execução</i> . . . . .	72
5.1.4	<i>Análise dos Resultados</i> . . . . .	74
5.1.4.1	<i>Maturidade</i> . . . . .	74
5.1.4.2	<i>Disponibilidade</i> . . . . .	75
5.1.4.3	<i>Recuperabilidade</i> . . . . .	76
5.1.4.4	<i>Tolerância a falhas</i> . . . . .	77
5.1.4.5	<i>Robustez</i> . . . . .	77
5.2	<b>Discussão</b> . . . . .	78
5.3	<b>Ameaças à validade</b> . . . . .	79
5.3.1	<i>Validade interna</i> . . . . .	79
5.3.2	<i>Validade de construção</i> . . . . .	79
5.3.3	<i>Validade externa</i> . . . . .	80
5.4	<b>Considerações finais</b> . . . . .	81
6	<b>CONCLUSÃO</b> . . . . .	82
6.1	<b>Visão Geral</b> . . . . .	82
6.2	<b>Resultados Alcançados</b> . . . . .	83
6.3	<b>Produções Bibliográficas</b> . . . . .	84
6.4	<b>Trabalhos Futuros</b> . . . . .	85
	<b>REFERÊNCIAS</b> . . . . .	88
	<b>APÊNDICE A –PLANILHAS GQM DO PARTICIPANTE 1</b> . . . . .	96
	<b>APÊNDICE B –PLANILHAS GQM DO PARTICIPANTE 2</b> . . . . .	98

<b>APÊNDICE C –PLANILHAS GQM DO PARTICIPANTE 3</b>	<b>. . . . . 99</b>
<b>APÊNDICE D –PLANILHAS GQM DO PARTICIPANTE 4</b>	<b>. . . . . 100</b>
<b>APÊNDICE E –PLANILHAS GQM DO PARTICIPANTE 5</b>	<b>. . . . . 103</b>

# 1 INTRODUÇÃO

Esta dissertação de mestrado tem por objetivo propor um conjunto de medidas de qualidade para avaliar a confiabilidade de sistemas autoadaptativos. Este Capítulo apresenta, na Seção 1.1, o domínio e o contexto em que o trabalho está inserido. A Seção 1.2 aponta a motivação para a realização deste trabalho. A Seção 1.3 apresenta os objetivos desta dissertação e discorre sobre os métodos utilizados no desenvolvimento deste trabalho. Por fim, a estrutura da dissertação é apresentada na Seção 1.4.

## 1.1 Contextualização

Sistemas autoadaptativos (*Self-Adaptive Systems - Self-Adaptive Systems (SAS)*) são caracterizados por realizar operações com o mínimo ou nenhuma interferência humana (BRUN *et al.*, 2009). Eles são capazes de adaptar seu comportamento e sua arquitetura em tempo de execução e ainda verificar se seu desempenho atual corresponde ao desempenho necessário para realizar operações (LADDAGA; ROBERTSON, 2004).

Sistemas autoadaptativos estão sendo cada vez mais utilizados em diversos domínios, tais como cidades inteligentes, telecomunicações e prevenção de catástrofes (ADJOYAN; SERIAI, 2017). Nesse contexto, percebe-se que os software se tornaram complexos e isso pode ser explicado pelo aumento no nível de heterogeneidade dos componentes de software, níveis mais altos de segurança, e mudanças mais frequentes no contexto, objetivos e requisitos em tempo de execução (SALEHIE; TAHVILDARI, 2009).

Além disso, SAS exigem evolução contínua das suas capacidades de adaptação e a garantia da qualidade das tarefas de reconfiguração em tempo de execução, fazendo com que seja necessário o monitoramento e a avaliação da qualidade das adaptações e do comportamento do sistema (RAIBULET *et al.*, 2016).

Nesse sentido, é importante que exista uma preocupação com a avaliação da qualidade, uma vez que, um sistema deve executar adaptações para alcançar uma configuração válida visando atender às necessidades do contexto atual. Sanchez *et al.* (2014) sugerem que a adaptação deve atender as regras de configuração, as restrições de recursos e as preferências dos *stakeholders*, especialmente no tocante aos atributos de qualidade. Por esse motivo, a melhor configuração do sistema é aquela que busca otimizar um conjunto de atributos de qualidade quantificados mediante medidas de qualidade.

Uma forma de dar suporte à garantia de qualidade é realizar avaliações de qualidade por meio de medições. As medições de software podem ser utilizadas para quantificar atributos específicos com o objetivo de verificar se o produto de software avaliado contempla os aspectos de qualidade desejados (ISO, 2011a). Em se tratando de sistemas autoadaptativos, atributos de qualidade se relacionam com propriedades *self\**, que determinam um conjunto de operações específicas que esses sistemas podem realizar. Por exemplo, a propriedade de autoconfiguração (*self-configuration*) impacta atributos de qualidade, tais como manutenibilidade, funcionalidade, portabilidade e usabilidade (SALEHIE; TAHVILDARI, 2009).

Entretanto, ainda existem lacunas quando se trata da avaliação de sistemas autoadaptativos. Em relação às abordagens baseadas em medição, pesquisadores têm apresentado soluções direcionadas a uma perspectiva ou um fator de qualidade específico (RAIBULET, 2014), ocasionando *gaps* em relação a outros aspectos. Nesse sentido, neste trabalho é realizado um estudo de Mapeamento Sistemático Mapeamento Sistemático (MS), que tem por objetivo identificar na literatura quais atributos e medidas de qualidade têm sido utilizados para avaliar os SAS.

A norma Organização Internacional de Normalização (ISO) 25010 (ISO, 2011a) apresenta um conjunto de terminologias para medidas e atributos de qualidade que podem ser utilizadas para medir vários níveis e estágios do processo de desenvolvimento de software, bem como para avaliar a qualidade do produto de software e tem se mostrado adequada para avaliar a qualidade de sistemas de softwares intensivos, como sistemas autoadaptativos (RAIBULET *et al.*, 2016).

Entre as características de qualidade disponibilizadas pela norma ISO (ISO, 2011a), uma importante a ser avaliada em sistemas autoadaptativos é a confiabilidade. Essa característica pode ter impacto em operações relacionadas a propriedades dos SAS, como autocura, autoproteção e autoconfiguração (SALEHIE; TAHVILDARI, 2009).

Além disso, SAS com baixa confiabilidade podem apresentar problemas no monitoramento do desempenho e na qualidade de seus serviços (NETI; MULLER, 2007). Portanto, devido ao impacto desta característica no funcionamento dos SAS, a avaliação da confiabilidade do SAS se faz essencial.

## 1.2 Motivação

A norma ISO/IEC 25000, também conhecida como *Software Product Quality Requirement and Evaluation* (SQuaRE) (ISO, 2011a), apresenta uma grande quantidade de conhecimentos sobre qualidade de software e é bastante utilizada para auxiliar em medições de software, oferecendo um grande volume de informações distribuído em modelos, conceitos, atributos e medidas de qualidade.

A norma ISO 25010, que é parte da norma SQuaRE, é adequada para avaliação de sistemas de software como sistemas autoadaptativos. Entretanto, o seu uso pode não contemplar todas as particularidades dos SAS, como suas propriedades dinâmicas (RAIBULET *et al.*, 2016) (NETI; MULLER, 2007). Tendo isso em vista, pesquisadores têm desenvolvido novas abordagens, características e medidas para melhor avaliar a qualidade dos SAS.

A maioria das abordagens disponíveis que utiliza medidas para realizar uma avaliação de qualidade em SAS considera apenas uma perspectiva (e.g., desempenho ou esforço de desenvolvimento), ou é específica para um domínio de aplicação (RAIBULET, 2014), o que deixa lacunas quanto à avaliação de outros aspectos dos SAS, como as reconfigurações dinâmicas e outras propriedades de autoadaptação (RAIBULET *et al.*, 2016).

O trabalho apresentado em (SOUSA *et al.*, 2019) descreve um conjunto de desafios enfrentados ao realizar avaliações de qualidade em SAS. Entre eles, pode-se citar a complexidade inerente dos SAS, uma grande variedade de arquiteturas e de propriedades *self\**, falta de ferramentas automatizadas, e a escassez de medidas de qualidade direcionadas a SAS.

Os desafios relacionados à avaliação da qualidade de sistemas autoadaptativos se estendem para a avaliação de características de qualidade específicas. Algumas características, como eficiência de desempenho, possuem uma grande quantidade de soluções e medidas de qualidade propostas, enquanto outras características, como a confiabilidade, têm suas sub-características pouco exploradas e poucas medidas bem definidas.

Dada a importância da confiabilidade para os sistemas autoadaptativos, possíveis soluções para ajudar a garantir que os SAS tenham mais confiabilidade podem incluir avaliações direcionadas apenas a essa característica por meio do uso de medidas de qualidade específicas, definição de novas medidas que atendam às especificidades desses sistemas, estudos sobre as sub-características de confiabilidade e seu impacto no funcionamento do SAS, mapeamento de possíveis conflitos entre a confiabilidade dos SAS com outras características, ou atributos de qualidade.

### 1.3 Objetivos e Metodologia

Com base no que foi apresentado nas seções anteriores e considerando a experiência do grupo de pesquisa<sup>1</sup> e o projeto<sup>2</sup> no qual esta dissertação está inserida, este trabalho tem o objetivo de propor um conjunto de medidas de qualidade específicas para dar suporte à avaliação da confiabilidade de sistemas autoadaptativos.

Para alcançar o objetivo geral deste trabalho foram definidos objetivos específicos listados a seguir:

- Investigar características, sub-características, atributos e medidas de qualidade usados na avaliação da qualidade de sistemas autoadaptativos reportados na literatura.
  - Identificar as características de qualidade mais importantes para sistemas autoadaptativos.
  - Selecionar uma característica de qualidade relevante para sistemas autoadaptativos.
- Definir medidas de qualidade relacionadas à característica de qualidade selecionada.
- Automatizar medidas de qualidade definidas para a característica de qualidade selecionada.
- Avaliar as medidas de qualidade definidas por meio da aplicação dessas medidas em sistemas autoadaptativos e analisar os resultados.

Para atingir os objetivos específicos definidos, o desenvolvimento deste trabalho se baseia em um conjunto de atividades aliadas a métodos científicos. A Figura 1 ilustra a associação entre os métodos científicos e as atividades realizadas para atingir o objetivo deste trabalho.

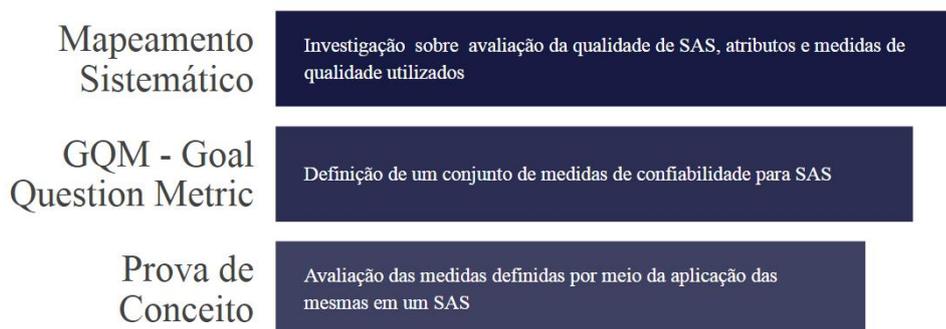


Figura 1 – Metodologia de Desenvolvimento da Dissertação.

<sup>1</sup> Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREat)

<sup>2</sup> ADApt - Abordagem para Desenvolvimento e Avaliação de Aplicações no Ambiente de Cidades Inteligentes, financiado pelo INES 2.0 ([www.ines.org.br](http://www.ines.org.br)), processo FACEPE APQ-0399-1.03/17, processo CAPES 88887.136410/2017-00, e processo CNPq 465614/2014-0.

Para atingir o primeiro objetivo específico, é realizada uma atividade de investigação na literatura apoiada pelo método de mapeamento sistemático, que é um estudo do tipo Revisão Sistemática e é caracterizado como um estudo secundário, pois o resultado primário é constituído por um conjunto de estudo primários, como experimentos, (KITCHENHAM; CHARTERS, 2007). Além disso, mapeamentos sistemáticos possibilitam que uma visão ampla sobre um determinado assunto seja obtida, além de facilitar a identificação de pesquisas futuras (PETERSEN *et al.*, 2008) (KITCHENHAM; CHARTERS, 2007). Essa metodologia foi selecionada para desenvolver este trabalho, pois permite a extração sistematizada de informações da literatura e a categorização dos dados extraídos, como características, sub-características, atributos e medidas de qualidade descritas primeiro objetivo específico.

Para atingir a meta que trata da definição de um conjunto de medidas direcionadas a sistemas autoadaptativos é adotado o método *Goal Question Metric* (GQM), que é uma abordagem sistemática que busca integrar objetivos a processos de software, produtos de software e perspectivas de qualidade de acordo com as necessidades de um determinado projeto ou organização e seu princípio mais importante é que as medições devem ser orientadas a objetivos (SOLINGEN; BERGHOUT, 1999). Além do método GQM, existe também o PSM que é um método também sistemático para planejar e executar medições e análises de software (Card, 2003). Entretanto, o GQM foi escolhido por ser bastante difundido quando se trata de derivação de medidas de software e pela experiência em sua utilização possuída pelo grupo de pesquisa onde este trabalho foi desenvolvido.

Em relação à meta de automatização das medidas, é desenvolvida uma biblioteca na linguagem Java, denominada CECILIA (biblioteca de medidas de Confiabilidade para Sistemas Autoadaptativos), com o propósito de facilitar a coleta das medidas para a sua posterior avaliação.

Para alcançar a meta de avaliação de medidas é adotado o método de prova de conceito, que busca verificar a viabilidade de determinado conceito, modelo, produto ou solução e analisar se o que está sendo avaliado traz benefícios (GLASS *et al.*, 2002). Existem outros métodos que poderiam ser utilizados para avaliar as medidas definidas, um deles é o estudo de caso. Entretanto, para realizar um estudo de caso seria necessária a aplicação da proposta de solução em um ambiente real, uma vez que são estudos que buscam entender determinado fenômeno em ambientes reais, analisando o impacto da implantação de uma nova solução (WOHLIN *et al.*, 2012). Dessa forma, o método de prova de conceito é escolhido, uma vez que

permite a verificação da possibilidade prática de soluções em pequenas escalas e os recursos necessários para a sua execução são viáveis para o contexto de desenvolvimento deste trabalho.

#### 1.4 Estrutura da Dissertação

Além do Capítulo 1, que apresenta a introdução do trabalho, esta dissertação está organizada em outros cinco Capítulos, a seguir:

- **Capítulo 2:** aborda a base teórica que fundamenta esta dissertação, são eles: Sistemas Autoadaptativos, Qualidade de Software e Avaliação da Qualidade em Sistemas Autoadaptativos.
- **Capítulo 3:** apresenta um estudo de mapeamento sistemático realizado para investigar a avaliação da qualidade de sistemas autoadaptativos e capturar os principais atributos de qualidade para sistemas autoadaptativos.
- **Capítulo 4:** discorre sobre a metodologia utilizada para definição de medidas de qualidade, bem como apresenta o conjunto de medidas definidas.
- **Capítulo 5:** apresenta os resultados obtidos por meio da utilização das medidas propostas, bem como a metodologia utilizada para avaliá-las.
- **Capítulo 6:** sumariza os resultados obtidos neste trabalho, apresenta os artigos publicados durante o programa de mestrado e aponta estudos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este Capítulo apresenta o embasamento teórico deste trabalho. A Seção 2.1 discorre sobre as definições de autoadaptação e sistemas autoadaptativos. A Seção 2.2 apresenta uma visão geral sobre a qualidade de software e as principais normas que a regem. A Seção 2.3 apresenta os conceitos da qualidade de software direcionados para a avaliação de sistemas autoadaptativos. Por fim, A Seção 2.4 apresenta considerações finais sobre este Capítulo.

### 2.1 Sistemas Autoadaptativos

Sistemas de software estão se tornando cada vez mais complexos, adquirindo propriedades que os tornam configuráveis no momento do seu funcionamento. Atualmente, os softwares são capazes de explorar contextos e propriedades dinâmicas para adaptar seu próprio comportamento quando determinado contexto é alterado. Esses sistemas apresentam capacidades que proporcionam um comportamento autônomo que funciona por meio de um controlador que gerencia e executa políticas e algoritmos com base em informações capturadas do ambiente (RAIBULET *et al.*, 2016).

(BRUN *et al.*, 2009) definem autoadaptação como a habilidade do software para ajustar seu próprio comportamento em resposta a mudanças no ambiente. Nesse sentido, o prefixo “auto” indica que o sistema tem autonomia para tomar decisões (i.e., com mínima ou nenhuma interferência) sobre como realizar a adaptação ou se reorganizar para acomodar as mudanças advindas de contextos e ambientes. Entretanto, a definição de adaptação tem sido utilizada de maneira genérica para descrever diversos tipos de sistemas, incorporando uma variedade de subdomínios, como, por exemplo, Linhas de Produto de Software Dinâmicas (LPSDs), em que a variabilidade é vinculada em momento de execução para melhorar a qualidade do serviço (SABATUCCI *et al.*, 2018) e Sistemas Ciber-Físicos, que são constituídos por ambientes físicos que geram informações sobre seus usuários e por dispositivos de comunicação distribuídos nesse ambiente (MAIA *et al.*, 2016).

Sistemas autoadaptativos podem ser definidos como softwares capazes de avaliar seu próprio comportamento e alterá-lo caso sua avaliação indique que é possível obter melhores funcionalidades ou desempenho, ou ainda que o software não está fazendo o que deveria fazer (LADDAGA; ROBERTSON, 2004). Dessa forma, esses sistemas são capazes de adaptar-se automaticamente em resposta as mudanças no ambiente em que estão inseridos, sem que ocorra

interação humana (SEN *et al.*, 2015).

Sistemas autoadaptativos possuem características que podem estar escondidas no projeto do sistema. Entretanto, existem características que sistemas autoadaptativos de diversos domínios têm em comum, que são: 1) tipicamente as decisões de projeto são parcialmente feitas em tempo de execução; e 2) os sistemas são capazes de raciocinar sobre o seu estado no ambiente. Esse raciocínio normalmente envolve quatro fases: coletar, analisar, decidir e agir (CHENG *et al.*, 2009a).

O *loop* de controle, ilustrado na Figura 2, é iniciado com a **coleta**. Nessa etapa do ciclo de controle, são capturados dados relevantes por meio de sensores do ambiente e outros recursos que refletem o estado atual do sistema. Em seguida, o sistema faz a **análise** os dados coletados. Para realizar essa etapa, existem diversas abordagem para estruturar os dados coletados e analisá-los (CHENG *et al.*, 2009a). O terceiro passo é a tomada de **decisão**, no qual o sistema deve determinar como se adaptar de modo a alcançar o estado desejado. Por fim, para implementar a decisão, o sistema realiza uma **ação** por meio de atuadores disponíveis (CHENG *et al.*, 2009a). Com base nessas operações, os SAS executam adaptações para atender as necessidades dos contextos em que estão operando.

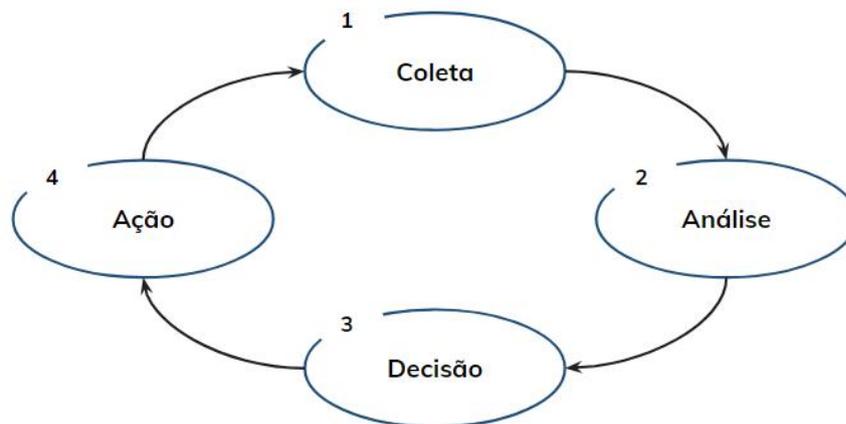


Figura 2 – *Loop* de controle. Adaptado de (CHENG *et al.*, 2009a)

Sistemas autoadaptativos são sistemas *self-\**, os quais apresentam um conjunto de propriedades específicas que indicam conjuntos de operações que esses sistemas são capazes de realizar. A Figura 3 ilustra essas propriedades em uma estrutura hierárquica em três níveis.

O nível geral ilustrado na Figura 3 contém as propriedades globais de softwares autoadaptativos. Um sistema com propriedades desse nível tipicamente possuem muitos elementos

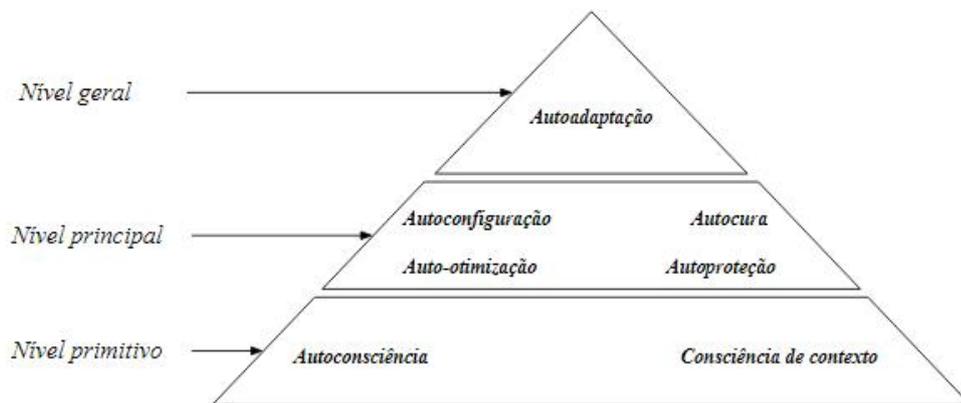


Figura 3 – Propriedades de sistemas *self*-. Adaptado de (SALEHIE; TAHVILDARI, 2009)

capazes de interação, mas que são total ou parcialmente inconscientes sobre o sistema como um todo.

O nível principal contém propriedades definidas para os mecanismos de autoadaptação. As propriedades desse nível são detalhadas a seguir:

- *Autoconfiguração* é a capacidade de se reconfigurar automaticamente e dinamicamente em resposta a mudanças, realizando instalações, atualizações, integrações e decompondo e compondo entidades de software.
- *Autocura* é a capacidade de descobrir, diagnosticar e reagir a rupturas. Essa propriedade também é capaz de antecipar potenciais problemas e agir para prevenir o sistema contra falhas. O autodiagnóstico e o autoreparo também estão relacionados a propriedade de autocura. O autodiagnóstico se refere à identificação de erros<sup>1</sup>, faltas<sup>2</sup>, e falhas<sup>3</sup>. O autoreparo se concentra em se recuperar de eventuais erros, faltas ou falhas.
- *Auto-otimização* é a capacidade de gerenciar o desempenho e a alocação de recursos do sistema para satisfazer os requisitos dos usuários. Tempo de resposta e taxa de transferência são exemplos de responsabilidades desta propriedade.
- *Autoproteção* é a capacidade de detectar brechas na segurança e se recuperar dos seus efeitos. Esta propriedade é responsável por defender o sistema contra ataques maliciosos, e antecipando problemas, agir para mitigar seus efeitos.

O nível primitivo é composto pelas propriedades de autoconsciência e consciência de contexto. As propriedades são descritas a seguir:

<sup>1</sup> Condição que pode advir de intervenção humana e que produz um resultado incorreto. São muitas vezes responsáveis por evidenciar defeitos. (IEEE, 1990)

<sup>2</sup> Um defeito em dispositivo ou componente, como um código implementado incorretamente, um cabo quebrado ou um curto circuito (IEEE, 1990)

<sup>3</sup> Incapacidade do sistema de realizar as funções requeridas com o desempenho especificado ou a apresentação de um resultado incorreto e inesperado (IEEE, 1990)

- *Autoconsciência* significa que o sistema tem conhecimento sobre seu próprio estado e comportamento.
- *Consciência de contexto* é a capacidade do sistema de estar consciente do ambiente em que está operando.

As propriedades de sistemas autoadaptativos possuem um relacionamento direto com características e atributos de qualidade. Por exemplo, confiabilidade e manutenibilidade são duas características de qualidade que se relacionam fortemente com a propriedade de autocura (*self-healing*) (NETI; MULLER, 2007). Nesse sentido, para cada uma das propriedades que um sistema autoadaptativo possui, é importante verificar se as características de qualidade relacionadas estão sendo atendidas.

Essas propriedades também relacionam com o conceito de variabilidade dinâmica. Esse conceito é normalmente usado quando se fala de Linhas de Produto de Software Dinâmicas, uma vez que essas são capazes realizar adaptação, customização e reconfiguração em tempo de execução, tornando a autoadaptação uma propriedade em comum com sistemas autoadaptativos. Além das LPSDs, outros tipos de sistemas também compartilham as propriedades de sistemas autoadaptativos, como aplicações IoT e sistemas ubíquos (CAPILLA *et al.*, 2014).

A variabilidade pode ser definida como a habilidade que um software possui de alterar-se ou customizar-se a partir da descoberta de um ponto de variação (GURP *et al.*, 2001). A variabilidade pode ser estática ou dinâmica, a qual também é chamada de variabilidade em tempo de execução (BENCOMO *et al.*, 2008). A variabilidade dinâmica é vista como a capacidade do software de se customizar em tempo de execução, isso se dá devido às variações do produto que surgem em um determinado contexto de execução (BENCOMO *et al.*, 2012).

Sistemas autônomos que necessitam ser reconfigurados em tempo de execução precisam de mecanismos de variabilidade em tempo de execução para lidar com os seguintes aspectos (CAPILLA *et al.*, 2016):

- Ativação e desativação de *features* do sistema após a implantação;
- Modificação estrutural da variabilidade de forma dinâmica;
- Adição e remoção de restrições de *features* em tempo de execução; e
- Obtenção de uma reconfiguração ótima das variantes do sistema.

Um sistema dinamicamente adaptativo desenvolvido com suporte de uma plataforma *middleware* pode ser conceitualizado como uma linha de produto, em que suas variabilidades são vinculadas em tempo de execução. Cada configuração baseada em componente pode ser

considerada como um produto ou variante do sistema dinamicamente adaptativo (BENCOMO *et al.*, 2008).

Um dos principais pontos a serem considerados quando se fala em variabilidade dinâmica, é como a configuração pode influenciar a capacidade dos mecanismos no momento de alterar e customizar as variantes. O impacto de transportar a configuração para o momento de execução pode ocasionar uma execução lenta e pouco eficiente, devido aos vínculos de variantes no momento de funcionamento do sistema (CETINA *et al.*, 2013).

Os impactos, que podem ser causados no sistema devido a operações de reconfiguração e adaptação, destacam a importância de garantir a qualidade do sistema quando operações dinâmicas são realizadas. Para garantir a qualidade desses sistemas, é necessário utilizar modelos de referência para conduzir avaliações de qualidade de software.

## **2.2 Qualidade de Software**

A qualidade de software é um conceito importante de engenharia de software e pode ser definida como grau em que um sistema é capaz de atender os requisitos que foram estabelecidos para ele (ISO, 2011a). Este trabalho é baseado em conceitos-chaves de qualidade de software que são discutidos nesta Seção. A subseção 2.2.1 apresenta o modelo de qualidade SQuARE com foco na qualidade do produto de software e na característica de confiabilidade que é o foco desta dissertação e a subseção 2.2.3 discorre sobre o modelo de medição proposto pela norma ISO/IEC 25010/2011 (ISO, 2011a) e os elementos que o compõe.

### **2.2.1 Modelo de Qualidade SQuARE**

Uma das normas mais importantes para a qualidade de software é a norma ISO/IEC 25010/2011 (ISO, 2011a). Ela foi criada como uma evolução das normas ISO/IEC 9126 e ISO/IEC 14598 que tratam da qualidade do produto de software. Em 1998, a Comissão de Estudos de Qualidade de Software da ABNT (Associação Brasileira de Normas Técnicas) percebeu que, para que o leitor tivesse uma boa compreensão sobre um determinado assunto ele poderia precisar consultar diversos documentos. Dessa forma, a comissão propôs um manual de utilização com o objetivo de auxiliar os leitores a compreenderem os conceitos das normas. A proposta do manual foi muito bem aceita, mas embora o conteúdo abordado pelas normas ISO/IEC 9126 e ISO/IEC 14598 fosse relevante, foi considerado que a forma de apresentar o

conteúdo merecia ser repensada (KOSCIANSKI; SOARES, 2007).

Em seguida, foi iniciado o projeto SQuaRE que significa *Software product Quality Requirement and Evaluation* (Requisitos de Qualidade e Avaliação de Produtos de Software). Nesse projeto, o material existente foi reorganizado, mas não foram realizadas mudanças radicais no material preexistente. A norma SQuaRE surgiu de uma maneira muito sólida pelos seguintes motivos (KOSCIANSKI; SOARES, 2007):

- Abrange amplamente o assunto qualidade do produto de software;
- Estabelece uma base precisa tanto para definir o modelo quanto para realizar a avaliação;
- Os documentos apresentam caráter didático;
- Representa um conjunto de experiências únicas sobre o assunto, uma vez que é resultado de esforços e consenso de centenas de pesquisadores; e
- Outros modelos de qualidade elaborados por pesquisadores de maneira pontual podem ser mapeados para o modelo SQuaRE.

A série de normas SQuaRE (ISO, 2011a) define em seu escopo um modelo de qualidade em uso e um modelo de qualidade do produto. O modelo de qualidade em uso é formado por cinco características, sendo algumas destas divididas em sub-características. As características do modelo de qualidade em uso estão relacionadas ao resultado de uma interação, que ocorre quando determinado produto é utilizado em um contexto de uso específico. O modelo de qualidade do produto é composto por oito características e todas possuem sub-características. As características do modelo de qualidade do produto estão associadas as propriedades estáticas e dinâmicas de um sistema computacional.

As características definidas pelos modelos de qualidade em uso e qualidade do produto são importantes para todo tipo de software. Os modelos apresentam uma terminologia apropriada para avaliar, especificar e medir a qualidade de produtos de software. Na Figura 4, é ilustrado o modelo de qualidade do produto.

O modelo apresentado na Figura 4 categoriza as propriedades de qualidade de um produto de software em oito características: adequação funcional, eficiência de desempenho, compatibilidade, usabilidade, confiabilidade, segurança, manutenibilidade e portabilidade. Cada característica possui um conjunto de sub-características relacionadas e podem ser entendidas como categorias de atributos de qualidade em que se baseia a qualidade de um software (ISO, 2011a). As características e sub-características de qualidade são descritas a seguir:

- **Adequação funcional:** grau em que um produto ou sistema fornece funções que cor-

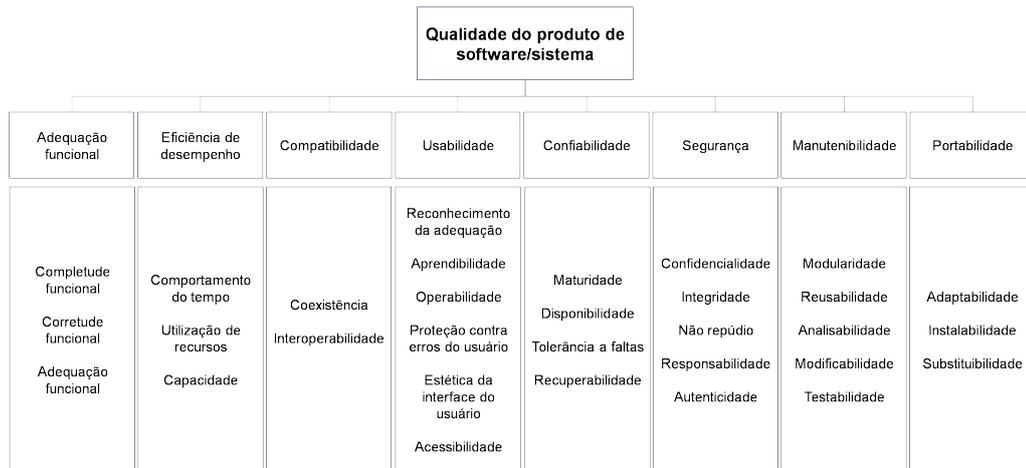


Figura 4 – Modelo de Qualidade do Produto. Adaptado de (ISO, 2011a).

respondam às necessidades explícitas e implícitas quando utilizado sob as condições especificadas.

- **Completeness funcional:** grau no qual um conjunto de funções abrange todas as tarefas especificadas e os objetivos do usuário.
- **Corretude funcional:** grau no qual um produto ou sistema fornece resultados corretos, com um determinado grau de precisão.
- **Funcionalidade apropriada:** grau em que as funções facilitam a realização das tarefas e dos objetivos para os quais o sistema foi especificado.
- **Confiabilidade:** grau em que um sistema, produto ou componente mantém, ao longo do tempo, um comportamento consistente com o esperado, sob as condições especificadas.
  - **Maturidade:** grau no qual um sistema, produto ou componente satisfaz às necessidades de confiabilidade em sua operação normal.
  - **Tolerância a falhas:** grau em que um sistema, produto ou componente opera como pretendido, apesar da presença de falhas de hardware ou software.
  - **Recuperabilidade:** grau em que, em caso de interrupção ou falha, um produto ou sistema pode recuperar os dados diretamente afetados e reestabelecer o estado desejado do sistema.
  - **Disponibilidade:** grau em que um sistema, produto ou componente está operacional e acessível quando requisitado para uso.
- **Usabilidade:** grau em que um produto ou sistema pode ser utilizado por usuários específicos para atingir metas especificadas com eficácia, eficiência e satisfação, em um determinado contexto.
  - **Conhecimento adequado:** grau em que os usuários podem reconhecer se um produto

ou sistema é apropriado para suas necessidades.

- **Apreensibilidade:** grau em que um produto ou sistema pode ser usado por usuários para alcançar objetivos específicos de aprendizagem para utilização do produto ou sistema com eficácia, eficiência, inexistência de risco e satisfação, em um contexto de uso especificado.
- **Operabilidade:** grau de facilidade com a qual um produto ou sistema é operado ou controlado.
- **Acessibilidade:** grau em que um produto ou sistema pode ser usado por pessoas com a mais ampla gama de características e capacidades, a fim de alcançar um objetivo especificado em um contexto de uso determinado.
- **Proteção de erro do usuário:** grau em que um sistema protege os usuários de cometer erros.
- **Estética da interface de usuário:** grau em que uma interface de usuário permite interação agradável e satisfatória.
- **Eficiência de desempenho:** desempenho do produto em relação à quantidade dos recursos utilizados sob condições estabelecidas.
  - **Comportamento no tempo:** grau em que os tempos de resposta e de processamento e taxas de transferência de um produto ou sistema, no desempenho das suas funções, atende aos requisitos.
  - **Utilização de recursos:** diz respeito à quantidade de recursos necessários para que um produto ou sistema atenda aos requisitos.
  - **Capacidade:** grau em que os limites máximos do produto ou sistema satisfazem os requisitos.
- **Manutenibilidade:** grau de eficácia e eficiência com que um produto ou sistema pode ser modificado pela equipe de manutenção.
  - **Analisabilidade:** grau de eficácia e eficiência com a qual é possível avaliar o impacto sobre um produto ou sistema de uma mudança em uma ou mais de suas partes.
  - **Modificabilidade:** grau em que um produto ou sistema pode ser modificado de forma eficiente e eficaz sem a introdução de defeitos ou sem degradação de sua qualidade.
  - **Modularidade:** grau em que um sistema ou programa de computador é composto por componentes discretos, de forma que uma mudança em um componente tem um impacto mínimo sobre os demais.

- Reusabilidade: grau em que um produto pode ser utilizado em mais do que um sistema, ou na construção de outros produtos.
- Testabilidade: grau de eficácia e eficiência com que critérios de teste podem ser estabelecidos e executados.
- **Portabilidade:** grau de eficácia e eficiência com a qual um sistema, produto ou componente pode ser transferido de um hardware, software ou ambientes de uso.
  - Adaptabilidade: grau em que um produto ou sistema pode eficazmente e eficientemente ser adaptado para um hardware, software ou ambientes de uso diferentes ou em evolução.
  - Instalabilidade: grau de eficácia e eficiência com que um produto ou sistema pode ser instalado e/ou desinstalado com sucesso em num ambiente especificado.
  - Substituibilidade: grau em que um produto pode substituir outro produto de software especificado para o mesmo fim no mesmo ambiente.
- **Segurança:** grau em que as funções e os dados, de um produto ou sistema, são protegidos do acesso não autorizado e o grau em que são disponibilizados para acesso autorizado.
  - Confidencialidade: grau em que um produto ou sistema garante que os dados são acessíveis somente por pessoas autorizadas.
  - Integridade: grau em que um sistema, produto ou componente evita o acesso não autorizado ou a modificação de programas de computador ou dados.
  - Não-repúdio: grau em que um produto ou sistema permite constatar que ações ou acessos foram efetivamente realizados, de forma que não possam ser negados posteriormente.
  - Responsabilização: grau em que as ações de uma entidade podem ser atribuídas exclusivamente a esta entidade.
  - Autenticidade: grau em que a identidade de uma entidade (pessoa ou recurso) pode ser comprovada a quem requisitar.
- **Compatibilidade:** grau em que um produto, sistema ou componente pode trocar informações com outros produtos, sistemas ou componentes, e/ou executar suas funções, enquanto compartilham o mesmo ambiente de hardware ou software.
  - Coexistência: grau em que um produto pode desempenhar as suas funções de forma eficiente ao compartilhar um ambiente e recursos comuns com outros produtos, sem impacto negativo em qualquer outro produto.

- Interoperabilidade: grau em que dois ou mais sistemas, produtos ou componentes podem trocar informações e utilizar as informações que foram trocadas.

Para cada sub-característica, a capacidade do software é determinada por um conjunto de propriedades estáticas que podem ser medidas. As características e sub-características podem ser medidas externamente por meio da extensão das capacidades providas pelo sistema que contém o software (ISO, 2011a).

O escopo do presente trabalho se limita a medições relacionadas à característica confiabilidade e suas sub-características com direcionamento para sistemas autoadaptativos.

### 2.2.2 *Característica de Confiabilidade*

A confiabilidade, conforme já descrito na subseção 2.2.1, determina o grau em que o sistema é capaz de executar funções específicas sob determinadas condições por um determinado período de tempo (ISO, 2011a), ou a habilidade em que o sistema se mantém operacional no decorrer do tempo (LOUKIL *et al.*, 2014).

Na literatura, a confiabilidade apresenta relações com diferentes aspectos de sistemas, por exemplo, no trabalho desenvolvido por Criado *et al.* (2016), a confiabilidade está relacionada a taxa de erro e custo de erro; já para Märtin e Nicolai (2014) está relacionada a tolerância a falhas. A avaliação da confiabilidade pode variar de acordo com os fatores ou atributos de qualidade que se deseja quantificar, e também pode ser avaliada em termos de sub-características (i.e., maturidade, disponibilidade, recuperabilidade e tolerância a falhas).

De acordo com a norma ISO (2011a), a confiabilidade tem influência em diversos aspectos de sistema do ponto de vista de diferentes *stakeholders*:

- Usuário primário: ator que interage diretamente com o sistema para realizar tarefas para as quais o sistema foi desenvolvido.
  - Impacto na interação: *Quão confiável um sistema precisa ser para que um usuário realize uma tarefa?*
- Usuário secundário: ator que fornece suporte ao sistema
  - Provedor de conteúdo (impacto na interação): *Quão confiável é atualizar o sistema com um novo conteúdo?*
  - Mantenedor (impacto na manutenção ou portabilidade): *Quão confiável é manter ou portar o sistema?*
- Usuário indireto: ator que faz uso das saídas do sistema

- Impacto nas informações de saída do sistema *Quão confiáveis são as saídas do sistema?*

Além dos impactos relacionados aos atores que utilizam o sistema, as necessidades gerais de confiabilidade podem depender de outros requisitos, como maturidade, disponibilidade, tolerância a falhas e recuperabilidade. Além disso, elas podem impactar outras propriedades do sistema, como a eficácia, o desempenho e a satisfação (ISO, 2011a).

### 2.2.3 *Medição de Software*

A medição de software é definida por (SOLINGEN; BERGHOUT, 1999) como um processo contínuo de definir, coletar e analisar dados do processo de desenvolvimento de software e seus produtos com o objetivo de alcançar melhor compreensão e controle dos processos e dos produtos, e fornecer informações significantes para melhorá-los.

As medidas de qualidade são medidas definidas de acordo com atributos de qualidade e o método utilizado para quantificá-los, podendo haver uso de uma função matemática ou não (ISO, 2011a). Essas medidas são utilizadas no processo de medição da qualidade de um software, com o objetivo de obter dados para caracterizar um produto em relação à sua qualidade com base em um modelo de referência. Uma vez que essas medições são realizadas, problemas e suas causas são identificados e o produto que está sendo medido é melhorado (PARK *et al.*, 1996).

Na Figura 5, é possível identificar os componentes que contribuem para a qualidade de um produto de software e como eles se relacionam. Os elementos de medição da qualidade, ilustrados na Figura 5, são resultados da aplicação de um método de medição. Um método de medição é um conjunto de procedimentos aplicados de forma lógica com o objetivo de quantificar propriedades de acordo com uma determinada escala. As características e sub-características podem ser quantificadas por meio da aplicação de funções de medição. O resultado das funções de medição são as medidas de qualidade, que se tornam quantificadores das características e sub-características de qualidade de um produto.

As medidas de software podem ser classificadas de acordo com a norma SQuaRE em medidas externas, internas e medidas de qualidade em uso. Os tipos de medidas são descritos a seguir:

- **Medidas externas** são as que fornecem uma visão de “caixa preta” do software e envolvem propriedades relacionadas a execução do software em um computador com a utilização de hardware e sistema operacional.

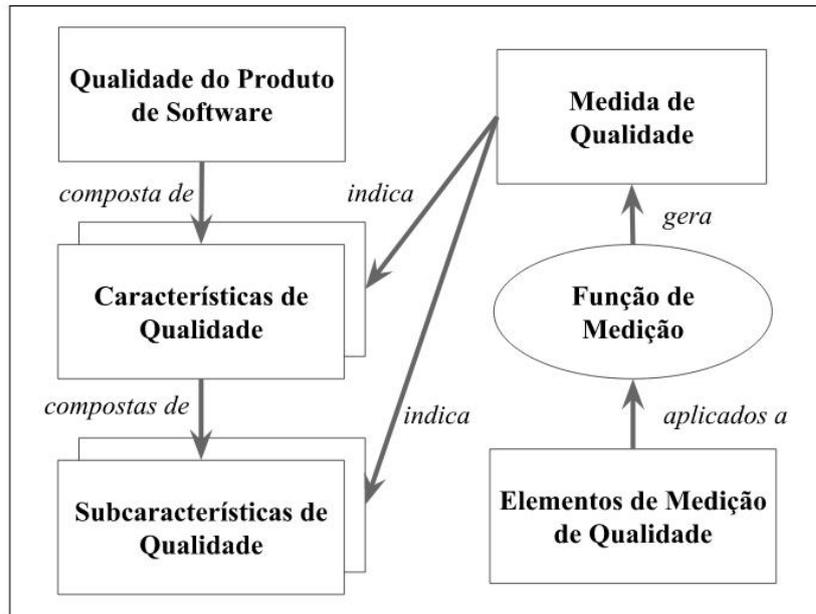


Figura 5 – Modelo de Referência para a Medição da Qualidade do Produto. Adaptado de (ISO, 2011a).

- **Medidas internas** são as que fornecem uma visão de “caixa branca” do software, envolvendo propriedades do produto que são tipicamente disponíveis para a avaliação do momento de desenvolvimento.
- **Medidas de qualidade em uso** são as medidas relacionadas as propriedades do sistema que podem incluir hardware, software, comunicação e usuários. Essas medidas tem impacto nos *stakeholders*.

#### 2.2.4 Método Goal Question Métric - GQM

O método GQM (*Goal Question Métric*) é um método sistemático cujo principal pilar do GQM são as medições orientadas a objetivo. Esse método permite a customização e integração de objetivos para modelos, processos de software, produtos e perspectivas de qualidade de acordo com as principais necessidade de um determinado projeto ou organização (SOLINGEN; BERGHOUT, 1999).

Para aplicação do GQM deve-se considerar três níveis de tarefas para a definição de medidas em uma perspectiva *top-down* e três níveis para a interpretação dos resultados em uma perspectiva *bottom-up* (SOLINGEN; BERGHOUT, 1999), conforme ilustra a Figura 6.

Na etapa de definição, a organização deve definir os objetivos de medição, esses objetivos são convertidos em questões, e as questões refinadas em métricas. Na etapa de interpretação, os resultados das métricas são analisados e é verificado se as questões foram

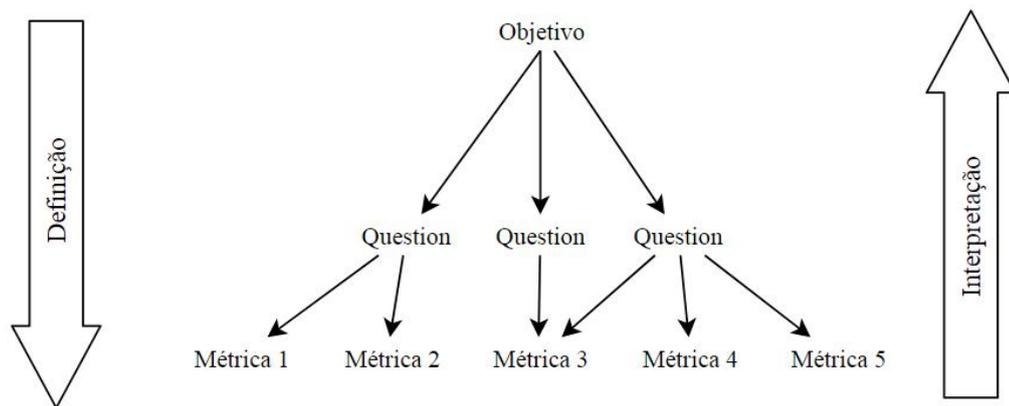


Figura 6 – Paradigma GQM. Adaptado de Solingen e Berghout (1999)

respondidas. Se as questões forem respondidas, então o objetivo foi atingido (SOLINGEN; BERGHOUT, 1999).

Além da estrutura de três níveis proposta pelo método, o GQM contém quatro fases:

1. *Fase de Planejamento*. Um projeto é selecionado, definido, caracterizado e planejado, resultando em um plano de projeto.
2. *Fase de Definição*. O programa de medição é definido (objetivos, questões, medidas e hipóteses são definidos) e documentado.
3. *Fase de Coleta de Dados*. Dados reais são coletados, resultando em um conjunto de dados.
4. *Fase de Interpretação*. Os dados coletados são analisados no que tange às métricas definidas nos resultados da medição, que fornecem respostas para as questões, e depois o alcance do objetivo pode ser avaliado.

O resultado da aplicação do GQM é a especificação de um programa de medição direcionado a um conjunto específico de problemas e um conjunto de regras de interpretação dos dados de medições. Com a aplicação desse método tem sido possível definir medidas sob duas perspectivas (SOLINGEN; BERGHOUT, 1999):

- Definição de medidas por membros do time de projeto, usando técnicas GQM; e
- Definição de medidas com base em modelos de processo de software e produtos.

Neste trabalho foi utilizada a primeira perspectiva, aliada a *templates* também providos pelo GQM, para definir medidas de confiabilidade para SAS.

### 2.3 Avaliação da Qualidade em Sistemas Autoadaptativos

Os sistemas autoadaptativos em sua evolução passaram a realizar avaliações e monitoramento contínuo, agregando mais complexidade, uma vez que são necessárias medidas específicas para medir a qualidade das adaptações e as atividades de tempo de execução. Além disso, é preciso avaliar em tempo de projeto o grau de adaptabilidade da arquitetura do sistema e suas propriedades de autoadaptação (RAIBULET *et al.*, 2016).

Em um determinado contexto de execução, um SAS possui diversas configurações válidas, mas apenas uma delas deverá ser escolhida para a adaptação. A seleção pela melhor configuração do sistema envolve a escolha da configuração candidata que é considerada mais apropriada dado um conjunto de atributos de qualidade quantificado por medidas. Essa operação envolve a consideração de aspectos como maximização da acurácia, obtenção do melhor desempenho do sistema e substituição da configuração atual por uma configuração mais simples (SANCHEZ *et al.*, 2014).

A avaliação dos mecanismos de adaptação pode depender de atributos de qualidade e de medidas de qualidade, podendo ser realizada em tempo de projeto e em tempo de execução (RAIBULET, 2014). A avaliação em tempo de projeto trata de soluções mais direcionadas a arquitetura que dão suporte a autoadaptação em um sistema de software. A avaliação das propriedades autoadaptativas em tempo de execução são mais relacionadas a medidas destinadas a prover uma avaliação qualitativa ou quantitativa do software (RAIBULET *et al.*, 2016).

De acordo com Raibulet (2014), os objetivos da avaliação de sistemas com propriedades *self*-\* podem ser:

- estabelecer um vocabulário unificado para a avaliação de sistemas *sel*-\*;
- prover mecanismos para avaliar os benefícios de um sistema *self*-\* em relação a um sistema tradicional;
- estabelecer bases para comparação entre dois sistemas *self*-\*, bem como para sua integração;
- estabelecer um grau de propriedades *self*-\* (e.g., esse sistema é mais ou menos adaptativo que outros?); e
- estimar o esforço para o desenvolvimento de propriedades *self*-\* em relação aos benefícios que elas proporcionam.

Em LPDSs, um dos domínios de SAS, as avaliações em tempo de projeto podem ser realizadas utilizando o principal artefato do domínio, que são os modelos de *features*,

responsáveis pode representar a variabilidade da LPSD (BEZERRA *et al.*, 2016). Os modelos de *features* podem ser avaliados com base em medições do ponto de vista de características como tamanho, flexibilidade e dinamicidade (BEZERRA *et al.*, 2018).

Salehie e Tahvildari (2009) relatam a existência de um relacionamento entre as propriedades dos SAS e fatores de qualidade. Por exemplo, autoconfiguração tem impacto sobre vários fatores de qualidade, tais como manutenibilidade, funcionalidade, portabilidade e usabilidade. Já a auto-otimização está relacionada com eficiência e a autoproteção se conecta fortemente com a confiabilidade, podendo também estar ligada a funcionalidade.

O principal objetivo dos sistemas autocuráveis, considerando a preocupação com as rupturas que podem ocorrer no sistema, é maximizar a capacidade de sobrevivência (*survivability*), manutenibilidade e confiabilidade (GANEK; CORBI, 2003). A confiabilidade, em sistemas autocuráveis (*self-healing systems*), pode ser dividida em tolerância a faltas e robustez, e a manutenibilidade pode ser dividida em modificabilidade e extensibilidade (NETI; MULLER, 2007).

As propriedades mais primitivas, como, consciência de contexto e autoconsciência, podem impactar os fatores, tais como, manutenibilidade, funcionalidade e portabilidade (SALEHIE; TAHVILDARI, 2009). Este trabalho se concentra na avaliação da confiabilidade para a sistemas autoadaptativos, considerando suas propriedades estáticas e dinâmicas.

## 2.4 Considerações Finais

Os conceitos apresentados neste Capítulo compõem o arcabouço de conhecimento necessário para o desenvolvimento deste trabalho, que tem por objetivo definir um conjunto de medidas de confiabilidade para auxiliar a avaliação de sistemas autoadaptativos.

No que concerne aos Sistemas Autoadaptativos, foram apresentadas definições, características e propriedades dos mesmos, afim de destacar suas especificidades, domínios que podem ser aplicados e suas capacidades dinâmicas. Diferentemente dos sistemas tradicionais, SAS podem realizar diversas operações sem interferência humana e avaliar o seu próprio comportamento.

Neste Capítulo também foram apresentados conceitos de qualidade de software, principalmente sobre a qualidade do produto de software. Foram abordadas definições de características, sub-características, atributos e medidas de qualidade, bem como foram apresentadas definições relacionadas aos processos de medição e definição de medidas de software.

Por fim, foram explorados conceitos sobre a avaliação de SAS que são imprescindíveis para a execução deste trabalho, como os momentos em que as avaliações podem ser realizadas, como elas podem ser realizadas e que propriedades dos sistemas devem ser consideradas.

Para aprofundar o conhecimento nesse tema e investigar como os SAS têm sido avaliados, foi realizado um estudo de mapeamento sistemático, apresentado no Capítulo seguinte.

### **3 QUALIDADE EM SAS: MAPEAMENTO SISTEMÁTICO**

Neste Capítulo, são apresentadas as etapas do mapeamento sistemático realizado com o objetivo de explorar o estado da arte da avaliação da qualidade de sistemas autoadaptativos com foco especial em características, sub-características e atributos de qualidade relacionados aos aspectos dinâmicos dos SAS. Como resultado desse mapeamento sistemático, foi obtido um conjunto de características, sub-características, atributos e medidas de qualidade pertinentes para SAS.

Este Capítulo está organizado em 5 seções. A Seção 3.1 apresenta a metodologia adotada para o desenvolvimento deste estudo. Na Seção 3.2, são apresentados os passos definidos para a execução do estudo. A Seção 3.3 descreve como o estudo foi conduzido. A Seção 3.4 reporta os resultados obtidos no estudo e apresenta as respostas às questões de pesquisa. A Seção 3.5 apresenta uma discussão sobre os resultados do mapeamento sistemático. Por fim, a Seção 3.6 apresenta as considerações finais sobre o mapeamento sistemático.

#### **3.1 Metodologia do Mapeamento Sistemático**

Mapeamentos sistemáticos podem ser realizados com o objetivo de obter um panorama geral sobre determinada área ou tópico de forma sistematizada, permitindo identificar, analisar e interpretar dados de estudos correspondentes a questões de pesquisas específicas (KITCHENHAM; CHARTERS, 2007).

Mapeamentos sistemáticos são compostos por três fases: Planejamento, Execução e Análise de Resultados. O planejamento trata de especificar todos os elementos que serão importantes para a condução do estudo, como as questões de pesquisa, os critérios de seleção e as bases de dados. A fase de execução se concentra em buscar e selecionar estudos relacionados ao tema estudado, bem como extrair dados dos estudos selecionados. A fase de análise de resultados tem o objetivo de sumarizar os resultados obtidos (KITCHENHAM; CHARTERS, 2007).

Para realizar o MS apresentado neste Capítulo, foi adotado o processo proposto por Kitchenham e Charters (2007), conforme ilustrado na Figura 7. Essa metodologia foi escolhida por permitir que seja feito um levantamento de um conjunto de estudos de forma sistemática, bem como a extração e classificação de dados desses estudos.

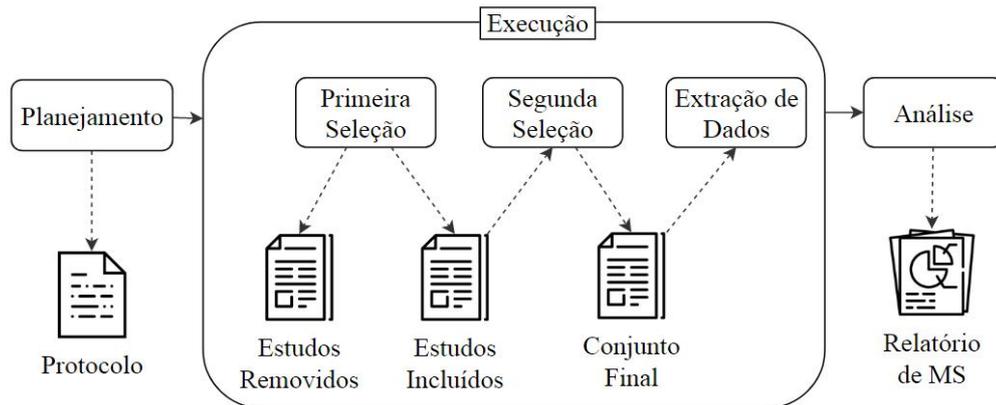


Figura 7 – Processo de mapeamento sistemático adotado. Adaptado de Kitchenham e Charters (2007)

### 3.2 Planejamento

Na fase de planejamento, os objetivos de pesquisa foram definidos e o protocolo do mapeamento sistemático foi produzido. No protocolo de mapeamento sistemático foram definidos: objetivos e questões de pesquisa; estratégia de busca; critérios de seleção; procedimentos para a seleção de estudos; e extração e análise de dados.

O objetivo do mapeamento sistemático é fornecer uma visão do estado da arte em relação a avaliação da qualidade de sistemas autoadaptativos. Nesse sentido foram estabelecidas as seguintes questões de pesquisa:

- **QP1.** Quais atributos de qualidade têm sido usados para avaliar sistemas autoadaptativos?
- **QP2.** Quais medidas de qualidade têm sido usadas para avaliar sistemas autoadaptativos?
- **QP3.** Quais soluções têm sido propostas para avaliar a qualidade de sistemas autoadaptativos?

Para responder às questões de pesquisa, foram considerados três tipos de estudos:

- Soluções que usam características, sub-características, atributos de qualidade ou medidas de qualidade para avaliar reconfigurações dinâmicas.
- Soluções que realizam avaliações de qualidade em sistemas autoadaptativos usando características, sub-características, atributos ou medidas de qualidade.
- Soluções que usam características, sub-características, atributos ou medidas de qualidade para avaliar a variabilidade dinâmica em sistemas autoadaptativos.

Para estabelecer a estratégia de busca, e dessa forma, responder às questões de pesquisa, foi definida a *População* (i.e., sistemas autoadaptativos) e a *Intervenção* (i.e., qualidade de sistemas autoadaptativos) do Mapeamento Sistemático de acordo com a estratégia *PICO* (PAI

*et al.*, 2004).

Consequentemente, foram identificadas quatro palavras-chave: “Sistemas Autoadaptativos”, “Características de Qualidade”, “Atributos de Qualidade” e “Medidas de Qualidade”. Foram identificados sinônimos e outros termos relacionados a essas palavras-chave e foram considerados os plurais de todas as palavras-chave.

Foram utilizados os operadores booleanos *OR* e *AND* para construir uma *string* de busca contendo as palavras-chave identificadas, assim como termos relacionados e seus sinônimos. Optou-se por utilizar apenas a língua inglesa, devido à maior difusão nos veículos de publicação. A *string* construída é ilustrada a seguir.

(“dynamic variability” OR “runtime variability” OR “reconfigura\*” OR “context-aware\*” OR “self-adaptive\*”) AND (“quality characteristic” OR “quality attribute” OR “non-functional requirement\*”) AND (“metric” OR “measure\*”)

Responder às questões de pesquisa levantadas envolve também selecionar as fontes de informações apropriadas para aplicar a estratégia de busca. No presente trabalho, optou-se por fazer um mapeamento exaustivo, com o objetivo obter uma grande quantidade de trabalhos e, por esse motivo, inicialmente foram selecionadas cinco bases de dados. Visando selecionar as bases mais adequadas para esta pesquisa, foram consideradas as bases *ACM Library*, *Scopus*, *ScienceDirect*, *IEEE Xplore* e *Springer*, pois estas são fontes importantes de informações relacionadas ao tema desta pesquisa (Dyba *et al.*, 2007), (KITCHENHAM; CHARTERS, 2007).

Nessa fase, também foram definidos os critérios de seleção dos artigos que foram utilizados para avaliar os estudos selecionados em cada base de dados. O uso desses critérios permitiu incluir estudos importantes para responder às questões de pesquisa, assim como permitiu excluir estudos irrelevantes. Os critérios de inclusão (INC) e exclusão (EXC) estabelecidos foram:

- **INC<sub>1</sub>**: O resumo do trabalho menciona explicitamente atributos ou medidas de qualidade no contexto de sistemas autoadaptativos.
- **INC<sub>2</sub>**: Artigos que descrevem abordagens que usam atributos de qualidade ou medidas relacionadas a reconfigurações dinâmicas.
- **INC<sub>3</sub>**: Capítulos de livro relacionados à qualidade de reconfigurações dinâmicas de sistemas autoadaptativos.
- **INC<sub>4</sub>**: Artigos publicados no período de 1995 a 2018.

Com a definição dos critérios de inclusão, a intenção foi capturar da literatura trabalhos (artigos ou capítulos de livro) que propusessem alguma solução relacionada a avaliação da qualidade de sistemas autoadaptativos. O período de tempo de 1995 a 2018 teve intenção de agregar uma maior quantidade de estudos e verificar tendências de pesquisa ao longo dos anos.

- **EXC<sub>1</sub>**: O artigo está fora do escopo de qualidade de sistemas autoadaptativos. Atributos e medidas de qualidade não são parte da contribuição do artigo.
- **EXC<sub>2</sub>**: Artigos que não estão escritos em inglês.
- **EXC<sub>3</sub>**: Artigos que possuem menos de 3 páginas.
- **EXC<sub>4</sub>**: Teses de doutorado ou dissertações de mestrado.
- **EXC<sub>5</sub>**: Estudos secundários.

Os critérios de exclusão definidos tiveram a finalidade de remover da seleção de estudos, os trabalhos que não apresentam atributos e medidas de qualidade e fora do escopo do tema do presente trabalho. Além disso, a intenção do MS também foi identificar estudos com propostas de solução mais maduras, o que levou a definição dos critérios de exclusão 4, 5 e 6.

A *String* de busca foi construída e aplicada nas bases de dados selecionadas. Foram selecionados estudos no período de 1995 à fevereiro de 2018, nas áreas de engenharia e computação. Como resultado, foi obtido um conjunto de estudos possivelmente relacionados ao tópico de pesquisa. Nesse momento, foram lidos os títulos, resumos e palavras-chave e os critérios de inclusão e exclusão foram aplicados. Como resultado, um conjunto de estudos primários potencialmente relevantes foi definido.

Posteriormente, cada estudo primário selecionado foi lido por completo e analisado novamente quando aos critérios de inclusão e exclusão. Se a decisão sobre a inclusão ou exclusão de determinado estudo não estava clara, o estudo era analisado por duas revisoras especialistas. Em caso de discordâncias, discussões eram conduzidas.

Os pesquisadores participantes do mapeamento sistemático foram: uma estudante de mestrado<sup>1</sup> com 2 anos de experiência em qualidade de software e LPDSs no âmbito acadêmico; e uma professora doutora, com 15 anos de experiência em qualidade de software e 9 anos de experiência em LPDSs.

Cada estudo primário selecionado foi submetido a uma extração de dados, que ocorreu por meio da aplicação de um formulário ilustrado na Tabela 1. Esse formulário contém campos para preenchimento de dados relacionados a cada questão de pesquisa.

---

<sup>1</sup> Autora do presente trabalho

Durante a atividade de extração, os dados de cada estudo primário foram capturados pela pesquisadora autora deste trabalho. Em caso de dúvida, discussões com uma segunda pesquisadora eram conduzidas. Para responder às questões de pesquisa, foi realizada uma análise qualitativa, mapeando o formulário de extração de acordo com as questões.

Tabela 1 – Formulário de Extração.

Questão de Pesquisa	Item de Extração
QP1. Quais atributos de qualidade têm sido usados para avaliar sistemas autoadaptativos?	Características, sub-características e atributos de qualidade para sistemas autoadaptativos (e.g., acurácia, capacidade, confiabilidade).
QP2. Quais medidas de qualidade têm sido usadas para avaliar sistemas autoadaptativos?	Medidas de qualidade para sistemas autoadaptativos (e.g., tempo de resposta, frequência de falhas).
QP3. Quais soluções têm sido propostas para avaliar a qualidade de sistemas autoadaptativos?	Soluções que avaliam a qualidade de sistemas autoadaptativos (e.g., modelos, técnicas, procedimentos). Domínios que usam sistemas autoadaptativos (e.g., ambiente inteligentes, sistemas espaciais, dispositivos móveis). Mecanismos de controle usados em sistemas autoadaptativos (e.g., MAPE, MAPE-K, MIAC, MRAC).

### 3.3 Execução

O MS foi iniciado em Fevereiro de 2018 e terminou em Novembro de 2018. Durante a fase de execução, foram realizadas duas seleções. Na primeira seleção, foi utilizada a *string* de busca previamente adaptada para cada base de dados. No processo da primeira seleção, a base de dados *ACM Library* foi removida do conjunto das fontes de dados, uma vez que a mesma retornou apenas dois artigos. Assim, foram utilizadas apenas quatro bases de dados (*Scopus*, *SpringerLink*, *IEEE Xplorer* e *ScienceDirect*). A Tabela 2 ilustra o número de estudos obtidos em cada base de dados.

Tabela 2 – Estudos por base de dados

Base de Dados	Artigos Retornados	Artigos Incluídos
Scopus	415	9
IEEE Xplorer	169	10
ScienceDirect	412	10
Springer	481	7

No total foram obtidos 1477 documentos. Nesse conjunto alguns documentos não eram artigos científicos (e.g., glossários, índices e capítulos de introdução de livros de outros temas) e, portanto, foram removidos na primeira seleção. Em seguida, foram removidos estudos duplicados (i.e., 78 estudos), o que resultou em um conjunto primário de 1368 artigos.

Para apoiar o gerenciamento do MS foi utilizada a ferramenta StArt<sup>2</sup>, que permite manter informações sobre os artigos (e.g., título, autores e resumo), como também permite adicionar critérios de seleção e manter os dados do protocolo de MS. Como resultado da primeira seleção, realizada de acordo com os critérios descritos na seção 3.2, foi obtido um conjunto de 39 artigos para uma inspeção mais detalhada.

A exclusão dos documentos obtidos no conjunto primário se deu por meio da utilização dos critérios apresentados na seção 3.2, e quantidade de documentos removidos por cada critério é listada abaixo:

- **EXC<sub>1</sub>**. Artigos fora do escopo de qualidade de sistemas autoadaptativos. Atributos e medidas de qualidade não são parte da contribuição do artigo: *1322 documentos*
- **EXC<sub>2</sub>**. Artigos que não estão escritos em inglês: *0 documentos*
- **EXC<sub>3</sub>**. Artigos que possuem menos de 3 páginas: *13 documentos*
- **EXC<sub>4</sub>**. Teses de doutorado ou dissertações de mestrado: *2 documentos*
- **EXC<sub>5</sub>**. Estudos secundários: *8 documentos*
- **Não eram documentos válidos**: *31 documentos*

Na segunda seleção, o conjunto de 39 artigos selecionados foi inteiramente lido e os critérios de inclusão e exclusão foram aplicados novamente. Como resultado, 36 estudos foram incluídos e 3 estudos foram excluídos, conforme ilustra a Figura 8.

Foi observado que o número de estudos foi reduzido em uma grande proporção, isto se deu pelo fato de que foram obtidos muitos estudos relacionados ao tema, mas a grande maioria não trazia em sua contribuição atributos e/ou medidas de qualidade relacionadas a SAS.

### 3.4 Análise

Na fase de análise, os resultados foram agrupados por questões de pesquisa, e foi realizada uma inspeção dos artigos com o objetivo de obter informações (e.g., atributos e medidas de qualidade e tipos de soluções propostas) sobre o estado da arte da avaliação da qualidade em SAS. O conjunto final de artigos é apresentado na Tabela 3, por limitação de espaço os nomes das bases foram abreviados: Scopus (S), SpringerLink (SL) IEEEXplorer (IEEE) e ScienceDirect (SD).

Entre os estudos selecionados, foi verificada a diversidade de domínios, tais como *Internet of Things* (IoT), LPDSs e Sistemas Espaciais, o que indica a adoção de SAS em cenários

<sup>2</sup> [http://lapes.dc.ufscar.br/tools/start\\_tool](http://lapes.dc.ufscar.br/tools/start_tool)

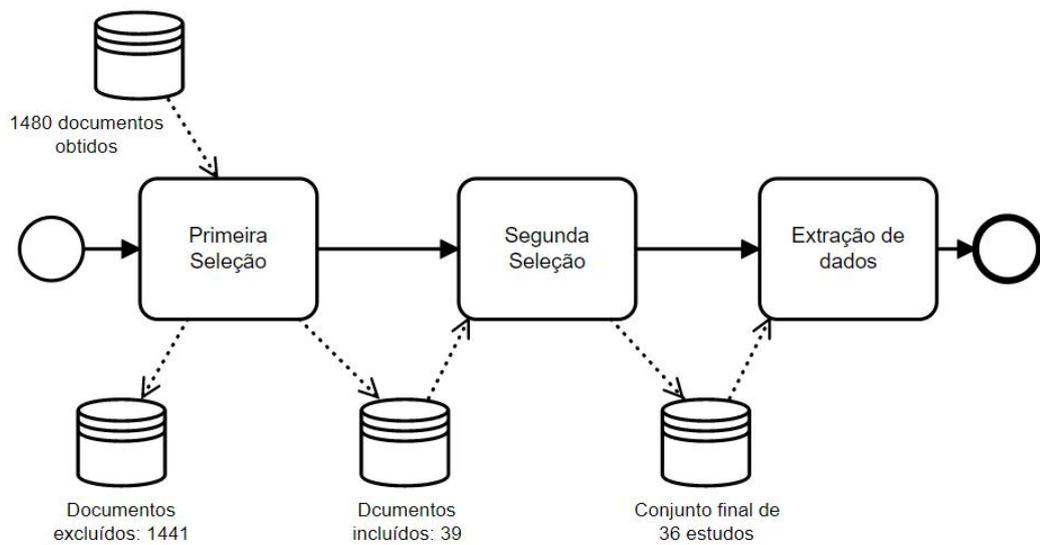


Figura 8 – Processo de Seleção do Mapeamento Sistemático

variados. Os resultados são apresentados nas próximas seções.

Neste ponto do estudo, foram detectados trabalhos relacionados a esta dissertação de mestrado, seja em relação à sua contribuição, metodologia, objetivos ou domínio da pesquisa. Os trabalhos relacionados são descritos a seguir.

O trabalho proposto por Kim (2016) apresenta um modelo de qualidade para aplicações do paradigma IoT (*Internet of Things*). O modelo tem o objetivo de auxiliar na avaliação da qualidade de aplicações IoT, levando em consideração as características particulares das aplicações IoT, tais como, conexão com diversos dispositivos de hardware e monitoramento de serviços inteligentes. De forma semelhante ao presente trabalho, o estudo realizado em Kim (2016) se preocupa com características dinâmicas de aplicações mais complexas, no caso IoT, mas não considera adaptações em tempo de execução e tem pouco foco em medidas de qualidade.

Carvalho *et al.* (2018) propõem um conjunto de 24 medidas com o objetivo de avaliar aplicações ubíquas com foco na interação. As medidas foram validadas de maneira teórica e empírica, e com o auxílio de duas aplicações ubíquas em dispositivos móveis. Durante a validação, as medidas propostas foram capazes de identificar problemas com respeito a características de qualidade específicas de sistemas ubíquos. O trabalho realizado por Carvalho *et al.* (2018) está relacionado com a presente dissertação de mestrado, tanto quanto a metodologia e o foco na definição de medidas de qualidade para sistemas dinâmicos. Entretanto, esse trabalho se concentra na característica de usabilidade do sistema.

Bezerra *et al.* (2016) apresentam uma ferramenta que oferece suporte à extração automática de medidas de qualidade a partir de um Modelo de *Features* (MF), que são represen-

Tabela 3 – Conjunto Final de Estudos Seleccionados

ID	Título	Referência	Base
E1	An evaluation method for Self-Adaptive systems	(FARAHANI <i>et al.</i> , 2017)	S
E2	The reassessment of preferences of non-functional requirements for better informed decision-making in self-adaptation	(PAUCAR; BEN-COMO, 2017)	S
E3	DyMMer: A measurement-based tool to support quality evaluation of DSPL feature models	(BEZERRA <i>et al.</i> , 2016)	S
E4	A tool for monitoring and maintaining system trustworthiness at runtime	(GOLDSTEEN <i>et al.</i> , 2015)	S
E5	Software architecture adaptability metrics for QoS-based self-adaptation	(PEREZ-PALACIN <i>et al.</i> , 2011)	S
E6	A framework for evaluating quality-driven self-adaptive software systems	(VILLEGAS <i>et al.</i> , 2011)	S
E7	Towards runtime optimization of software quality based on feedback control theory	(CHEN <i>et al.</i> , 2009)	S
E8	Metrics evaluation for context-aware computing	(MANTORO, 2009)	S
E9	Infrastructure for ubiquitous computing: Improving quality with modularisation	(MUNNELLY; CLARKE, 2008)	S
E10	An Overview on Quality Evaluation of Self-Adaptive Systems	(RAIBULET <i>et al.</i> , 2017)	SD
E11	On the relationships between QoS and software adaptability at the architectural level	(PEREZ-PALACIN <i>et al.</i> , 2014)	SD
E12	Improving self-adaptation planning through software architecture-based stochastic modeling	(FRANCO <i>et al.</i> , 2016)	SD
E13	A Model Driven Approach to the Analysis of Quality Scenarios within Self-Adaptable SOA Systems	(PEREZ; CORREAL, 2011)	SD
E14	Building reliable and maintainable Dynamic Software Product Lines: An investigation in the Body Sensor Network domain	(PESSOA <i>et al.</i> , 2017)	SD
E15	An approach to model dependability of cyber-physical systems	(SANISLAV <i>et al.</i> , 2016)	SD
E16	Optimization of adaptation plans for a service-oriented architecture with cost, reliability, availability and performance tradeoff	(POTENA, 2013)	SD
E17	An Approach to use MDE in Dynamically Reconfigurable Networked Embedded SOAs	(MARCOS <i>et al.</i> , 2011)	SD
E18	Prototyping Dynamic Software Product Lines to evaluate run-time reconfigurations	(CETINA <i>et al.</i> , 2013)	SD
E19	Quality attribute tradeoff through adaptive architectures at runtime	(YANG <i>et al.</i> , 2009)	SD
E20	Quality attribute scenario based architectural modeling for self-adaptation supported by architecture-based reflective middleware	(ZHU <i>et al.</i> , 2004)	IEEE
E21	Self-Adaptation of Service Based Systems Based on Cost/Quality Attributes Tradeoffs	(MIRANDOLA; POTENA, 2010)	IEEE
E22	Evaluation of dynamic adaptivity through metrics: an achievable target?	(RAIBULET; MASCI-ADRI, 2009a)	IEEE
E23	An Approach for Managing Quality Attributes at Runtime Using Feature Models	(SANCHEZ <i>et al.</i> , 2014)	IEEE
E24	Human-Intention Driven Self Adaptive Software Evolvability in Distributed Service Environments	(MING <i>et al.</i> , 2008)	IEEE
E25	Towards Evaluation Mechanisms for Runtime Adaptivity: From Case Studies to Metrics	(RAIBULET; MASCI-ADRI, 2009b)	IEEE
E26	Evaluating the effectiveness of the Rainbow self-adaptive system	(CHENG <i>et al.</i> , 2009b)	IEEE
E27	Towards Automated Context-Aware Software Quality Management	(GRAMBOW; OBERHAUSER, 2010)	IEEE
E28	Towards self-reconfiguration of space systems on architectural level based on qualitative ratings	(MÄRTIN; NICOLAI, 2014)	IEEE
E29	Software, performance and resource utilisation metrics for context-aware mobile applications	(RYAN; ROSSI, 2005)	IEEE
E30	Monitoring of Quality of Service in Dynamically Adaptive Systems	(LOUKIL <i>et al.</i> , 2014)	SL
E31	Exploring Quality-Aware Architectural Transformations at Run-Time: The ENIA Case	(CRIADO <i>et al.</i> , 2016)	SL
E32	Model-Driven Assessment of QoS-Aware Self-Adaptation	(GRASSI <i>et al.</i> , 2009)	SL
E33	Performance Aware Reconfiguration of Software Systems	(MARZOLLA; MIRANDOLA, 2010)	SL
E34	Dealing with Non-Functional Requirements for Adaptive Systems via Dynamic Software Product Lines	(GHEZZI; SHARIFLOO, 2013)	SL
E35	Personalized adaptation in pervasive systems via non-functional requirements	(SERRAL <i>et al.</i> , 2018)	SL
E36	An approach based on feature models and quality criteria for adapting component-based systems	(SANCHEZ <i>et al.</i> , 2015)	SL

tações da variabilidade de sistemas que pertencem a uma família de produtos, como as Linhas de Produto de Software Dinâmicas. Os autores discutem sobre a importância de avaliar qualidade do MF, uma vez que um modelo de qualidade ajuda a garantir que erros não irão se espalhar por toda a linha.

As Linhas de Produto de Software Dinâmicas compõem o arcabouço dos mais variados tipos de sistemas autoadaptativos, portanto, o trabalho desenvolvido por Bezerra *et al.* (2016) se relaciona com o domínio da pesquisa desenvolvida no presente trabalho. Um outro ponto em comum entre os trabalhos é o foco em medidas de qualidade para uma característica específica, no caso, a característica de manutenibilidade.

### **3.4.1 QP1. Quais atributos de qualidade tem sido utilizados para avaliar sistemas autoadaptativos?**

Essa questão de pesquisa busca identificar quais características, sub-características e atributos de qualidade tem sido utilizados para avaliar SAS. Foi observado que a maioria dos artigos não faz uso dos termos apresentados pela norma ISO (ISO, 2011a). Ao invés disso, os autores se referem a características, sub-características e atributos de qualidade por meio de termos genéricos, como fatores de qualidade, critérios de qualidade, objetivos de qualidade e requisitos não funcionais.

Nesse sentido, após a extração de dados, foram identificadas características, sub-características e atributos de qualidade apresentados na norma ISO (ISO, 2011a). Em seguida, as características, sub-características e atributos de qualidade foram analisados e agrupados de acordo com o modelo apresentado pela norma ISO.

Foram identificadas no total, 14 características de qualidade distintas, 24 sub-características e 100 atributos de qualidade diferentes, conforme ilustra a Tabela 4.

A Tabela 4 permite observar que algumas características extraídas da literatura não são apresentadas na norma ISO 25010. Isso pode indicar a necessidade de adaptar características existentes ou até criar novas características para dar suporte a avaliação de SAS.

Por outro lado, as características mais recorrentes e as com os maiores números de atributos encontrados na literatura são aquelas providas pela norma ISO 25010. A partir disso, pode-se inferir que a maioria dos pesquisadores busca suporte nos modelos oferecidos pela ISO, o que demonstra a importância da utilização dessa norma mesmo em sistemas dinâmicos.

Essa questão de pesquisa, também teve o propósito de identificar quais atributos

Tabela 4 – Número de sub-características e atributos de qualidade por característica

Característica	Número de Sub-características	Número de Atributos
Eficiência de Desempenho	3	28
Confiabilidade	8	9
Manutenibilidade	5	12
Segurança	3	1
Dependabilidade	0	6
Usabilidade	2	1
Funcionalidade	1	0
Portabilidade	3	1
Adaptação em tempo de execução	0	3
Confiança	0	0
Eficiência	2	0
Gerenciabilidade	0	2
Proteção	0	2
Escalabilidade	0	3
Sem Característica	0	56

foram mais avaliados ou validados com a intenção de verificar a sua relevância na literatura.

Dessa forma, os atributos foram classificados em duas categorias:

- Atributo Mencionado: apenas mencionado no artigo, ou utilizado para construir uma linha de raciocínio.
- Atributo Avaliado/Validado: avaliado em aplicações reais, ou validado em sistemas por meio de experimentos.

Na Figura 9, são apresentadas as ocorrências das características identificadas no MS e para cada característica o número de estudos em que ela foi avaliada/validada.

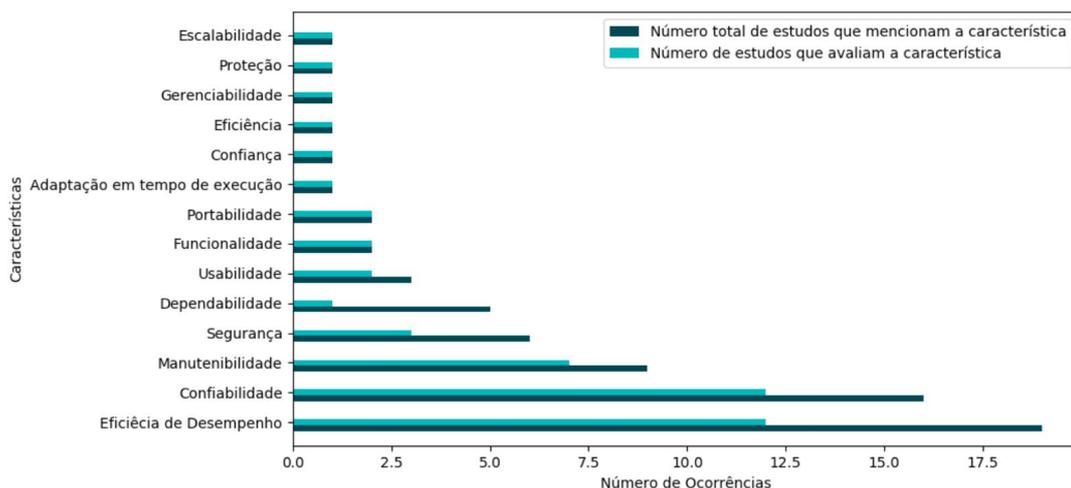


Figura 9 – Características capturadas no mapeamento sistemático por número de ocorrência

Em relação as características de qualidade mais recorrentes pode-se destacar: eficiência de desempenho, confiabilidade e manutenibilidade. A sub-característica de qualidade mais recorrente nos estudos selecionados foi a disponibilidade, o que reforça a importância da confiabilidade para sistemas autoadaptativos, uma vez que elas estão intrinsecamente relacionadas.

Vale ressaltar que o termo eficiência de desempenho foi adotado no presente trabalho pois é o termo proposto na norma ISO 25010 (ISO, 2011a). Nos estudos selecionados foi verificada a ocorrência do termo "desempenho (*performance*)", apresentando a mesma definição que o termo apresentado pela ISO 25010. Portanto, neste trabalho, essa característica foi mapeada como "eficiência de desempenho".

Também é importante observar a ocorrência da característica eficiência, que é uma característica apresentada do modelo de qualidade em uso da norma ISO 25023 (ISO, 2011b) e não deve ser confundida com eficiência de desempenho ou com suas sub-características. No modelo de qualidade em uso a eficiência está relacionada ao grau em que o sistema permite que o usuário alcance seus objetivos de forma completa e precisa (ISO, 2011b), enquanto a eficiência de desempenho diz respeito às propriedades computacionais (ISO, 2011a).

### 3.4.2 QP2. *Quais medidas de qualidade tem sido utilizadas para avaliar sistemas autoadaptativos?*

Esta questão de pesquisa busca investigar quais medidas têm sido utilizadas para avaliar SAS e com quais características de qualidade essas medidas estão relacionadas. Além disso, tal questão também investiga que tipos de estudos têm apresentado medidas de qualidade para SAS no presente estado da arte. Dessa forma, para categorizar os estudos selecionados quanto a sua forma de avaliação, foi utilizada como base a classificação utilizada por Garcés *et al.* (2017), que é apresentada na Tabela 5.

Tabela 5 – Classificação de estudos quanto ao tipo de avaliação. Adaptado de Garcés *et al.* (2017)

Abordagem de Avaliação	Descrição
Uso Industrial	Utilização efetiva das medidas nas indústrias
Estudos em Indústrias	Medidas definidas ou avaliadas em produtos reais ou na indústria
Estudos Acadêmicos	Medidas obtidas/avaliadas a partir de experiências laboratoriais controladas ou resultados baseados em evidências
Observações ou opiniões de especialistas	Medidas definidas como resultado de entrevistas com especialistas
Demonstrações ou exemplos <i>toy</i>	Avaliações por meio de protótipos, uma boa linha de argumentação ou exemplos de aplicação

Como resultado, foi obtido um total de 242 medidas distintas e dentro desse conjunto, algumas medidas se relacionam com mais de uma característica (e.g., LCOM, que se relaciona com manutenibilidade e gerenciabilidade (MUNNELLY; CLARKE, 2008)). Também foram

capturadas medidas que, embora descritas com o mesmo nome, possuem objetivos de medição diferentes (e.g., disponibilidade, para Cetina *et al.* (2013), é medida em termos de números de pontos de falha, já para Perez e Correal (2011), a disponibilidade representa a probabilidade de um serviço ser acessado em um dado período de tempo). A Figura 10 apresenta a quantidade de medidas encontradas por características de qualidade.

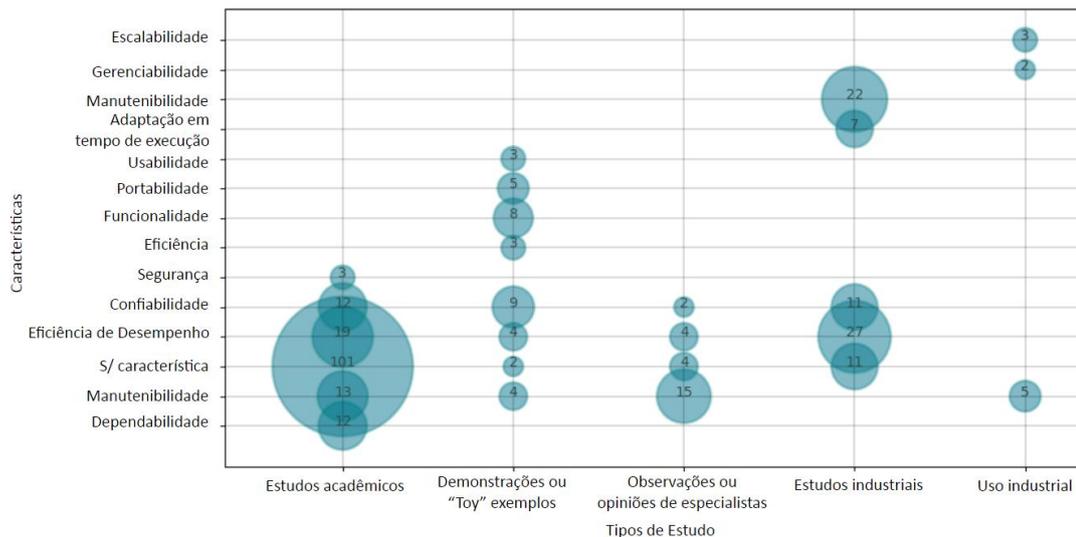


Figura 10 – Número de medidas por características e tipos de estudo.

Na Figura 10, pode-se observar que a maioria das medidas apresentadas pelos artigos não está relacionada com nenhuma característica. Isso pode indicar que as propriedades ou atributos de qualidade medidos nos sistemas autoadaptativos não são contemplados pela norma ISO. Além disso, pode-se notar que a maioria das medidas são utilizadas em estudos acadêmicos, o que demonstra interesse da parte da comunidade de pesquisadores em estudos sobre a avaliação da qualidade de sistemas autoadaptativos.

Ainda na Figura 10, pode-se notar que poucas medidas têm sido aplicadas em ambientes reais ou na indústria. Entretanto, nos estudos realizados identificados pode-se destacar as medidas relacionadas as características de manutenibilidade, desempenho e confiabilidade.

Durante a extração de dados, foi observado que grande parte das medidas encontradas eram mencionadas nos artigos sem apresentar fórmula de cálculo, descrição de como deveriam ser interpretadas e com definições muito breves, dificultando o entendimento para a utilização das mesmas.

Entre as medidas extraídas, as mais recorrentes nos estudos selecionados foram: **disponibilidade** (VILLEGAS *et al.*, 2011), (RAIBULET *et al.*, 2017), (PEREZ; CORREAL,

2011), (CETINA *et al.*, 2013); **confiabilidade** (FRANCO *et al.*, 2016) (VILLEGAS *et al.*, 2011), (RAIBULET, 2014), (PEREZ; CORREAL, 2011); e **tempo de resposta** (CHEN *et al.*, 2009), (VILLEGAS *et al.*, 2011), (PEREZ; CORREAL, 2011), (POTENA, 2013), (SANCHEZ *et al.*, 2015).

### 3.4.3 QP3. *Quais soluções têm sido propostas para avaliar sistemas autoadaptativos?*

Esta questão de pesquisa busca investigar quais tipos de soluções têm sido propostas para avaliar sistemas autoadaptativos, que neste trabalho, consistem na forma que a solução é apresentada (e.g., abordagens, métodos, modelos). Além disso, esta questão de pesquisa tem o objetivo de identificar quais mecanismos de controle os sistemas apresentam e que tipo de pesquisas tem sido conduzidas.

Primeiramente, para identificar os tipos de pesquisas que têm sido conduzidas, foi realizada uma classificação dos estudos selecionados de acordo com as categorias propostas por Petersen *et al.* (2008), listadas a seguir. A classificação do conjunto final de artigos é ilustrada na Tabela 6.

- **Pesquisas de Validação:** As técnicas investigadas são novas e ainda não foram implementadas na prática. As técnicas utilizadas são, por exemplo, experimentos.
- **Pesquisas de Avaliação:** As técnicas são implementadas na prática e a avaliação da técnica é realizada. Ou seja, é mostrado como a técnica é implementada na prática (implementação da solução) e quais são as consequências da implementação em termos de benefícios e desvantagens (avaliação da implementação). Isso também inclui identificar problemas na indústria.
- **Propostas de Solução:** Uma solução para um problema é proposta, a solução pode ser uma novidade ou uma extensão significativa de uma técnica existente. Os benefícios potenciais e a aplicabilidade da solução são demonstrados por um pequeno exemplo ou uma boa linha de argumentação.
- **Trabalhos Filosóficos:** Estes artigos esboçam uma nova forma de olhar para as coisas existentes, estruturando o campo em forma de uma taxonomia ou estrutura conceitual.
- **Trabalhos Opinativos:** São trabalhos que expressam a opinião pessoal de alguém sobre uma determinada técnica, se é uma técnica boa ou ruim, ou como as coisas devem ser feitas. Esses trabalhos não dependem de trabalhos relacionados ou metodologias de pesquisa.
- **Relatos de Experiência:** Os trabalhos de relatos de experiência explicam o que e como

algo foi feito na prática. O artigo deve tratar da experiência pessoal do autor.

Na Tabela 6, pode-se observar que a maioria dos estudos está categorizado como proposta de solução e pesquisa de validação. Esse resultado corrobora com o que foi comentado anteriormente, em relação aos dados da Tabela 5. Esses resultados apontam que a qualidade de sistemas autoadaptativo tem sido estudada, entretanto as técnicas propostas não têm sido avaliadas em sistemas reais ou em indústrias.

Tabela 6 – Classificação por tipos de pesquisa

Categoria de Pesquisa	Artigos	Número de Artigos
Pesquisas de Avaliação	(FARAHANI <i>et al.</i> , 2017), (MUNNELLY; CLARKE, 2008), (MÄRTIN; NICOLAI, 2014), (CRIADO <i>et al.</i> , 2016)	5
Trabalhos Filosóficos	(RAIBULET <i>et al.</i> , 2017)	1
Propostas de Solução	(BEZERRA <i>et al.</i> , 2016), (GOLDSTEEN <i>et al.</i> , 2015), (PEREZ-PALACIN <i>et al.</i> , 2011), (VILLEGAS <i>et al.</i> , 2011), (POTENA, 2013), (MARCOS <i>et al.</i> , 2011), (ZHU <i>et al.</i> , 2004), (MIRANDOLA; POTENA, 2010), (RAIBULET; MASCIADRI, 2009a), (SANCHEZ <i>et al.</i> , 2014) (MING <i>et al.</i> , 2008), (RAIBULET; MASCIADRI, 2009b), (GRAMBOW; OBERHAUSER, 2010), (LOUKIL <i>et al.</i> , 2014), (GRASSI <i>et al.</i> , 2009), (MARZOLLA; MIRANDOLA, 2010), (GHEZZI; SHARIFLOO, 2013)	17
Pesquisas de Validação	(PAUCAR; BENCOMO, 2017), (CHEN <i>et al.</i> , 2009), (MANTORO, 2009), (PEREZ-PALACIN <i>et al.</i> , 2014), (FRANCO <i>et al.</i> , 2016), (PEREZ; CORREAL, 2011), (PESSOA <i>et al.</i> , 2017), (SANILAV <i>et al.</i> , 2016), (CETINA <i>et al.</i> , 2013), (YANG <i>et al.</i> , 2009), (CHENG <i>et al.</i> , 2009b), (SERRAL <i>et al.</i> , 2018), (SANCHEZ <i>et al.</i> , 2015)	13

A Figura 11 apresenta o número de estudos por tipo de contribuição e área, com esse dados é possível verificar que poucos trabalhos propõem medidas de qualidade, o que caracteriza uma oportunidade de pesquisa. Além disso, foi identificado um total de 19 áreas diferentes onde há presença de sistemas autoadaptativos e 09 tipos de contribuições diferentes. A maior parte dos artigos apresenta soluções em forma de abordagem.

Nesta questão de pesquisa, também foi verificado quais mecanismos de controle de adaptação têm sido utilizados pelos SAS. Poucos artigos mencionam qualquer tipo de mecanismo. Entre os trabalhos que reportaram mecanismos de controle de adaptação, foram identificados quatro tipos: **MAPE-K** - Monitorar-Analisar-Planejar-Executar sobre um Conhecimento Compartilhado (*Monitor-Analyze-Plan-Execute over a Shared Knowledge*) (FARAHANI *et al.*, 2017),

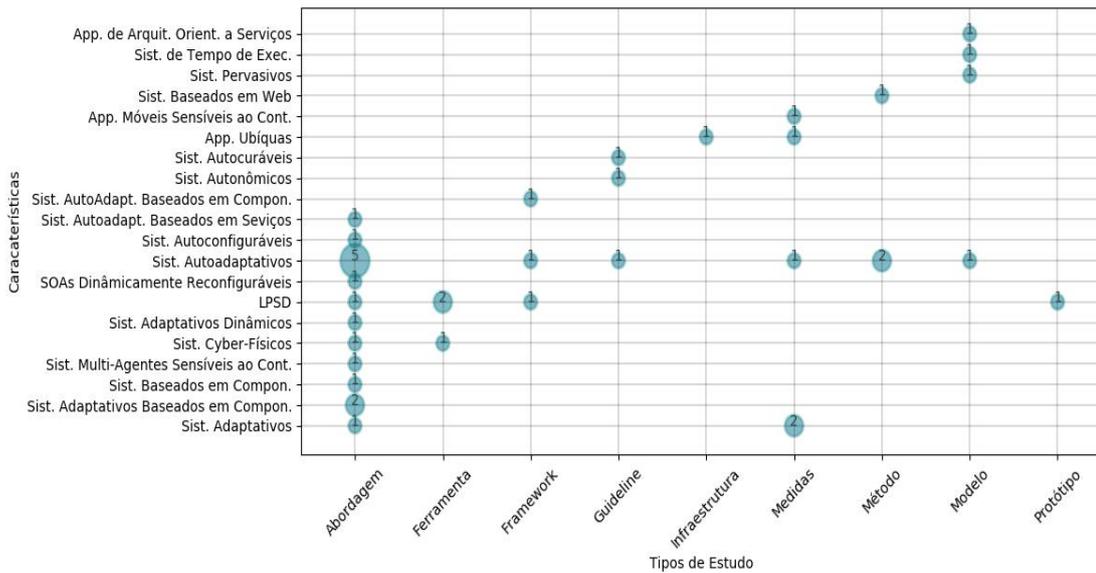


Figura 11 – Número de estudos por tipo de contribuição e área

(GOLDSTEEN *et al.*, 2015), (VILLEGAS *et al.*, 2011), (RAIBULET *et al.*, 2017), (FRANCO *et al.*, 2016), (PESSOA *et al.*, 2017), (GHEZZI; SHARIFLOO, 2013), (SANCHEZ *et al.*, 2015)); **MAPE** - Monitoramento, Análise, Planejamento e Execução (*Monitoring, Analysis, Planning and Execution*) (VILLEGAS *et al.*, 2011), (RAIBULET *et al.*, 2017), (MÄRTIN; NICOLAI, 2014), (CHENG *et al.*, 2009b), (LOUKIL *et al.*, 2014)); **MIAC** - Controle Adaptativo de Identificação de Modelo (*Model Identification Adaptive Control*) (VILLEGAS *et al.*, 2011)); e **MRAC** - Controle Adaptativo do Modelo de Referência (*Model Reference Adaptive Control*) (VILLEGAS *et al.*, 2011)).

### 3.5 Discussão

Com a realização do estudo do mapeamento sistemático foi obtido um conjunto de 36 artigos dos quais foram extraídos dados para investigação. Foi observado que a maioria dos autores não utilizou termos e modelos providos pela norma ISO (ISO, 2011a). Ao invés disso, os artigos apresentaram termos mais genéricos, como fatores de qualidade e critérios de qualidade.

Nos resultados, foram identificadas 14 características de qualidades, 24 subcaracterísticas e 242 medidas de software diferentes. Na análise dos dados coletados, foi verificado que as medidas identificadas foram definidas de forma insuficiente, em termos de forma de cálculo, descrição e elementos de medição, o que pode tornar difícil entender a medida e sua aplicação.

Também foi observado que as características de qualidade mais recorrentes nos estudos são as características apresentadas na norma ISO (ISO, 2011a), indicando que os

pesquisadores tem recorrido aos modelos propostos pela norma para realizar avaliações em sistemas autoadaptativos. Entretanto, existem artigos que abordam características de qualidade que não são providas pelas normas ISO para qualidade de software (e.g., gerenciabilidade, adaptatividade em tempo de execução), e medidas sem relação com as características propostas pela norma, o que pode indicar a necessidade por novas características e medidas de qualidade que se ajustem aos aspectos dinâmicos específico dos SAS.

Uma das características de qualidade mais recorrentes nos estudos foi a confiabilidade, sendo utilizada em pesquisas acadêmicas e em estudos na indústria. Por meio da realização do estudo de mapeamento sistemático, foi observado que a confiabilidade tem uma influencia significativa nas propriedades dos SAS, na performance do sistema, além de impactar a qualidade do serviço. Nesse sentido, para reforçar a avaliação da qualidade de SAS, o escopo deste trabalho se concentra na definição de medidas de confiabilidade.

### **3.6 Considerações Finais**

Neste Capítulo foi apresentado um estudo de Mapeamento Sistemático da literatura. Este estudo foi realizado com o objetivo de investigar avaliações de qualidade em sistemas autoadaptativos e sua diversidade e capturar da literatura características, sub-características, atributos e medidas de qualidade utilizadas para avaliar estes sistemas.

Com a realização do estudo de mapeamento sistemático foi possível identificar um conjunto de características, sub-características, atributos e medidas de qualidade e avaliar a importância de cada elemento. Após a extração de dados foi verificada a importância da característica confiabilidade para os sistemas autoadaptativos.

Com estudo de mapeamento sistemático foi possível observar que embora característica de qualidade confiabilidade cause grande impacto em sistemas autoadaptativos suas sub-características são pouco exploradas as medidas de confiabilidade apresentadas na literatura para SAS não são bem definidas. Nesse sentido, com o objetivo de dar suporte a avaliação da confiabilidade foi definido um conjunto de medidas para sistemas autoadaptativos, apresentado no Capítulo 4.

## 4 MEDIDAS DE CONFIABILIDADE PARA SAS

Este Capítulo apresenta as etapas e métodos utilizados para definir as medidas de confiabilidade para sistemas autoadaptativos propostas nesta dissertação. A Seção 4.1 apresenta as etapas realizadas para a definição das medidas de confiabilidade. A Seção 4.2 descreve o processo de revisão e a Seção 4.3 apresenta a documentação do conjunto final de medidas. Por fim, a Seção 4.4 apresenta as considerações finais deste Capítulo.

### 4.1 Definição de Medidas de Confiabilidade

Como comentado no Capítulo 3, foi observado por meio de um estudo de mapeamento sistemático que a confiabilidade é uma característica importante para sistemas autoadaptativos e pode ser avaliada por meio de medições. Nesse sentido, foi definido um conjunto de medidas de confiabilidade com apoio do método GQM para auxiliar a avaliação de sistemas autoadaptativos no que diz respeito a essa característica. Nesta Seção, são apresentados os passos e os métodos utilizados na definição de medidas de confiabilidade para sistemas autoadaptativos.

#### 4.1.1 Definição do Objetivo

O primeiro passo para definir as medidas de confiabilidade foi a definição formal dos objetivos de medição. Esses objetivos apontam o que será verificado, melhorado, entendido ou controlado em um produto ou processo em particular (SOLINGEN; BERGHOUT, 1999). A definição do objetivo foi realizada e documentada de acordo com o *template* proposto por Solingen e Berghout (1999) e é ilustrada na Tabela 7.

Tabela 7 – Definição do Objetivo (SOLINGEN; BERGHOUT, 1999)

<b>Analisar</b>	<i>sistemas autoadaptativos</i>
<b>Com o propósito de</b>	<i>avaliar</i>
<b>Com respeito a</b>	<i>confiabilidade</i>
<b>Do ponto de vista do</b>	<i>usuário final e desenvolvedor</i>
<b>No contexto de</b>	<i>aplicações de sistemas autoadaptativos</i>

#### 4.1.2 Classificação

A segunda etapa do processo de definição das medidas foi a extração de medidas de confiabilidade apresentadas na literatura. Para extrair essas medidas, foi realizado um

mapeamento sistemático sobre a avaliação da qualidade de sistemas autoadaptativos, como apresentado no Capítulo 3. Em seguida, as medidas de confiabilidade extraídas no mapeamento sistemático foram analisadas com o objetivo de verificar se as mesmas continham todos os elementos necessários para serem utilizadas em uma avaliação de qualidade. Nesse sentido, as medidas foram classificadas (detalhes na Tabela 8) em três categorias conforme apresentado por Carvalho *et al.* (2018):

- **Bem definida:** a medida possui descrição e função de medição, permitindo que um ou mais elementos de medição sejam identificados.
- **Definida mas sem função de medição:** a medida é definida considerando apenas uma propriedade para quantificar.
- **Não definida:** a medida não apresenta nenhuma informação sobre a forma de cálculo, o que torna difícil saber o que está sendo medido.

#### 4.1.3 Definição de Medidas com a Utilização do Método GQM

##### 4.1.3.1 Perfil dos Participantes

Depois da etapa de classificação das medidas, foram conduzidas entrevistas com pessoas que possuem experiência com sistemas autoadaptativos. Foi selecionado um grupo de cinco especialistas com diferentes níveis de formação. Todos os especialistas que participaram da definição de medidas possuem entre um a cinco anos de experiência com sistemas autoadaptativos, com participação em projetos relacionados a sistemas autoadaptativos no âmbito de pesquisa científica. A formação dos participantes é apresentada na Figura 12.

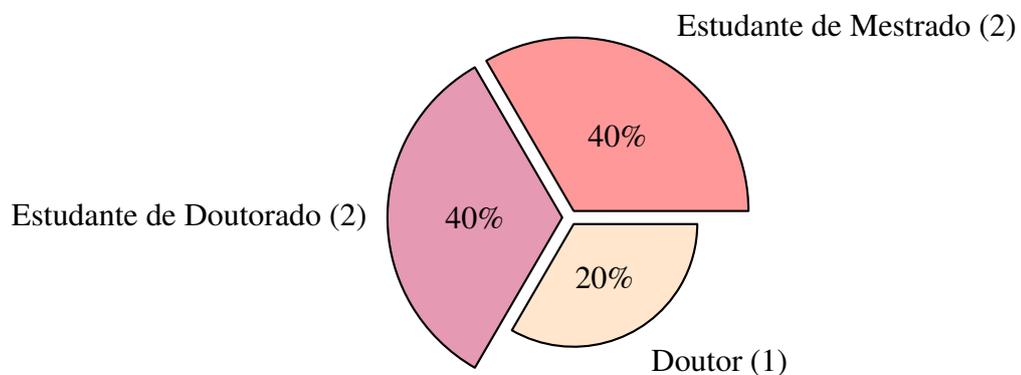


Figura 12 – Formação dos Especialistas

Foi perguntado aos especialistas se eles já haviam utilizado ou analisado medidas de

Tabela 8 – Classificação das medidas extraídas no mapeamento sistemático

ID	Medida de Software	Classificação	Referência
M1	Probabilidade de maximizar a confiabilidade	Definida mas sem função de medição	(PAUCAR; BEN-COMO, 2017)
M2	Disponibilidade	Definida	(VILLEGAS <i>et al.</i> , 2011)
M3	Tempo de Inoperabilidade	Definida	
M4	Confiabilidade Esperada	Definida mas sem função de medição	
M5	Maturidade	Definida mas sem função de medição	(MANTORO, 2009)
M6	Tolerância a Falha	Definida mas sem função de medição	
M7	Escalabilidade	Definida mas sem função de medição	
M8	Recuperabilidade	Definida mas sem função de medição	
M9	Transparência	Definida mas sem função de medição	
M10	Disponibilidade	Não definida	(RAIBULET <i>et al.</i> , 2017)
M11	Confiabilidade	Não definida	
M12	Índice de Robustez	Definida mas sem função de medição	
M13	Tempo Médio Antes da Falha	Não definida	
M14	Previsibilidade de Falhas	Não definida	
M15	Recursos Ativos	Definida mas sem função de medição	(FRANCO <i>et al.</i> , 2016)
M16	Corretude do Funcionamento	Definida mas sem função de medição	
M17	Comportamento de Falha	Definida mas sem função de medição	
M18	Carga	Definida mas sem função de medição	
M19	Disponibilidade	Definida mas sem função de medição	(PEREZ; CORREAL, 2011)
M20	Confiabilidade	Definida mas sem função de medição	
M21	Disponibilidade de um serviço externo provido pelo sistema	Definida	(POTENA, 2013)
M22	Disponibilidade	Definida	(CETINA <i>et al.</i> , 2013)
M23	Severidade da Falha	Definida	
M24	Complexidade Ciclomática	Não definida	(GRAMBOW; OBERHAUSER, 2010)
M25	Complexidade	Não definida	
M26	Fator de Acoplamento	Não definida	
M27	Índice de Defeitos	Não definida	
M28	<i>ClassFanOut</i>	Não definida	
M29	Taxa de Perda de Mensagens (monitorado no nível do componente indivisível/composto)	Definida	(LOUKIL <i>et al.</i> , 2014)
M30	Taxa de Perda de Mensagens (monitorado a nível de comunicação)	Definida	
M31	Taxa de Erro (e tipo de erro)	Não definida	(CRIADO <i>et al.</i> , 2016)
M32	Custo de Erro	Não definida	
M33	Consumo de Energia	Não definida	(GHEZZI; SHARIFLOO, 2013)

qualidade para a avaliação de softwares. Três participantes responderam sim, mas apenas no âmbito da pesquisa, um utilizou medidas tanto no âmbito de pesquisa quanto de indústria, e um não utilizou medidas para avaliar software.

Também foi requisitado que os participantes fizessem uma autoavaliação dos seus conhecimentos técnicos relacionados aos conceitos que constituem o escopo deste trabalho. A Figura 13 ilustra o grau de conhecimento que os participantes alegaram ter em cada tópico.

O perfil dos participantes deste estudo é formado por pessoas que possuem um grau de conhecimento mais profundo em relação aos sistemas autoadaptativos e um nível de

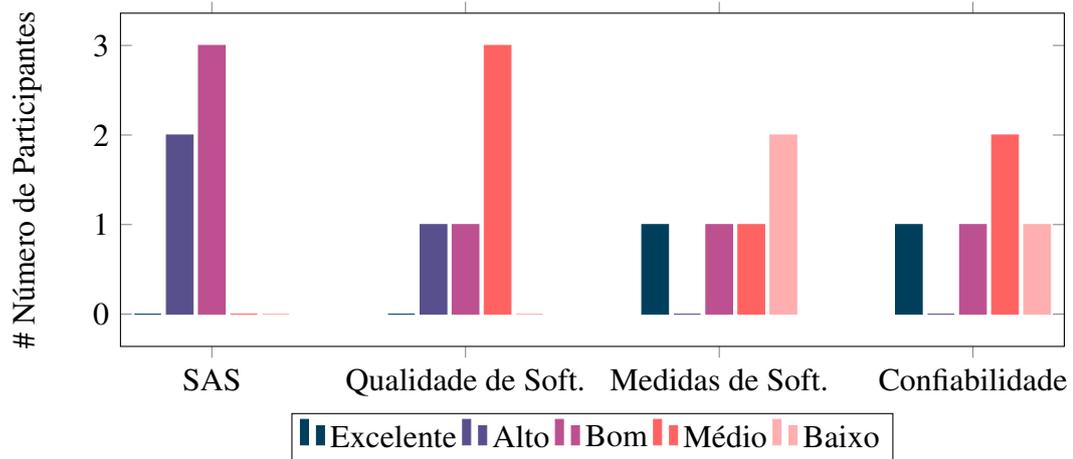


Figura 13 – Grau de conhecimento dos participantes da definição de medidas por tópico

conhecimento mediano no que diz respeito à qualidade de software, confiabilidade e medidas de qualidade.

#### 4.1.3.2 Preenchimento da Planilha de Abstração

A primeira tarefa desta fase foi a apresentação da definição de confiabilidade para os participantes. Em seguida, foi selecionada uma sub-característica de confiabilidade e sua definição também foi apresentada.

Uma vez que o especialista compreendeu a definição da sub-característica, deu-se início ao preenchimento da planilha de abstração proposta pelo método GQM (SOLINGEN; BERGHOUT, 1999). Os participantes preencheram a planilha de abstração na presença da pesquisadora para esclarecer possíveis dúvidas. O *template* da planilha de abstração utilizada neste estudo é ilustrado na Figura 14.

Planilha de abstração			
Objeto	Propósito	Foco de qualidade	Ponto de vista
<b>Foco de qualidade</b>		<b>Fatores de Variação</b>	
<i>Quais são as possíveis medidas para medir um objeto (SAS) em relação a um determinado objetivo (confiabilidade ou subcaracterísticas), de acordo com os membros do projeto (especialistas)?</i>		<i>Que fatores (ambientais) o membro do projeto (especialista) espera que sejam influenciados pelas medidas?</i>	
<b>Hipótese de Base (estimativas)</b>		<b>Impacto dos Fatores de Variação</b>	
<i>Qual é o conhecimento atual do membro do projeto em relação a essas métricas? Suas expectativas são documentadas como "hipóteses de linha de base" das métricas.</i>		<i>Como esses fatores de variação poderiam influenciar as medidas reais? Que tipo de dependências são estabelecidas entre as métricas e os fatores que as influenciam?</i>	

Figura 14 – Planilha de Abstração. Adaptada de (SOLINGEN; BERGHOUT, 1999).

As planilhas de abstração foram preenchidas por um especialista por vez e na ordem determinada pelo método GQM: Quadrante 1 (Foco da Qualidade), Quadrante 2 (Hipótese de Base (Estimativas), Quadrante 3 (Fatores de Variação), e Quadrante 4 (Impacto dos Fatores de Variação). A Figura 15 apresenta o processo de preenchimento da planilha de abstração.

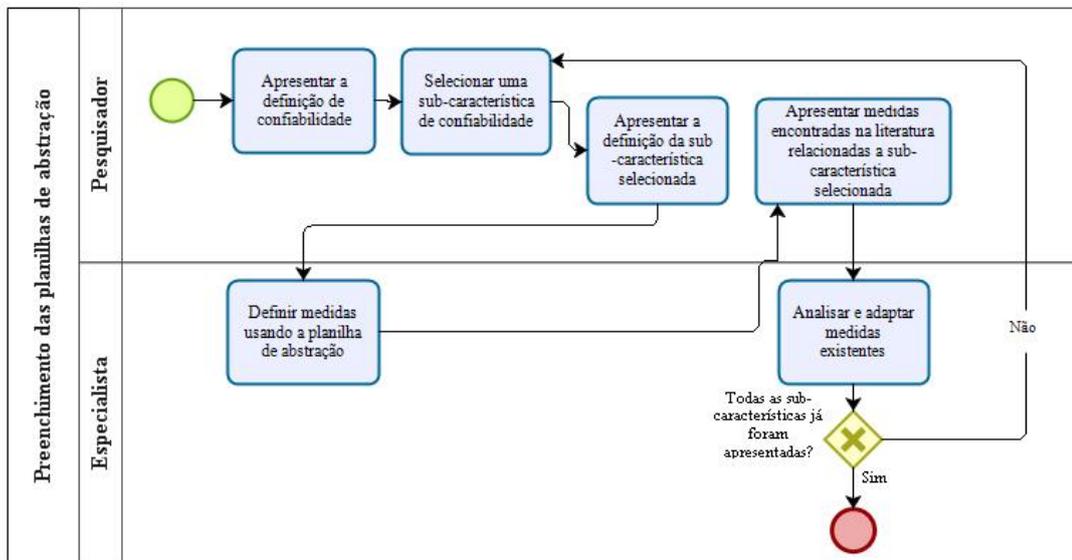


Figura 15 – Processo de Preenchimento das Planilhas de Abstração

Os três primeiros passos tratam da apresentação das definições de confiabilidade e de suas sub-características e são executados pela pesquisadora<sup>1</sup> responsável pela entrevista. Em seguida o especialista utiliza a planilha de abstração para a definição das medidas relacionadas a sub-característica selecionada no passo 2.

Após o término do preenchimento de uma planilha de abstração, referente a uma determinada sub-característica, a pesquisadora apresenta as medidas extraídas do mapeamento sistemático relacionadas a mesma sub-característica. Nesse momento, o participante dá sua opinião sobre as medidas e se deveriam ser adaptadas.

Ao final do processo de definição de medidas, foi obtido um conjunto total de 22 medidas distribuídas entre as sub-características de confiabilidade. Além das sub-características apresentadas pela norma ISO 25010 (ISO, 2011a), também foi incluída a sub-característica Robustez por sugestão de um especialista. A Tabela 9 apresenta a distribuição da quantidade de medidas por sub-características.

Após a definição do conjunto de medidas de confiabilidade, as medidas foram agrupadas por sub-característica e documentadas, considerando os atributos ID, nome, descrição,

<sup>1</sup> Autora desta dissertação de mestrado.

Tabela 9 – Distribuição da quantidade de medidas definidas por sub-característica

Sub-característica	Medidas
Maturidade	3
Disponibilidade	8
Recuperabilidade	3
Tolerância a Falhas	5
Robustez	2

função de medição, elementos de medição, análise, origem da medida quanto a sua natureza de definição e a referência no caso de medidas que foram adaptadas.

Esses atributos foram usados para documentar as medidas pois permitem identificar elementos essenciais para as medições, como o propósito da medida, sua fórmula de cálculo e a interpretação dos resultados. Os atributos utilizados são descritos a seguir:

- ID: identificador da medida (M01, M02, M03 ...).
- Nome: nome da medida.
- Descrição: breve apresentação do propósito da medida.
- Função de Medição: fórmula de cálculo da medida.
- Elementos de Medição: variáveis observadas para o cálculo da medida.
- Análise: descrição de como o resultado da medida deve ser interpretado.
- Origem da Medida: determina a natureza da definição da medida (sugerida por especialista/ existente).
- Referência: no caso de medidas existentes ou adaptadas, indica de onde a medida foi extraída.

#### 4.2 Revisão do Conjunto Inicial de Medidas de Confiabilidade

Após documentação do conjunto inicial de medidas, iniciou-se a etapa de revisão das medidas. Essa etapa teve o objetivo de avaliar as medidas quanto a sua clareza, adequação e relevância para os sistemas autoadaptativos. O propósito dessa revisão foi verificar e tratar ambiguidades presentes na descrição de cada medida, eliminar medidas irrelevantes, caso necessário e melhorar a descrição dos elementos de cada medida para facilitar sua aplicação.

Para realizar a revisão das medidas, foi utilizado um *checklist* de verificação de qualidade de medidas de software e foram recrutados especialistas em sistemas autoadaptativos, formando um grupo de três respondentes. Vale ressaltar que os especialistas que participaram da revisão não participaram da definição das medidas. Os perfis dos especialistas são listados a

seguir:

- 1 especialista com doutorado;
- 1 especialista com mestrado; e
- 1 especialista com mestrado cursando doutorado.

Todos os participantes possuem experiência com sistemas autoadaptativos com atuação em projetos no âmbito da pesquisa científica. Um dos participantes declarou ter entre cinco e dez anos de experiência e dois participantes declararam ter entre um e cinco anos de experiência com sistemas autoadaptativos.

Em relação à medidas de qualidade, dois participantes informaram ter entre um e cinco anos de experiência com medições de software na esfera da pesquisa científica. Também foi solicitado aos participantes que indicassem seu grau de conhecimento nos principais tópicos que norteiam este estudo. A Figura 16 apresenta o grau de conhecimento dos participantes por tópico.

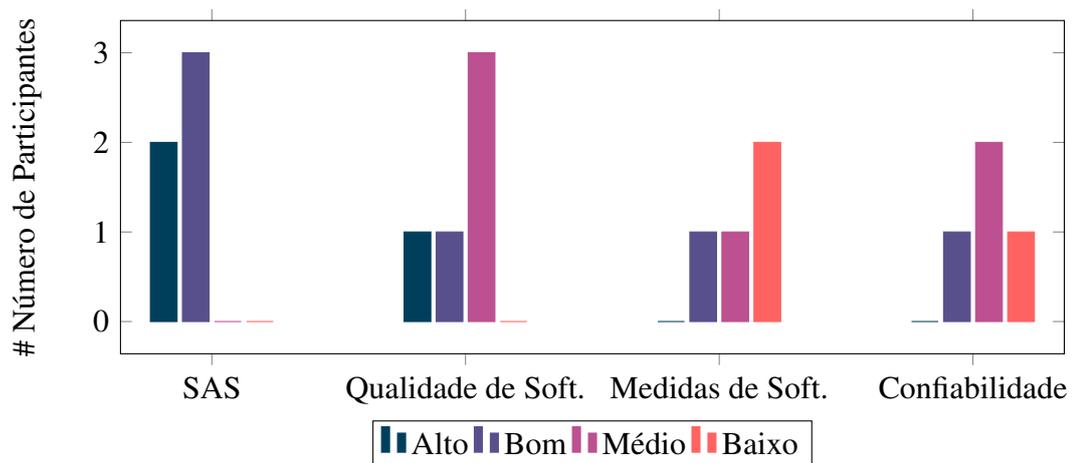


Figura 16 – Grau de conhecimento dos participantes da revisão por tópico

Para guiar a revisão das medidas e analisar cada atributo referente as mesmas, foi utilizado um *checklist* de avaliação de medidas de software adaptado do trabalho de Barcellos *et al.* (2010) que faz parte de um conjunto de *checklists* que buscam avaliar a adequação de um repositório de medição para controle estatístico de processos. O *checklist* utilizado no presente trabalho foi adaptado para contemplar os atributos estabelecidos para as medidas de confiabilidade definidas, considerando a corretude e clareza dos elementos que as definem e é apresentado na Tabela 10.

Tabela 10 – *Checklist* de avaliação de medidas. Adaptado de Barcellos *et al.* (2010)

<b>Medida:</b>	<b>Avaliação</b>					
<b>Legenda:</b> A = <i>Atende</i> , AL = <i>Atende Largamente</i> , AP = <i>Atende Parcialmente</i> , AR = <i>Atende Razoavelmente</i> , NA = <i>Não Atende</i> , NFPA = <i>Não Foi Possível Avaliar</i>						
1. A definição da medida é correta e satisfatória	A ()	AL ()	AP ()	AR ()	NA ()	NFPA ()
A definição da medida inclui de forma clara:	A ()	AL ()	AP ()	AR ()	NA ()	NFPA ()
1.1 Uma definição/ descrição	A ()	AL ()	AP ()	AR ()	NA ()	NFPA ()
1.2 Elementos de medição	A ()	AL ()	AP ()	AR ()	NA ()	NFPA ()
1.3 Fórmula de cálculo	A ()	AL ()	AP ()	AR ()	NA ()	NFPA ()
1.4 Descrição sobre o procedimento de análise	A ()	AL ()	AP ()	AR ()	NA ()	NFPA ()
1.5 Intervalos de dados esperados	A ()	AL ()	AP ()	AR ()	NA ()	NFPA ()
1.6 Descrição dos métodos de coleta	A ()	AL ()	AP ()	AR ()	NA ()	NFPA ()
1.7 Nome adequado ao propósito da medida	A ()	AL ()	AP ()	AR ()	NA ()	NFPA ()
2. O resultado da medida é relevante para a avaliação de sistemas autoadaptativos	A ()	AL ()	AP ()	AR ()	NA ()	NFPA ()
3. A medida é adequada para avaliar a confiabilidade de sistemas autoadaptativos	A ()	AL ()	AP ()	AR ()	NA ()	NFPA ()
4. A medida está associada a uma sub-característica de forma adequada	A ()	AL ()	AP ()	AR ()	NA ()	NFPA ()
Observações:						
Sugestões de melhoria:						

### 4.3 Documentação do Conjunto Final de Medidas de Confiabilidade

Após a etapa de revisão, todas as observações e sugestões de melhorias apontadas pelos especialistas participantes foram analisadas. Nessa análise, algumas medidas foram descartadas por motivos tais como: subjetividade, dificuldade de coletar dados, a relevância da medida não ficou evidente ou a medida era muito similar a alguma outra. São listadas a seguir as medidas removidas e a justificativa apresentada pelos especialistas para cada remoção.

- *Tempo de disponibilidade*: foi removida, pois sua fórmula de cálculo e descrição eram muito similares a M02 (tempo médio sem falhar);
- *Intervalo entre o recebimento de estímulos*: foi removida, pois a sua relevância não ficou evidente para os especialistas, que também alegaram insatisfação com a fórmula de cálculo, que teria que considerar também os requisitos do sistema e outros fatores;
- *Precisão do monitoramento*: foi removida, pois a descrição da medida não estava clara e não ficou evidente a relevância para SAS do ponto de vista dos especialistas;
- *Taxa de falhas toleradas em regras de adaptação*: foi removida, pois foi relatado ser difícil mapear e definir o que é uma regra com falha;
- *Previsibilidade de falhas*: foi removida, pois os especialistas não identificaram uma forma de coleta para realizar o cálculo da medida; e
- *Features operacionais dada restrição de recursos*: foi removida, pois trata-se de uma

medição subjetiva que depende das especificações internas do SAS.

Todas as medidas definidas neste trabalho são baseadas nos conceitos das normas ISO (ISO, 2011b) e ISO 25010, e possuem, portanto, uma sub-característica de qualidade e uma característica de qualidade. Uma visão geral de medidas por sub-característica é discutida a seguir.

As medidas de maturidade são propostas para verificar se o sistema atende aos requisitos que lhe foram propostos sob condições normais, como comentado na Seção 2. Essas medidas devem ser capazes de medir o quão estável é o sistema para lidar com falhas decorrentes de faltas no seu próprio código, com que frequência o sistema apresenta comportamentos inesperados e até se a cobertura de testes realizados no sistema tem impactos na forma do seu funcionamento (ISO, 2011b).

As medidas de disponibilidade se concentram em avaliar se o sistema está operacional e acessível no momento em que o usuário precisa (ISO, 2011a) (ISO, 2011b). Medidas de disponibilidade geralmente envolvem muitas variáveis relacionadas ao tempo, verificando de forma geral a proporção do tempo total de funcionamento em que sistema está funcionando normalmente (*uptime*). Nesse sentido, quanto mais tempo durar a coleta dos elementos de medição para as medidas de disponibilidade mais assertivo será o resultado. Além das medidas relacionadas ao tempo em que o sistema permanece operacional, também cabe verificar a proporção de serviços ou funcionalidades fornecidas pelo sistema.

Em relação às medidas de tolerância a falhas, deve-se verificar se o sistema se comporta como o esperado mesmo na presença de falhas de software ou de hardware. O comportamento esperado do sistema está relacionado aos níveis de performance que o sistema é capaz de manter em uma situação de falha ou em uma situação em que sua interface gráfica é comprometida (ISO, 2011b). Nesse caso, pode ser verificada a redundância de componentes críticos, número de falhas evitadas por tratamento de exceções e se há algum tipo de registro de falhas no sistema.

As medidas de recuperabilidade devem ser capazes de medir se o sistema é capaz de restabelecer seu nível adequado de desempenho e recuperar os dados que foram afetados pela ocorrência de uma determinada falha (ISO, 2011a). Uma das formas de determinar a recuperabilidade de um sistema é verificando o intervalo de tempo que o sistema precisa para se recuperar de uma falha (ISO, 2011b).

As medidas de robustez verificam se o sistema é capaz de manter o desempenho

e a interface de forma aceitável mesmo em condições adversas de estresse, sobrecarga ou até entradas inválidas do usuário (WIEGERS; BEATTY, 2013). Uma das formas de atestar a robustez de um software é realizando testes de estresse de forma gradual, coletando os dados de cada medição até que o sistema pare de funcionar.

O conjunto final de medidas proposto neste trabalho é composto por: medidas existentes que foram simplesmente incluídas sem nenhuma alteração, medidas existentes que foram adaptadas de acordo com sugestões de especialistas e medidas novas, que foram definidas a partir dos conhecimentos de especialistas estruturados nas planilhas de abstração propostas pelo método GQM. As planilhas de abstração preenchidas pelos especialistas são apresentadas nos apêndices A, B, C, D, E.

No conjunto final, pode-se observar que a maioria das medidas são novas, o que pode indicar que os especialistas vislumbraram aspectos a serem avaliados que não são cobertos por medidas existentes. Em contrapartida, também foram preservadas perspectivas de software tradicionais como pontos a serem avaliados, o que resultou na inclusão de medidas apresentadas nas normas ISO (ISO, 2011b), como as medidas M01, M04 (ver Tabelas 11, 12).

As medidas adaptadas, como M02, M08 e M09, foram extraídas dos resultados do mapeamento sistemático (Capítulo 3) e foram adaptadas por não apresentarem definições satisfatórias, como a falta da fórmula de cálculo ou do método de coleta. Entretanto, elas foram incluídas no conjunto de medidas deste trabalho, pois os especialistas julgaram que a essência do conceito delas seria relevante para avaliar a confiabilidade sistemas autoadaptativos.

Por fim, o conjunto final de medidas de confiabilidade foi formado por 16 medidas distribuídas em cinco sub-características, conforme ilustrado nas Tabelas 11, 12, 13, 14, 15.

A maior parte das medidas do conjunto final está associada à sub-característica disponibilidade, o que condiz com os resultados do mapeamento sistemático que apontaram uma relevância significativa em relação a essa sub-característica. Uma possível explicação para esses resultados é que essa sub-característica tem impactos positivos no desempenho dos sistemas autoadaptativos e representa o quão operacional o sistema está quando se precisa dele.

Considerando o *background* de sistemas autoadaptativos, seu funcionamento e mecanismos de controle de autoadaptação, vale ressaltar que as medidas definidas neste trabalho tratam dados que podem ser coletados durante a etapa de Execução dos loops de controle MAPE e MAPE-K, que são utilizados na maioria dos sistemas autoadaptativos. Além disso, as medidas propostas são divididas em medidas genéricas (aplicáveis em sistemas tradicionais) e em medidas

específicas para sistemas autoadaptativos.

O conjunto de medidas proposto neste trabalho pode ser utilizado para verificar a confiabilidade dos sistemas autoadaptativos, comparar a confiabilidade de sistemas que usam algoritmos de estratégia de adaptação diferentes e ajudar na melhoria da qualidade desses sistemas.

Tabela 11 – Medidas de Maturidade

ID	Medida	Descrição	Elementos de Medição	Forma de Cálculo	Intervalo de dados esperados	Análise	Unidade de medida	Forma de Coleta	Origem	Ref
M01	Tempo médio entre falhas	Representa a frequência com a qual o sistema falha durante a sua operação	A = tempo de operação B = número total de falhas detectadas	A/B	$\mathbb{R} = \{1, 2, 3, \dots\}$	Quanto maior, melhor	Tempo / Falhas	Instrument. de código / Observação	Existente	(ISO, 2011b)
M02	Tempo médio sem falhar	Representa o tempo em percentagem que o sistema realiza operações sem nenhuma falha	$TTO =$ tempo total de operação $X_i =$ instante final de recuperação da falha $i$ $Y_i =$ instante em que a falha $i$ ocorre $N =$ número de falhas $i =$ iésima falha, variando de 0 a N	$1 - \frac{\sum(X_i - Y_i)}{TTO}$	0 a 100%	Quanto maior, melhor	Tempo	Logs/ Observação do sistema	Sugerida por especialista	Adaptada de Raibulet <i>et al.</i> (2016)
M03	Tempo de operação com o desempenho adequado	Representa a percentagem do tempo em que o SAS opera com o desempenho adequado (sem atrasos ou travamentos, considerando o tempo de resposta e variação de dados)	A = tempo de operação com o desempenho requerido B = tempo total de operação	A/B	0 a 100%	Quanto maior, melhor	Tempo	Logs/ Instrument. de código/ Observação/ Monitoramento do desempenho	Sugerida por especialista	Nova Medida

Tabela 12 – Medidas de Disponibilidade

ID	Medida	Descrição	Elementos de Medição	Fórmula de Cálculo	Intervalo de dados esperados	Análise	Unidade Medida	Forma de Coleta	Origem	Ref
M04	Tempo médio de inatividade	Representa o tempo médio que o sistema fica indisponível quando ocorre uma falha	A = tempo total de indisponibilidade B = número de interrupções ou falhas observadas	A/B	$R = \{1, 2, 3, \dots\}$	Quanto menor, melhor	Tempo/Falhas	Logs/ Observação do sistema	Existente	(ISO, 2011b)
M05	Tempo de resposta de uma requisição dada a ocorrência de uma adaptação em paralelo	Representa o tempo que o SAS leva para responder a uma requisição enquanto uma adaptação está ocorrendo	A = instante de resposta da requisição B = instante de envio da requisição	A – B	0 a 1000ms	Quanto menor, melhor	Tempo	Logs/ Observação do sistema	Sugerida por especialista	Nova Medida
M06	Taxa de respostas respondidas	Representa a taxa de requisições respondidas pelo SAS	A = número de requisições respondidas B = número total de requisições	A/B	0 a 100%	Quanto maior, melhor	Requisições	Logs/ Observação do sistema	Sugerida por especialista	Nova Medida
M07	Taxa de Interrupções causadas por adaptação	Representa a taxa de interrupções no SAS que decorrem de adaptações	A = número interrupções por adaptação B = número total de interrupções	A/B	0 a 100%	Quanto menor, melhor	Interrupções	Logs/ Observação do sistema	Sugerida por especialista	Nova Medida
M08	Taxa de adaptações corretas	Representa o número de adaptações sem falhas	A = número de adaptações sem falha B = número total de adaptações	A/B	0 a 100%	Quanto maior, melhor	Adaptações	Logs/ Observação do sistema	Sugerida por especialista	Adaptada de Franco <i>et al.</i> (2016)

Tabela 13 – Medidas de Recuperabilidade

ID	Medida	Descrição	Elementos de Medição	Fórmula de Cálculo	0 a 100%	Quanto maior, melhor	Unidade de Medida	Forma de Coleta	Origem	Ref
M09	Tempo médio de recuperação	Representa o tempo médio que o sistema leva para se recuperar totalmente de uma falha	$X_i$ = instante final da recuperação $Y_i$ = instante inicial da recuperação $i$ = i-ésima falha, variando de 0 até $N$ $N$ = número de falhas reparadas	$\Sigma(X_i - Y_i)/N$	Tempo	Quanto menor, melhor	Tempo / Número de Falhas Reparadas	Logs / Observação do sistema	Sugerida por especialista	Adaptada de ISO (2011b)
M10	Taxa de recuperação de falhas operacionais	Representa a quantidade média de falhas operacionais recuperadas pelo SAS	$A$ = número de falhas recuperadas $B$ = número total de falhas ocorridas	$A/B$	0 a 100%	Quanto maior, melhor	Falhas	Logs / Observação do sistema	Sugerida por especialista	Nova Medida
M11	Taxa de recuperação de falhas de adaptação	Representa a quantidade média de falhas de adaptação recuperadas pelo SAS	$A$ = número de falhas recuperadas $B$ = número total de falhas ocorridas	$A/B$	0 a 100%	Quanto maior, melhor	Falhas	Logs / Observação do sistema	Sugerida por especialista	Nova Medida

Tabela 14 – Medidas de Tolerância a Falhas

ID	Medida	Descrição	Elementos de medição	Fórmula de cálculo	Intervalo de dados esperados	Análise	Unidade de medida	Forma de coleta	Origem	Ref
M12	Taxa de requisições respondidas na presença de falha no sistema	Representa a porcentagem de requisições atendidas dada a ocorrência de uma falha no sistema	A = número de requisições respondidas enquanto o sistema se recupera de uma falha B = número total de requisições	A/B	0 a 100%	Quanto maior, melhor	Requisições	Logs / Observação do sistema	Sugerida por especialista	Nova Medida
M13	Falhas de adaptação toleradas	Representa a quantidade média de falhas de adaptação toleradas	A = número de falhas de adaptação toleradas B = número total de falhas de adaptação	A/B	0 a 100%	Quanto maior, melhor	Falhas	Logs / Observação do sistema	Sugerida por especialista	Nova Medida
M14	Nível de tolerância a falhas	Representa capacidade de manter um nível específico de desempenho no caso de falhas de software e dispositivos ou de violação de suas interfaces especificadas.	A = número de falhas toleradas B = número total de falhas	A/B	0 a 100%	Quanto maior, melhor	Falhas	Logs / Observação do sistema / Monitoramento do desempenho	Sugerida por especialista	Adaptada (MANTORO, 2009)

Tabela 15 – Medida de Robustez

ID	Medida	Descrição	Elementos de medição	Fórmula de cálculo	Intervalo de dados esperados	Análise	Unidade de medida	Forma de coleta	Origem	Ref
M15	Taxa de adaptações corretas dada inconsistência de contexto	Representa a porcentagem de adaptações que o SAS é capaz de executar corretamente, mesmo com inconsistências de contexto.	<p><math>A</math> = número de adaptações corretas dada uma inconsistência de contexto</p> <p><math>B</math> = número total de adaptações com inconsistência de contexto</p>	$A/B$	0 a 100%	Quanto maior, melhor	Adaptações	Logs / Observação do sistema	Sugerida por especialista	Nova Medida

#### 4.4 Considerações Finais

Neste Capítulo, foram apresentadas as atividades e métodos utilizados para a definição de medidas de confiabilidade propostas neste trabalho. O processo de definição de medidas contou com um conjunto total de cinco participantes, especialistas em sistemas autoadaptativos que atuam na área acadêmica.

Além do processo de definição, também foi apresentada a atividade de revisão das medidas que teve por objetivo verificar a relevância, clareza e pontos de melhoria em cada medida definida. Essa atividade baseou-se em uma análise orientada a um *checklist* de verificação de medidas de software e contou com a participação de três participantes.

Por fim, foi apresentado o conjunto final de medidas de confiabilidade definidas para sistemas autoadaptativos. Cada medida do conjunto contém 12 atributos que indicam principalmente o seu propósito, forma de coleta, forma de cálculo e interpretação.

Uma vez obtido e revisado um conjunto de medidas, é necessário verificar a potencial aplicabilidade de cada medida e seus benefícios para a melhoria da qualidade de sistemas autoadaptativos. Nesse sentido, foi realizada uma avaliação das medidas propostas por meio de uma prova de conceito aplicada em um exemplar de sistemas autoadaptativos, apresentada no Capítulo 5.

## 5 AVALIAÇÃO DAS MEDIDAS DE CONFIABILIDADE

Este Capítulo discorre sobre a avaliação das medidas de confiabilidade apresentadas no Capítulo 4. O objetivo da avaliação é verificar, por meio de uma prova de conceito, a viabilidade e aplicabilidade das medidas de confiabilidade definidas, bem como seus potenciais benefícios. A Seção 5.1 apresenta as atividades desenvolvidas para a execução da prova de conceito. A Seção 5.2 apresenta uma discussão sobre os resultados obtidos. A Seção 5.3 lista as ameaças à validade deste estudo, e por fim, a Seção 5.4 apresenta as considerações finais deste Capítulo.

### 5.1 Prova de Conceito

Como o próprio nome sugere, provas de conceito são técnicas utilizadas para demonstrações de conceitos numa perspectiva prática, visando atestar a viabilidade e aplicabilidade do conceito estudado (GLASS *et al.*, 2002). É uma metodologia bastante utilizada em pesquisas de engenharia de software para avaliar trabalhos do tipo “proposta de solução” (PETERSEN *et al.*, 2008) com o objetivo de demonstrar aplicações de abordagens, métodos, modelos matemáticos, artefatos e protótipos de sistemas (Latre *et al.*, 2016) (Dos Reis Fontes *et al.*, 2017) (BELL *et al.*, 1994).

A prova de conceito foi escolhida para a avaliação das medidas de confiabilidade para sistemas autoadaptativos propostas nesta dissertação, uma vez que a intenção da avaliação é verificar o quão aplicáveis são as medidas propostas na prática e identificar potenciais benefícios a partir da sua utilização.

Para realizar a prova de conceito, foi definido um conjunto de atividades, conforme ilustra a Figura 17. Primeiro, foi realizado um planejamento no qual foram definidos os recursos, o sistema utilizado e as tarefas para realizar a prova de conceito. Em seguida, na etapa de execução, as medidas são coletas no sistema selecionado com o auxílio de uma biblioteca em Java que foi desenvolvida para facilitar a coleta e o cálculo das medidas. A última etapa da prova de conceito é a análise dos resultados e verificação da aplicabilidade das medidas.

As Seções a seguir descrevem as atividades e recursos utilizados na prova de conceito com detalhes. A Seção 5.1.1 apresenta a biblioteca desenvolvida para apoiar a coleta e cálculo das medidas propostas. A Seção 5.1.2 discorre sobre as atividades de planejamento da prova de conceito. A Seção 5.1.3 apresenta os passos realizados na execução da prova de conceito. A

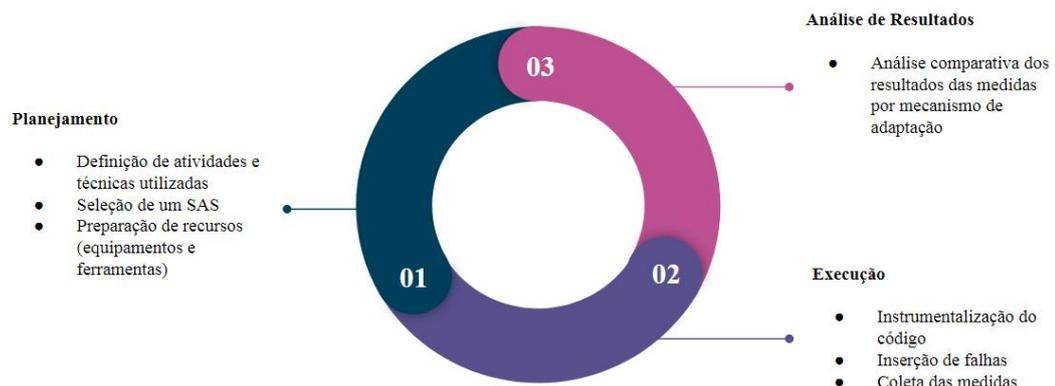


Figura 17 – Atividades da Prova de Conceito

Seção 5.1.4 discute os resultados encontrados.

### 5.1.1 Biblioteca CECILIA

Com o objetivo de facilitar a instrumentalização de código e a coleta das medidas propostas neste trabalho, foi implementada, com base nas medidas de confiabilidade propostas, a biblioteca de medidas de Confiabilidade para Sistemas Autoadaptativos (CECILIA). Essa biblioteca foi implementada na linguagem Java e seu uso diminui a adição de código no fonte do sistema evitando grandes impactos na manutenção do código, uma vez que o cálculo de cada medida está contido na biblioteca. Assim, para viabilizar a utilização da biblioteca CECILIA, os parâmetros utilizados para calcular as medidas devem ser identificados pelo avaliador e repassados para a função correspondente a medida que se deseja obter.

A biblioteca CECILIA<sup>1</sup> é composta pelas 15 medidas de confiabilidade propostas neste trabalho e são representadas por métodos divididos por sub-características. Ela pode ser utilizada para auxiliar na avaliação da confiabilidade de SAS desenvolvidos em Java, bem como para comparar diferentes sistemas que realizam adaptações dinâmicas.

### 5.1.2 Planejamento

No desenvolvimento deste estudo, foi selecionado o TAS (*Tele Assistance System*)<sup>2</sup>. O TAS é um exemplar de sistema autoadaptativo do domínio de *healthcare* baseado em serviços e, dessa forma, lida com serviços oferecidos por terceiros que muitas vezes são compostos de forma dinâmica e oferecem funcionalidades complexas (WEYNS; CALINESCU, 2015).

<sup>1</sup> <https://drive.google.com/drive/folders/1zEVbLt52-fBOr8Tp44Tr1KksBTe9KLVr?usp=sharing>

<sup>2</sup> <https://people.cs.kuleuven.be/~danny.weyns/software/TAS/>

Esse sistema fornece suporte de saúde para pessoas que sofrem de doenças crônicas em suas casas. O TAS utiliza uma combinação de sensores embutidos em um dispositivo vestível e serviços remotos de prestadores de serviços de saúde, farmacêuticos e de emergência. O fluxo de trabalho do TAS realiza medições periódicas dos parâmetros vitais de um paciente e emprega um serviço médico terceirizado para sua análise (WEYNS; CALINESCU, 2015).

Esse sistema é bastante conhecido pela comunidade de pesquisadores em sistemas autoadaptativos, e já tendo sido utilizado para avaliar diversas soluções nesse tema (BARESI *et al.*, 2007), (FILIERI *et al.*, 2012). Por ser um sistema bem aceito na comunidade e permitir acesso ao código, o TAS foi escolhido para ser utilizado nesta prova de conceito.

O sistema TAS possui duas versões, uma com mecanismo de adaptação mais simples e outra que os autores alegam ser mais eficiente, denominado *ActivForms* (WEYNS; CALINESCU, 2015). Neste trabalho de dissertação, foi utilizado o projeto com o mecanismo de adaptação *ActivForms*. Na Figura 18, é apresentada uma visão da tela inicial do sistema.

O TAS apresenta um conjunto de serviços que são acionados por meio de invocações que variam de acordo com o atributo de qualidade os quais são chamados de perfil de entrada, a seguir: *ReliabilityQoS*, *Cost* e *PreferredQoS*. Quando o usuário seleciona um determinado perfil de entrada, o sistema aciona os mecanismos de adaptação alterando seu fluxo de trabalho e parâmetros para invocar serviços (WEYNS; CALINESCU, 2015).

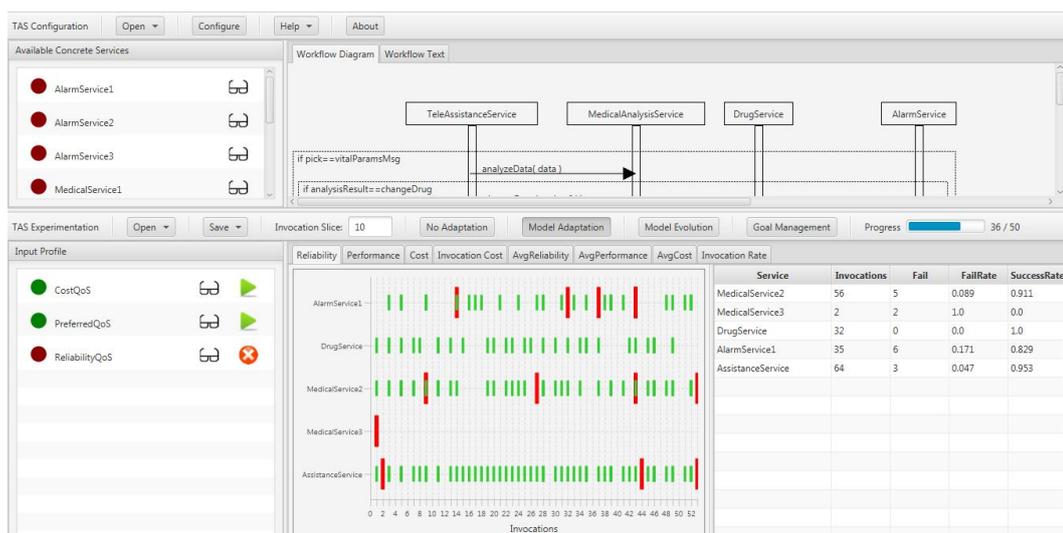


Figura 18 – Tela inicial do TAS (*Tele Assistance System*)

Além do sistema principal utilizado para avaliar as medidas de confiabilidade, também foram utilizadas ferramentas de suporte para a coleta dos dados das medidas. As ferramentas utilizadas são listadas a seguir:

- Eclipse IDE<sup>3</sup>: Importação do projeto e ambiente de coleta de medidas.
- JConsole<sup>4</sup>: Monitoramento do desempenho do sistema.

### 5.1.3 Execução

A primeira etapa da coleta das medidas foi o desenvolvimento da biblioteca CECILIA, que tem o objetivo de coletar os elementos de medição do sistema sob estudo e realizar os cálculos das medidas. Essa biblioteca foi desenvolvida em JAVA por ser a linguagem do sistema selecionado para a prova de conceito deste trabalho.

O projeto *Tele Assistance System* (TAS) foi importado na *Integrated Development Environment* (IDE) Eclipse 2019 em uma máquina com sistema operacional Windows 7 Professional, com processador core i7 e 8GB de *Random Access Memory* (RAM). O sistema é dividido em três projetos: “ResearchServicePlataform”, que contém classes relacionadas ao monitoramento dos serviços sistema, “TAS\_Gui”, que contém classes de visão e controle do sistema, e “TeleAssistanceSystem”, que contém as classes dos mecanismos de adaptação e os métodos de acionamento de serviços.

O código do sistema foi estudado e instrumentalizado com o auxílio da biblioteca CECILIA. A instrumentalização do sistema foi realizada considerando os principais elementos do sistema, que são: invocações, serviços, falhas e adaptação do fluxo de execução. Os elementos do sistema são descritos a seguir:

- Invocações: representam as operações executadas pelo sistema para a obtenção de um serviço.
- Serviços: representam o conjunto de funções que são solicitadas em tempo de execução para compor o sistema.
- Falhas: representam uma condição anormal em determinada porção do sistema, podendo impedir o fluxo de funcionamento planejado. Neste trabalho, as falhas do sistema TAS foram divididas por tipo:
  - Falhas de Invocação: são falhas que ocorrem ao solicitar um serviço e são monitoradas pelo próprio sistema.
  - Falhas de Adaptação: são falhas que ocorrem durante a execução da adaptação, podem causar interrupções, ou adaptações que não atendam ao contexto requerido.

<sup>3</sup> <https://www.eclipse.org/downloads/>

<sup>4</sup> <https://docs.oracle.com/javase/7/docs/technotes/guides/management/jconsole.html>

- Falhas Operacionais: são falhas relacionadas à funcionalidades, podem ser observadas por meio de exceções não tratadas, funções sem resposta ou travamentos.
- Adaptações: representam as alterações no fluxo de execução do sistema no momento do seu funcionamento.

Uma variável importante dentre as utilizadas para coletar as medidas foi o tempo de resposta das invocações e das adaptações. Esse tempo foi calculado a partir da subtração do instante final pelo instante inicial, tanto das invocações quanto das adaptações. A coleta desses dados foi feita por meio de instrumentalização de código, e por esse motivo, pode existir um impacto ou adição no tempo de processamento da resposta do sistema.

Além dos elementos listados, também foi considerado o tempo de execução dos métodos do sistema para fazer o cálculo das medidas de confiabilidade e foi observado o comportamento do sistema na perspectiva de usuário e na perspectiva de desempenho com o auxílio da ferramenta JConsole.

No código do sistema, foram instrumentalizadas 14 medidas do total de 16 medidas. A medida M03 (Tempo de operação com desempenho adequado) foi verificada por meio da ferramenta JConsole.

As coletas das medidas foram divididas pelos mecanismos de adaptação apresentados pelo sistema, que são: *No Adaptation*, *Model Adaptation*, *Model Evolution* e *Goal Management*. Cada coleta durou aproximadamente 15 minutos, entretanto, não foi possível coletar as medidas do sistema utilizando o mecanismo *Goal Management*, pois o sistema disparou muitas exceções e o processo de adaptação entrou em *deadlock*.

Durante o tempo de coleta foram realizados alguns procedimentos específicos iguais para todos os mecanismos de adaptação, a fim de forçar comportamentos no sistema que permitissem a coleta dos dados necessários para o cálculo das medidas. Esses procedimentos foram:

- Acionamento do perfil de entrada *ReliabilityQoS*.
- Acionamento do perfil de entrada *ReliabilityQoS* em paralelo ao perfil de entrada *PreferredQoS*.
- Acionamento do perfil de entrada *ReliabilityQoS* em paralelo ao perfil de entrada *CostQoS*.
- Acionamentos dos três perfis de entrada em paralelo.
- Acionamento do perfil de entrada *ReliabilityQoS* com interrupção antes da finalização do processo de adaptação.

Vale ressaltar que durante a coleta foi dada mais ênfase no perfil de entrada *ReliabilityQoS*, uma vez que seu propósito é maximizar a confiabilidade do sistema durante as adaptações. Por esse motivo, o perfil de entrada *ReliabilityQoS* está em todas as combinações dos procedimentos de execução do sistema.

#### 5.1.4 *Análise dos Resultados*

Nesta Seção, são apresentados os resultados obtidos por meio da coleta de dados das medidas de confiabilidade aplicadas para avaliar o TAS. Os resultados encontrados são discutidos para cada sub-característica de confiabilidade definida neste estudo. Vale ressaltar que os resultados obtidos com as medidas não podem ser generalizados uma vez que a coleta foi realizada em período curto de tempo e com um único sistema.

Esta Seção está dividida em cinco subseções que correspondem aos resultados das medidas de cada sub-característica. A subseção 5.1.4.1 apresenta os resultados das medidas de maturidade, a subseção 5.1.4.2, os resultados das medidas de disponibilidade, a subseção 5.1.4.3 os resultados das medidas de recuperabilidade, a subseção 5.1.4.4 os resultados das medidas de tolerância a falhas, e por fim, a subseção 5.1.4.5 apresenta os resultados da medida de robustez.

##### 5.1.4.1 *Maturidade*

Para verificar a maturidade do TAS, foram utilizadas as medidas M01, M02 cujos resultados estão dispostos na Tabela 16 e a medida M03 que foi coletada com o auxílio da ferramenta JConsole.

A medida M01 (tempo médio entre falhas) teve um resultado melhor no mecanismo de adaptação *Model Adaptation* (118,71s), pois é uma medida de interpretação “quanto maior, melhor”. Isso significa dizer que quanto maior o tempo entre falhas, menos falhas o sistema apresenta. Essa medida também pode ser utilizada para prever o momento de uma falha, no caso de avaliações mais robustas com coletas de dados com maior duração (i.e., meses, anos).

A medida M02 (tempo médio sem falhar) é similar à medida M01, entretanto a M02 considera que nenhuma falha aconteceu e que nenhuma falha está sendo reparada durante o tempo usado para o cálculo da medida. Nesse quesito, o resultado do sistema sem mecanismo de adaptação teve um resultado um pouco melhor (99%), pois as falhas de invocações ou operacionais acontecem muito rapidamente e não são reparadas nesse sistema. No caso dos dois mecanismos de adaptação, as falhas de adaptação levam um pouco mais de tempo para serem

recuperadas.

Tabela 16 – Resultados da coleta e cálculo das medidas de maturidade

Sub-característica	Medida	Resultados por Mecanismos de Adaptação		
		Sem adaptação	<i>Model Adaptation</i>	<i>Model Evolution</i>
Maturidade	M01	67,36s	118,71s	24,19s
	M02	99%	98,84%	98,36%

A medida M03 (tempo de operação com desempenho adequado) é uma medida que considera fatores de desempenho. Por esse motivo, para a verificação da mesma foi utilizada uma ferramenta de monitoramento de desempenho, que verificou uso de memória, processamento de CPU, e *threads* ativas. As três abordagens de adaptação tiveram comportamento semelhante, nenhuma delas apresentou um estouro de memória ou alto consumo de processamento. Em relação ao tempo de resposta, as invocações de serviços sem adaptações apresentaram mais rapidez com um valor médio de 3ms, o mecanismo *Model Adaptation* teve um tempo de resposta médio de 13ms, enquanto o mecanismo *Model Evolution* apresentou um tempo médio de 10ms.

Isso posto, foi considerado que em 100% do tempo de funcionamento do sistema o seu desempenho foi adequado. Vale ressaltar que para o uso dessa medida, os parâmetros de desempenho devem ser previamente definidos. Outros parâmetros além dos apresentados neste trabalho podem ser utilizados, como vazão de dados e número de operações por segundo.

#### 5.1.4.2 Disponibilidade

Em relação à sub-característica disponibilidade, foram coletadas cinco medidas e seus respectivos resultados são apresentados na Tabela 17. A medida M04 (tempo médio de inatividade) representa o tempo em média que o sistema está indisponível para uso. Nesta prova de conceito, o melhor resultado para a M04 foi obtido com a utilização do sistema sem o mecanismo de adaptação (0s), uma vez que as funcionalidades são menos complexas e menos propensas a causar interrupções no sistema. Quanto aos outros dois mecanismos de adaptação, o *Model Adaptation* apresentou menor tempo de inatividade, com 0,22s.

A medida M05 (tempo de resposta de uma requisição dada uma adaptação em paralelo) mede a disponibilidade do sistema em termos de resposta a uma solicitação mesmo quando o sistema executa adaptações em paralelo, o que é um cenário comum para esse tipo de sistema. Assim como as medidas apresentadas anteriormente, os resultados apresentam valores bem próximos. Entretanto, o mecanismo *Model Evolution* apresenta um resultado um pouco

Tabela 17 – Resultados da coleta e cálculo das medidas de disponibilidade

Sub-característica	Medida	Resultados por mecanismos de adaptação		
		Sem adaptação	<i>Model Adaptation</i>	<i>Model Evolution</i>
Disponibilidade	M04	0	0,22	0,37
	M05	-	5ms	3ms
	M06	98,48%	98,58%	99,43%
	M07	-	6,35%	0,00%
	M08	-	84,21%	88,46%

melhor (3ms) que o *Model Adaptation* (5ms).

A medida M06 (taxa de requisições respondidas) considera o número total de requisições atendidas pelo sistema, também podendo ser afetada pelas operações de adaptação que podem ocorrer em paralelo. O resultado dessa medida foi melhor para o mecanismo *Model Evolution* que, mesmo com as operações de adaptação, teve uma taxa de requisições respondidas superior (99,43%) ao sistema sem mecanismos de adaptação.

Tanto para a medida M07 (taxa de interrupções causadas por adaptação) quanto para a medida M08 (taxa de adaptações corretas), o mecanismo *Model Evolution* obteve os melhores resultados, 0,00% de interrupções causadas por adaptação e 88,46% de adaptações corretas. Em relação à medida M07, as interrupções foram contabilizadas como exceções, *deadlocks* e métodos que demoraram muito para responder em decorrência da adaptação, fazendo o sistema apresentar travamentos prejudiciais ao seu funcionamento.

Considerando esses resultados, o mecanismo *Model Evolution* tem pontos promissores em relação a disponibilidade, apresentando os melhores resultados em três das quatro medidas que correspondem a essa característica.

#### 5.1.4.3 Recuperabilidade

As medidas propostas para avaliar a recuperabilidade neste trabalho são M09 (tempo médio de recuperação), M10 (taxa de recuperação de falhas operacionais) e M11 (taxa de recuperação de falhas de adaptação). Os resultados de cada medida de recuperabilidade são apresentados na Tabela 18.

Tabela 18 – Resultados da coleta e cálculo das medidas de recuperabilidade

Sub-característica	Medida	Resultados por Mecanismos de Adaptação		
		Sem adaptação	<i>Model Adaptation</i>	<i>Model Evolution</i>
Recuperabilidade	M09	-	0,3s	0,5s
	M10	0,00%	0,00%	0,00%
	M11	-	73,02%	83,33%

A medida M09 teve resultado mais positivo para o mecanismo *Model Adaptation* com uma diferença de 0,2s em relação ao *Model Evolution*. Para o cálculo dessa medida foi considerado o tempo de recuperação para falhas de adaptação, pois não foi possível detectar recuperação de falhas operacionais, o que explica os resultados obtidos com a medida M10. Isso pode acontecer pelo fato de que o sistema prevê falhas de invocação de serviço e continua a funcionar sem causar prejuízos visíveis na interface do sistema, tornando essas falhas toleráveis.

Em relação à medida M11, foram verificadas as falhas de adaptação recuperadas em tempo de execução. Essas falhas ocorrem quando os serviços que deveriam compor o novo fluxo de execução do TAS falham. Nesse momento, o processo de adaptação executa novas tentativas de invocação de serviços para se recuperar da falha e realizar a adaptação corretamente. Entretanto, nem sempre o sistema é capaz de recuperar essas falhas de adaptação. Medindo esse aspecto nos dois mecanismos de adaptação, o mecanismo *Model Evolution* apresentou uma melhor performance em relação a essa medida, com 83,33%.

#### 5.1.4.4 Tolerância a falhas

Para avaliar a sub-característica tolerância a falhas foram coletadas e calculadas as medidas M12 (taxa de requisições respondidas na presença de falha), M13 (falhas de adaptação toleradas) e M15 (nível de tolerância a falhas), e seus respectivos resultados são apresentados na Tabela 19.

A medida M12 teve o objetivo de verificar as requisições respondidas na presença de falhas e o melhor resultado obtido foi com a utilização do mecanismo *Model Adaptation* com 98,58% das requisições respondidas, enquanto o mecanismo *Model Evolution* respondeu 95,24% das requisições na presença de falhas.

A medida M13 teve a finalidade de identificar a taxa de falhas de adaptação que o sistema é capaz de tolerar e, nesse quesito, o mecanismo *Model Evolution* apresentou o melhor resultado, tolerando 100% das falhas. Entretanto, quanto ao nível de tolerância a falhas representado pela medida M14, que inclui todos os tipos de falha, o mecanismo *Model Adaptation* obteve um resultado melhor, com uma taxa de 71,43%.

#### 5.1.4.5 Robustez

Em relação à sub-característica robustez, foi definida apenas a medida M15 (taxa de adaptações corretas dado inconsistência de contexto). Nesta prova de conceito, a inconsistência de

Tabela 19 – Resultados da coleta e cálculo das medidas de tolerância a falhas

Sub-característica	Medida	Resultados por Mecanismos de Adaptação		
		Sem adaptação	<i>Model Adaptation</i>	<i>Model Evolution</i>
Tolerância a Falhas	M12	98,48%	98,58%	95,24%
	M13	-	93,65%	100%
	M14	50%	71,43%	58,82%

contexto significa diferentes informações de contexto que podem levar a adaptações concorrentes ou uma informação de contexto corrompida. Nesse sentido, para a coleta dessa medida foram acionados no sistema dois perfis de entrada para forçar uma adaptação, considerando duas entradas de contextos diferentes e adaptações foram interrompidas antes de serem completadas. Nesse cenário, o mecanismo *Model Adaptation* se mostrou mais robusto entre os mecanismos avaliados executando 88,78% das adaptações corretamente mesmo com inconsistências de contexto.

Tabela 20 – Resultados da coleta e cálculo das medidas de robustez

Sub-característica	Medida	Resultados por Mecanismos de Adaptação		
		Sem adaptação	<i>Model Adaptation</i>	<i>Model Evolution</i>
Robustez	M15	-	88,78%	78,74%

## 5.2 Discussão

A utilização das medidas na avaliação do TAS apresentou bons resultados. Todas as 15 medidas foram instrumentalizadas no código com o auxílio da biblioteca CECILIA. O uso da biblioteca facilitou o processo de instrumentalização de código, possibilitando um volume menor de inserção de código, uma vez que as funções já estão implementadas na biblioteca e no código é inserida apenas a chamada da função.

Com base no resultado das medidas, foi possível identificar pontos fortes e fracos nos mecanismos de adaptação do TAS. Por exemplo, o mecanismo de adaptação *Model Evolution* apresentou os melhores resultados em quatro de cinco medidas de disponibilidade e, em contrapartida, o mecanismo *Model Adaptation* obteve os melhores resultados em relação à sub-característica de tolerância a falhas.

De maneira geral, considerando os resultados apresentados por meio da aplicação das medidas de confiabilidade propostas neste trabalho, o sistema TAS é considerado confiável, uma vez que para a maioria das medidas os resultados obtidos estão próximos dos valores esperados, como nas medidas M02, M06, M12 e M13. Entretanto, para interpretar esses resultados com

mais assertividade é necessário a existência de *thresholds* para todas as medidas.

Além dos resultados de confiabilidade do TAS, por meio da demonstração da prova de conceito foi observado que as medidas são aplicáveis para medir a confiabilidade de sistemas autoadaptativos, bem como a biblioteca CECILIA. O uso das medidas, aliada à biblioteca, pode trazer benefícios como um guia na avaliação da confiabilidade em SAS que também pode ser utilizada de forma automatizada.

### **5.3 Ameaças à validade**

Apesar do uso de métodos e procedimentos estruturados e planejados para a realização da prova de conceito, esta pesquisa pode ter sido afetada por diferentes fatores que podem invalidar suas principais conclusões. As ameaças e as ações executadas para mitigar o impacto desses fatores são descritas de acordo com a categorização proposta por Wohlin *et al.* (2012). Na subseção 5.3.1, são apresentadas as ameaças à validade interna. Na subseção 5.3.2, são apresentadas as ameaças à validade de construção. Por fim, as ameaças externas são apresentadas na subseção 5.3.3.

#### **5.3.1 Validade interna**

De acordo com Wohlin *et al.* (2012), a validade interna se concentra em examinar relações causais, preocupando-se com validade do ambiente onde o método de estudo foi desenvolvido e a confiabilidade dos resultados. Nesse sentido, esse tipo de ameaça analisa riscos de influência de fatores nos objetos de estudo que podem ter passado despercebidos pelo pesquisador. Um fator que pode ter impactado os resultados da prova de conceito foi o tamanho do código-fonte (108677 linhas de código) do sistema selecionado e o entendimento do funcionamento do sistema. Para mitigar esse problema, o sistema foi estudado e seus principais elementos foram mapeados.

#### **5.3.2 Validade de construção**

A validade de construção se concentra em verificar a relação entre a teoria e a prática. Ela representa a diferença entre o que o pesquisador idealizou para o estudo e o que realmente foi investigado (WOHLIN *et al.*, 2012). Sendo assim, a atividade de definição de medidas (apresentada no Capítulo 4) pode ter sido afetada por essa ameaça em relação à interpretação e

documentação das medidas. Para mitigar essa ameaça foi realizada uma revisão com especialistas com o auxílio de um *checklist* (ver Seção 4.2) específico para verificação de medidas de software.

Um ponto a se destacar como uma ameaça à validade de construção é a duração das coletas das medidas durante a execução da prova de conceito. Para mitigar essa ameaça seria melhor coletar as medidas por um período maior (i.e., dias, semanas, meses). Além disso, um outro ponto de ameaça de construção é a ausência de *thresholds*, o que pode prejudicar interpretação dos resultados da medição. A definição de *thresholds* para as medidas propostas neste trabalho é apontada como trabalho futuro no Capítulo 6.

Um outro possível tópico relacionado à validade de construção do presente trabalho é a adição de código no sistema para coleta de dados. Na medida de tempo de resposta, por exemplo, são adicionadas variáveis para coletar os instantes iniciais e finais das tarefas de adaptação e invocação. Portanto, é possível que essa alteração tenha requerido do sistema mais processamento, alterando os valores dos dados coletados.

### 5.3.3 *Validade externa*

A validade externa representa a extensão que os resultados obtidos podem ser generalizados para outros cenários de investigação relevantes (WOHLIN *et al.*, 2012). Os resultados da prova de conceito em relação aos mecanismos avaliados não podem ser generalizados, uma vez que a prova de conceito fez uso de apenas um SAS e a coleta das medidas foi realizada por um período de tempo considerado curto, aproximadamente 15 minutos para cada mecanismo de adaptação. Nesse sentido, sugere-se a realização de avaliações em maiores escalas, considerando pelo menos dois sistemas diferentes e durante um período de tempo maior (i.e., dias, semanas, meses).

Entretanto, os resultados sobre a viabilidade da aplicação das medidas para auxiliar a verificação de SAS são válidos externamente, e portanto, as medidas podem ser aplicadas em outros sistemas. Para mitigar essa ameaça externa quanto à viabilidade de aplicação das medidas, foi desenvolvida a biblioteca CECILIA tornando mais prática e automatizada a coleta de medidas em código fonte de SAS desenvolvidos em Java.

## 5.4 Considerações finais

Este Capítulo apresentou uma prova de conceito utilizando as medidas de confiabilidade propostas neste trabalho, bem como a biblioteca CECILIA. Este estudo teve como objetivo investigar a aplicabilidade das medidas de confiabilidade e potenciais benefícios ao utilizá-las para avaliar a confiabilidade de SAS. A aplicação das medidas foi feita por meio de instrumentação no código do sistema e também por meio da observação do seu comportamento para a verificação de falhas na interface externa.

Nesta avaliação, as medidas foram aplicadas para verificar a confiabilidade de dois mecanismos de adaptação do sistema TAS e, em seguida, foram feitas análises comparativas em relação a cada sub-característica presente no conjunto de medidas (i.e., disponibilidade, maturidade, tolerância a falhas, recuperabilidade e robustez). Os resultados da análise apontaram que, com base nas medidas, um dos mecanismos tem boa tolerância a falhas e outro tem boa disponibilidade.

Além dos resultados das medições, também foi considerado o processo de coleta de medidas e interpretação dos resultados. Com o auxílio da biblioteca CECILIA, a coleta das medidas foi facilitada e 14 de 15 medidas foram coletadas por meio do seu uso. Uma medida (M03) foi coleta por meio de uma ferramenta de monitoramento.

Considerando que todas as 15 medidas propostas puderam ser coletadas e calculadas, os resultados deste estudo sugerem que as medidas de confiabilidade podem ser utilizadas para apoiar a avaliação da confiabilidade de sistemas autoadaptativos.

Entretanto, para uma análise mais acertada dos resultados, faz-se necessária a definição de *thresholds*, que não são apresentados no presente trabalho. Além disso, um outro ponto que pode agregar mais benefícios à avaliação é a utilização de um guia de sugestões para aumentar a confiabilidade de SAS a partir dos resultados das medições.

## 6 CONCLUSÃO

Este trabalho de dissertação de mestrado propôs um conjunto de medidas de software para avaliar a confiabilidade de sistemas autoadaptativos. Neste Capítulo, uma visão geral do trabalho desenvolvido é apresentado na Seção 6.1. Em seguida, os resultados alcançados são sumarizados na Seção 6.2, as produções bibliográficas são apresentadas na Seção 6.3, e, por fim, os trabalhos futuros são apresentados na Seção 6.4.

### 6.1 Visão Geral

Sistemas autoadaptativos são produtos de software capazes de adaptar seu comportamento em tempo de execução para atender necessidades contextuais em tempo de execução (LADDAGA; ROBERTSON, 2004). Esses sistemas são compostos por diferentes propriedades que determinam as operações que eles são capazes de realizar, como, autocura, autoproteção e autoconfiguração (SALEHIE; TAHVILDARI, 2009).

Devido à natureza de suas operações e a dinamicidade de seu comportamento, SAS são considerados sistemas complexos e são utilizados em ambientes como *healthcare*, sistemas de aviação, cidades inteligentes e sistemas de prevenção de catástrofe. Nos ambientes em que atuam, eles devem manter altos níveis de qualidade para atender contextos considerando adaptações que podem não ter sido previstas em tempo de projeto.

Tendo em vista a qualidade de sistemas autoadaptativos, neste trabalho foi, primeiro, realizado um mapeamento sistemático (MS) com o objetivo de verificar quais atributos e medidas de qualidade tem sido utilizados para avaliar esses sistemas. Os resultados do MS indicaram que diferentes atributos e medidas de qualidade têm sido utilizados para avaliar sistemas autoadaptativos, incluindo características de qualidade que não estão apresentadas na norma ISO (ISO, 2011a), como gerenciabilidade e adaptação em tempo de execução. Entre os resultados do MS, pode-se destacar também as características de confiabilidade, desempenho e manutenibilidade como as mais recorrentes e consideradas mais importantes pelos estudos para os SAS.

Com base na análise dos estudos selecionados no MS, as características que mais se destacaram foram analisadas e a característica confiabilidade foi selecionada para um estudo mais profundo, devido a ser um dos fatores de qualidade mais importantes para os SAS. A confiabilidade impacta no recebimento de informações na perspectiva do usuário, no monitoramento do desempenho que é realizado pelo próprio sistema e na qualidade dos serviços providos pelo

sistema (NETI; MULLER, 2007), (SALEHIE; TAHVILDARI, 2009). Nesse sentido, para ajudar a garantir SAS mais confiáveis avaliações de qualidade podem ser realizadas.

Uma das formas utilizadas para ajudar a garantir a qualidade de sistemas são as medições, pois elas permitem verificar o grau de atendimento do sistema em relação aos objetivos dos mesmos (SOLINGEN; BERGHOUT, 1999). Sendo assim, este trabalho teve como objetivo principal a definição de um conjunto de medidas de confiabilidade para sistemas autoadaptativos.

## **6.2 Resultados Alcançados**

O estudo proporcionou a definição de uma quantidade considerável de medidas de confiabilidade. A depender do objetivo da medição em uma determinada avaliação, pode-se optar por utilizar apenas um subconjunto das medidas propostas, ainda considerando que algumas medidas são genéricas e utilizadas em sistemas tradicionais e outras são mais específicas e levam em conta as características dos SAS.

Foi definido inicialmente um conjunto de medidas de confiabilidade direcionadas a SAS contendo 22 medidas distribuídas nas sub-características: maturidade, disponibilidade, tolerância a falhas, recuperabilidade e robustez. O conjunto de medidas é composto por medidas existentes, medidas adaptadas dos trabalhos selecionados no MS e novas medidas. O processo de definição de medidas foi executado com o auxílio do método GQM, contando com o apoio de cinco especialistas.

No processo de definição, as medidas passaram por uma revisão com especialistas e o conjunto foi reduzido para 15 medidas de confiabilidade. Em seguida, essas medidas foram avaliadas por meio de uma prova de conceito. Na prova de conceito, as medidas foram utilizadas para avaliar um SAS sem realizar adaptações e usando dois mecanismos de adaptação diferentes. Os resultados da prova de conceito sugerem que as medidas podem ser utilizadas para avaliar sistemas autoadaptativos. Além disso, a biblioteca CECILIA, desenvolvida também neste trabalho com o objetivo de facilitar a coleta dessas medidas, pode ser utilizada em conjunto.

Os resultados do mapeamento sistemático realizado no presente trabalho fornecem um panorama geral sobre como são executadas as avaliações de sistemas autoadaptativos, bem como apresenta características, sub-características, atributos e medidas de qualidade mais utilizados nessas avaliações. A partir desses dados, tem-se uma visão geral atual sobre a avaliação da qualidade em sistemas autoadaptativos, permitindo que os leitores pesquisadores deste estudo identifiquem lacunas de pesquisa, melhorias de soluções existentes e tendências na área em

questão.

As medidas definidas, bem como a biblioteca, podem ser utilizadas como guia na avaliação de confiabilidade de sistemas autoadaptativos, de forma isolada ou combinada com outras medidas de qualidade, técnicas ou ferramentas. Seus resultados são capazes de indicar o nível de confiabilidade de vários aspectos do sistema, considerando a perspectiva de atributos presentes em todos os tipos de sistema (e.g., falhas, tempo de recuperação, disponibilidade) e atributos específicos do domínio de SAS (e.g., tempo de adaptação, falhas de adaptação, recuperação de falhas na adaptação).

De forma geral, espera-se que o presente trabalho possa ser utilizado como um guia para a avaliação da confiabilidade em sistemas autoadaptativos, dando suporte a avaliação da qualidade dos mesmos. Além disso, as medidas podem ser utilizadas para comparar a confiabilidade de sistemas com diferentes estratégias de adaptação.

### 6.3 Produções Bibliográficas

Nesta seção, são apresentados os trabalhos publicados pela autora desta dissertação durante o programa de mestrado, bem como a contribuição e conhecimentos adquiridos durante a elaboração de cada trabalho que puderam ser aplicados no desenvolvimento da presente dissertação.

- **De Sousa, Amanda Oliveira;** De Sousa Santos, Ismayle; Aragão, Bruno Sabóia; Andrade, Rossana M. C.. Towards an automatic approach to estimating test effort. In: the 17th Brazilian Symposium, 2018, Curitiba. Proceedings of the 17th Brazilian Symposium on Software Quality - SBQS. New York: ACM Press, 2018. p. 305.
- A autora da presente dissertação de mestrado contribuiu na produção do artigo “Towards an automatic approach to estimating test effort.” nas atividades de escrita, implementação de uma ferramenta na linguagem JAVA para cálculo do esforço de testes funcionais e coleta de medidas de esforço de testes. O desenvolvimento dessa pesquisa contribuiu para a prática da escrita científica e para a aquisição de conhecimentos em coleta e análise de medidas contribuindo diretamente no desenvolvimento do presente trabalho.
- **Sousa, Amanda;** Uchôa, Anderson; Fernandes, Eduardo; Bezerra, Carla I. M.; Andrade, Rossana M. C.. REM4DSPL: A Requirements Engineering Method for DSPL. In: the 18th Brazilian Symposium, 2019, Curitiba. Proceedings of the 17th Brazilian Symposium

- on Software Quality - SBQS. New York: ACM Press, 2019. (em processo de publicação)
- A autora da presente dissertação de mestrado contribuiu na produção do artigo “REM4DSPL: A Requirements Engineering Method for DSPL.” nas atividades de escrita, elaboração e avaliação da solução. O desenvolvimento dessa pesquisa contribuiu para a prática da escrita científica utilizada nesta dissertação.
  - **De Sousa, Amanda Oliveira; Bezerra, C. I. M.; Monteiro Filho, J. M. S; Andrade, Rossana M. C..** Avaliação da Qualidade de Reconfigurações Dinâmicas em Sistemas Autoadaptativos. In: XVII Simpósio Brasileiro de Qualidade de Software - Workshop de Teses e Dissertações, 2018, Curitiba. SBQS Proceedings of the 17th Brazilian Symposium on Software Quality, 2018.
    - Esse trabalho constitui os resultados iniciais desta dissertação relacionados a fase de proposta da dissertação e apresenta os primeiros resultados do MS. A produção e apresentação desse trabalho contribuiu para o amadurecimento do presente trabalho em razão dos *feedbacks* da banca avaliadora do SBQS.
  - **De Sousa, Amanda Oliveira; Bezerra, Carla I. M.; Andrade, Rossana M. C.; Filho, José M. S. M..** Quality Evaluation of Self-Adaptive Systems. In: the XXXIII Brazilian Symposium, 2019, Salvador. Proceedings of the XXXIII Brazilian Symposium on Software Engineering - SBES 2019. New York: ACM Press, 2019. p. 213.
    - Esse trabalho constitui uma parte dos resultados do MS realizado nesta dissertação de mestrado e apresenta desafios relacionados a avaliação da qualidade de SAS nos últimos cinco anos.

#### 6.4 Trabalhos Futuros

Os tópicos de trabalhos futuros que podem ser gerados a partir desta dissertação bem como tópicos para aperfeiçoamento do presente trabalho são descritos a seguir:

- **Definição de novas medidas para outras características.** Ainda existem muitas características inexploradas no que diz respeito à medição de SAS. Mesmo as características que são mais avaliadas em estudos científicos não apresentam um conjunto de medidas bem definidas para apoiar a avaliação da qualidade de SAS, como eficiência de desempenho e manutibilidade.
- **Definição de *thresholds* para as medidas definidas.** Um outro ponto importante que seria também um aprimoramento deste trabalho seria a definição de *thresholds* para as medidas

de confiabilidade. Esses valores seriam muito importantes para guiar a interpretação dos resultados das medições. Uma forma de definir os *thresholds* pode ser por meio da criação de *datasets* combinada a utilização de técnicas de aprendizagem de máquina. Vale ressaltar que a definição de *thresholds* de forma genérica pode ser complicada, fatores que impactam nas medidas coletadas devem ser verificados e estudados para aumentar a precisão da medição e da interpretação dos resultados. Dessa forma, recomenda-se a participação de especialistas do sistema sob avaliação para discutir e definir quais variáveis (além dos elementos de medição) devem ser monitoradas. Um outro tipo de desafio é que os mesmos *thresholds* dificilmente poderão ser utilizados para interpretar medições em sistemas diferentes, uma vez que os sistemas podem apresentar diferentes requisitos não funcionais e diferentes estratégias de adaptação. Logo, um resultado ótimo para um sistema A é diferente do resultado ótimo para o sistema B em relação a mesma medida de qualidade.

- **Avaliações mais robustas com outros sistemas.** Outro ponto de aperfeiçoamento deste trabalho é a utilização das medidas de confiabilidade em avaliações mais robustas com mais de um sistema, visando a generalização de resultados.
- **Definição de novas características.** Como comentado nos resultados do MS, existem trabalhos que apresentam características específicas para SAS, entretanto não existe uma definição clara da característica ou em que níveis ela pode ser avaliada. Considerando a dinamicidade das operações dos SAS e suas propriedades particulares faz-se necessário que características de qualidade específicas como a gerenciabilidade (MUNNELLY; CLARKE, 2008) e adaptação em tempo de execução (PESSOA *et al.*, 2017) sejam definidas e avaliadas.
- **Definição de índices globais.** A partir das medidas definidas no presente trabalho, pode-se verificar a possibilidade de combinação de todos os valores obtidos com a aplicação das medidas, resultando em índice global que pode indicar o grau de confiabilidade do sistema sob avaliação. Na mesma linha, também existe a possibilidade de conduzir estudos sobre a combinação de subconjuntos de medidas para obter índices mais amplos sobre aspectos de confiabilidade, como obter um índice de todas as medidas por sub-características, por exemplo, ou até combinar resultados de sub-características diferentes para um índice mais macro. Para esse tipo de estudo, deve-se considerar distribuições de pesos para as medidas utilizadas de acordo com os atributos de qualidade mais importantes para o sistema.

- **Construção de um guia de sugestões para aumentar a confiabilidade dos SAS.** Após a execução das medições, os resultados são capazes de indicar problemas e pontos de melhoria no sistema avaliado. A identificação de problemas de confiabilidade no sistema pode levar a um conjunto de ações para resolver os problemas encontrados e dessa forma, aumentar os níveis de confiabilidade. Entretanto, em se tratando de SAS, as ações devem ser cuidadosamente planejadas para não ocasionar novos problemas e portanto, deve-se considerar atacar os problemas encontrados de forma sistematizada. Nesse sentido, a construção de um guia de sugestões para aumentar a confiabilidade dos SAS se faz necessário.
- **Investigação sobre impactos no consumo de energia.** No presente trabalho não são considerados fatores como consumo de energia ou consumo sustentável de energia em dispositivos. Nesse sentido, pode-se desenvolver um estudo sobre como a priorização ou o aumento de atributos de qualidade pode impactar no consumo de energia de dispositivos de sistemas autoadaptativos.

## REFERÊNCIAS

- ADJOYAN, S.; SERIAI, A.-D. Reconfigurable service-based architecture based on variability description. In: **ACM. Proceedings of the Symposium on Applied Computing**. Marrakech, Morocco, 2017. p. 1154–1161.
- BARCELLOS, M. P.; ROCHA, A. R.; FALBO, R. de A. Evaluating the suitability of a measurement repository for statistical process control. In: **ACM. Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement**. New York, NY, USA, 2010. p. 27.
- BARESI, L.; BIANCULLI, D.; GHEZZI, C.; GUINEA, S.; SPOLETINI, P. Validation of web service compositions. **IET software**, IET, v. 1, n. 6, p. 219–232, 2007.
- BELL, J.; BELLEGARDE, F.; HOOK, J.; KIEBURTZ, R. B.; KOTOV, A.; LEWIS, J.; MCKINNEY, L.; OLIVA, D. P.; SHEARD, T.; TONG, L.; WALTON, L.; ZHOU, T. Software design for reliability and reuse: A proof-of-concept demonstration. In: **Proceedings of the Conference on TRI-Ada '94**. New York, NY, USA: ACM, 1994. (TRI-Ada '94), p. 396–404. ISBN 0-89791-666-2. Disponível em: <http://doi.acm.org/10.1145/197694.197740>. Acesso em: 5 set. 2019.
- BENCOMO, N.; BLAIR, G. S.; FLORES-CORTÉS, C. A.; SAWYER, P. Reflective component-based technologies to support dynamic variability. In: **CITeseer. VaMoS**. Germany, 2008. p. 141–150.
- BENCOMO, N.; HALLSTEINSEN, S.; ALMEIDA, E. S. D. A view of the dynamic software product line landscape. **Computer**, IEEE, v. 45, n. 10, p. 36–41, 2012.
- BENCOMO, N.; SAWYER, P.; BLAIR, G. S.; GRACE, P. Dynamically adaptive systems are product lines too: Using model-driven techniques to capture dynamic variability of adaptive systems. In: **SPLC (2)**. Limerick, Ireland: Lero Int. Science Centre, University of Limerick, Ireland, 2008. p. 23–32.
- BEZERRA, C. I.; BARBOSA, J.; FREIRES, J. H.; ANDRADE, R. M. C.; MONTEIRO, J. M. Dymmer: a measurement-based tool to support quality evaluation of dspl feature models. In: **ACM. Proceedings of the 20th International Systems and Software Product Line Conference**. New York, NY, USA: Association for Computing Machinery, 2016. p. 314–317. ISBN 9781450340502.
- BEZERRA, C. I. M.; ANDRADE, R. M. C.; MONTEIRO, J. M. S.; CEDRAZ, D. Aggregating measures using fuzzy logic for evaluating feature models. In: **Proceedings of the 12th International Workshop on Variability Modelling of Software-Intensive Systems**. New York, NY, USA: ACM, 2018. (VAMOS 2018), p. 35–42. ISBN 978-1-4503-5398-4. Disponível em: <http://doi.acm.org/10.1145/3168365.3168375>. Acesso em: 21 set. 2019.
- BRUN, Y.; SERUGENDO, G. D. M.; GACEK, C.; GIESE, H.; KIENLE, H.; LITOIU, M.; MÜLLER, H.; PEZZÈ, M.; SHAW, M. Engineering self-adaptive systems through feedback loops. In: \_\_\_\_\_. **Software Engineering for Self-Adaptive Systems**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 48–70. ISBN 978-3-642-02161-9. Disponível em: [https://doi.org/10.1007/978-3-642-02161-9\\_3](https://doi.org/10.1007/978-3-642-02161-9_3). Acesso em: 5 set. 2019.

CAPILLA, R.; BOSCH, J.; TRINIDAD, P.; RUIZ-CORTÉS, A.; HINCHEY, M. An overview of dynamic software product line architectures and techniques: Observations from research and industry. **Journal of Systems and Software**, Elsevier, v. 91, p. 3–23, 2014.

CAPILLA, R.; VALDEZATE, A.; DÍAZ, F. J. A runtime variability mechanism based on supertypes. In: IEEE. **Foundations and Applications of Self\* Systems, IEEE International Workshops on**. Augsburg, Germany, 2016. p. 6–11.

Card, D. N. Practical software measurement. In: **25th International Conference on Software Engineering, 2003. Proceedings**. Portland, OR, USA: IEEE, 2003. p. 738–739.

CARVALHO, R. M.; ANDRADE, R. M. C.; OLIVEIRA, K. M. Aquarium-a suite of software measures for hci quality evaluation of ubiquitous mobile applications. **Journal of Systems and Software**, Elsevier, v. 136, p. 101–136, 2018.

CETINA, C.; GINER, P.; FONS, J.; PELECHANO, V. Prototyping dynamic software product lines to evaluate run-time reconfigurations. **Science of Computer Programming**, v. 78, n. 12, p. 2399 – 2413, 2013. ISSN 0167-6423. Special Section on International Software Product Line Conference 2010 and Fundamentals of Software Engineering (selected papers of FSEN 2011). Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167642312001189>. Acesso em: 10 set. 2019.

CHEN, B.; PENG, X.; ZHAO, W. Towards runtime optimization of software quality based on feedback control theory. In: . New York, NY, USA: Association for Computing Machinery, 2009. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-72249100588&doi=10.1145%2f1640206.1640216&partnerID=40&md5=06ede331d0b1e5ba5bc13229afdce5c6>. Acesso em: 10 set. 2019.

CHENG, B. H. C.; LEMOS, R. de; GIESE, H.; INVERARDI, P.; MAGEE, J.; ANDERSSON, J.; BECKER, B.; BENCOMO, N.; BRUN, Y.; CUKIC, B.; SERUGENDO, G. D. M.; DUSTDAR, S.; FINKELSTEIN, A.; GACEK, C.; GEIHS, K.; GRASSI, V.; KARSAI, G.; KIENLE, H. M.; KRAMER, J.; LITOIU, M.; MALEK, S.; MIRANDOLA, R.; MÜLLER, H. A.; PARK, S.; SHAW, M.; TICHY, M.; TIVOLI, M.; WEYNS, D.; WHITTLE, J. Software engineering for self-adaptive systems: A research roadmap. In: \_\_\_\_\_. **Software Engineering for Self-Adaptive Systems**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 1–26. ISBN 978-3-642-02161-9. Disponível em: [https://doi.org/10.1007/978-3-642-02161-9\\_1](https://doi.org/10.1007/978-3-642-02161-9_1). Acesso em: 5 set. 2019.

CHENG, S.; GARLAN, D.; SCHMERL, B. Evaluating the effectiveness of the rainbow self-adaptive system. In: **2009 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems**. Vancouver, BC, Canada: IEEE, 2009. p. 132–141. ISSN 2157-2305.

CRIADO, J.; MARTÍNEZ-FERNÁNDEZ, S.; AMELLER, D.; IRIBARNE, L.; PADILLA, N. Exploring quality-aware architectural transformations at run-time: The enia case. In: BELLATRECHE, L.; PASTOR, Ó.; JIMÉNEZ, J. M. A.; AÏT-AMEUR, Y. (Ed.). **Model and Data Engineering**. Cham: Springer International Publishing, 2016. p. 288–302. ISBN 978-3-319-45547-1.

Dos Reis Fontes, R.; Campolo, C.; Esteve Rothenberg, C.; Molinaro, A. From theory to experimental evaluation: Resource management in software-defined vehicular networks. **IEEE Access**, v. 5, p. 3069–3076, 2017.

Dyba, T.; Dingsoyr, T.; Hanssen, G. K. Applying systematic reviews to diverse study types: An experience report. In: **First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)**. Madrid, Spain: IEEE, 2007. p. 225–234.

FARAHANI, A.; NAZEMI, E.; CABRI, G.; RAFIZADEH, A. An evaluation method for self-adaptive systems. In: **2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016 - Conference Proceedings**. Budapest, Hungary: IEEE, 2017. p. 2814–2820. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85015746818&doi=10.1109%2fSMC.2016.7844665&partnerID=40&md5=4abad7ccee83ce43be511a92c4c2d2b8>. Acesso em: 10 set. 2019.

FILIERI, A.; GHEZZI, C.; MIRANDOLA, R.; TAMBURRELLI, G. Conquering complexity via seamless integration of design-time and run-time verification. In: **Conquering Complexity**. London: Springer, 2012. p. 253–275.

FRANCO, J. M.; CORREIA, F.; BARBOSA, R.; ZENHA-RELA, M.; SCHMERL, B.; GARLAN, D. Improving self-adaptation planning through software architecture-based stochastic modeling. **Journal of Systems and Software**, v. 115, p. 42 – 60, 2016. ISSN 0164-1212. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0164121216000212>. Acesso em: 10 set. 2019.

GANEK, A. G.; CORBI, T. A. The dawning of the autonomic computing era. **IBM Systems Journal**, v. 42, n. 1, p. 5–18, 2003. ISSN 0018-8670.

GARCÉS, L.; AMPATZOGLOU, A.; AVGERIOU, P.; NAKAGAWA, E. Y. Quality attributes and quality models for ambient assisted living software systems: A systematic mapping. **Information and Software Technology**, Elsevier, v. 82, p. 121–138, 2017.

GHEZZI, C.; SHARIFLOO, A. M. Dealing with non-functional requirements for adaptive systems via dynamic software product-lines. In: \_\_\_\_\_. **Software Engineering for Self-Adaptive Systems II: International Seminar, Dagstuhl Castle, Germany, October 24-29, 2010 Revised Selected and Invited Papers**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 191–213. ISBN 978-3-642-35813-5. Disponível em: [https://doi.org/10.1007/978-3-642-35813-5\\_8](https://doi.org/10.1007/978-3-642-35813-5_8). Acesso em: 10 set. 2019.

GLASS, R.; VESSEY, I.; RAMESH, V. Research in software engineering: an analysis of the literature. **Information and Software Technology**, v. 44, n. 8, p. 491 – 506, 2002. ISSN 0950-5849. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0950584902000496>. Acesso em: 21 set. 2019.

GOLDSTEEN, A.; MOFFIE, M.; BANDYSZAK, T.; MOHAMMADI, N.; CHEN, X.; MEICHANETZOGLOU, S.; IOANNIDIS, S.; CHATZIADAM, P. A tool for monitoring and maintaining system trustworthiness at runtime. In: **Workshops at 21st International Conference on Requirements Engineering: Foundation for Software Quality (23/03/15 - 26/03/15)**. Essen, Germany: CEUR-WS.org, 2015. v. 1342, p. 142–147. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84925949673&partnerID=40&md5=6429a435bbaece4171f0b23c8164482b>. Acesso em: 10 set. 2019.

GRAMBOW, G.; OBERHAUSER, R. Towards automated context-aware software quality management. In: **2010 Fifth International Conference on Software Engineering Advances**. Nice, France: IEEE, 2010. p. 347–352.

- GRASSI, V.; MIRANDOLA, R.; RANDAZZO, E. Model-driven assessment of qos-aware self-adaptation. In: \_\_\_\_\_. **Software Engineering for Self-Adaptive Systems**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 201–222. ISBN 978-3-642-02161-9. Disponível em: [https://doi.org/10.1007/978-3-642-02161-9\\_11](https://doi.org/10.1007/978-3-642-02161-9_11). Acesso em: 10 set. 2019.
- GURP, J. V.; BOSCH, J.; SVAHNBERG, M. On the notion of variability in software product lines. In: IEEE. **Software Architecture, 2001. Proceedings. Working IEEE/IFIP Conference on**. Amsterdam, The NETHERLANDS, 2001. p. 45–54.
- IEEE. Ieee standard glossary of software engineering terminology. **IEEE Std 610.12-1990**, p. 1–84, Dec 1990.
- ISO. Iec 25010: 2011 systems and software engineering—systems and software quality requirements and evaluation (square)—system and software quality models. **International Organization for Standardization**, v. 34, p. 2910, 2011.
- ISO. Systems and software engineering – systems and software quality requirements and evaluation (square) – measurement of system and software product quality. **International Organization for Standardization**, p. 54, 2011.
- KIM, M. A quality model for evaluating iot applications. **International Journal of Computer and Electrical Engineering**, IACSIT Press, v. 8, n. 1, p. 66, 2016.
- KITCHENHAM, B.; CHARTERS, S. Guidelines for performing systematic literature reviews in software engineering. v. 2, 01 2007.
- KOSCIANSKI, A.; SOARES, M. dos S. **Qualidade de Software-2ª Edição: Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software**. Brasil: Novatec Editora, 2007.
- LADDAGA, R.; ROBERTSON, P. Self adaptive software: A position paper. In: **SELF-STAR: International Workshop on Self-\* Properties in Complex Information Systems**. Bertinoro, Italy: Springer, 2004. v. 31, p. 19.
- Latre, S.; Leroux, P.; Coenen, T.; Braem, B.; Ballon, P.; Demeester, P. City of things: An integrated and multi-technology testbed for iot smart city experiments. In: **2016 IEEE International Smart Cities Conference (ISC2)**. Trento, Italy: IEEE, 2016. p. 1–8.
- LOUKIL, S.; KALLEL, S.; RODRIGUEZ, I. B.; JMAIEL, M. Monitoring of quality of service in dynamically adaptive systems. In: BOUCHACHIA, A. (Ed.). **Adaptive and Intelligent Systems**. Cham: Springer International Publishing, 2014. p. 121–130. ISBN 978-3-319-11298-5.
- MAIA, M. E. F.; ANDRADE, R. M. C.; VIANA, W. Towards a component infrastructure for cyber-physical systems. In: **Proceedings of the 31st Annual ACM Symposium on Applied Computing**. New York, NY, USA: ACM, 2016. (SAC '16), p. 626–628. ISBN 978-1-4503-3739-7. Disponível em: <http://doi.acm.org/10.1145/2851613.2851934>. Acesso em: 21 set. 2019.
- MANTORO, T. Metrics evaluation for context-aware computing. In: **Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia**. New York, NY, USA: Association for Computing Machinery, 2009. p. 574–578. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-77955111665&doi=10.1145/>

2f1821748.1821859&partnerID=40&md5=9b7a5583c5a2a9d812799322220e0ef8. Acesso em: 10 set. 2019.

MARCOS, M.; ESTÉVEZ, E.; JOUVRAY, C.; KUNG, A. An approach to use mde in dynamically reconfigurable networked embedded soas. **IFAC Proceedings Volumes**, v. 44, n. 1, p. 14946 – 14951, 2011. ISSN 1474-6670. 18th IFAC World Congress. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1474667016460313>. Acesso em: 10 set. 2019.

MARZOLLA, M.; MIRANDOLA, R. Performance aware reconfiguration of software systems. In: ALDINI, A.; BERNARDO, M.; BONONI, L.; CORTELLESSA, V. (Ed.). **Computer Performance Engineering**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 51–66. ISBN 978-3-642-15784-4.

MING, H.; OYAMA, K.; CHANG, C. K. Human-intention driven self adaptive software evolvability in distributed service environments. In: **2008 12th IEEE International Workshop on Future Trends of Distributed Computing Systems**. Kunming, China: IEEE, 2008. p. 51–57. ISSN 1071-0485.

MIRANDOLA, R.; POTENA, P. Self-adaptation of service based systems based on cost/quality attributes tradeoffs. In: **2010 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing**. [S. l.: s. n.], 2010. p. 493–501.

MUNNELLY, J.; CLARKE, S. Infrastructure for ubiquitous computing: Improving quality with modularisation. In: . [S. n.], 2008. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-65249103860&doi=10.1145%2f1404891.1404898&partnerID=40&md5=54dcf31862e5ee76c2516a3ec746bca4>. Acesso em: 10 set. 2019.

MÄRTIN, L.; NICOLAI, A. Towards self-reconfiguration of space systems on architectural level based on qualitative ratings. In: **2014 IEEE Aerospace Conference**. [S. l.: s. n.], 2014. p. 1–8. ISSN 1095-323X.

NETI, S.; MULLER, H. A. Quality criteria and an analysis framework for self-healing systems. In: **International Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS '07)**. Minneapolis, MN, USA,: IEEE, 2007. p. 6–6. ISSN 2157-2305.

PAI, M.; MCCULLOCH, M.; GORMAN, J. D.; PAI, N.; ENANORIA, W.; KENNEDY, G.; THARYAN, P.; COLFORD, J. M. Systematic reviews and meta-analyses: an illustrated, step-by-step guide. **The National medical journal of India**, v. 17, n. 2, p. 86—95, 2004. ISSN 0970-258X. Disponível em: <http://europepmc.org/abstract/MED/15141602>. Acesso em: 6 set. 2019.

PARK, R. E.; GOETHERT, W. B.; FLORAC, W. A. **Goal-Driven Software Measurement. A Guidebook**. Pittsburgh, PA, 1996.

PAUCAR, L.; BENCOMO, N. The reassessment of preferences of non-functional requirements for better informed decision-making in self-adaptation. In: . [S. n.], 2017. p. 32–38. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85013104380&doi=10.1109%2fREW.2016.38&partnerID=40&md5=9ef7da861a76581d546e1253fd588afb>. Acesso em: 10 set. 2019.

PEREZ, B.; CORREAL, D. A model driven approach to the analysis of quality scenarios within self-adaptable soa systems. **Electronic Notes in Theoretical Computer Science**, v. 281, p. 113 –

126, 2011. ISSN 1571-0661. Proceedings of the 2011 Latin American Conference in Informatics (CLEI). Disponível em: <http://www.sciencedirect.com/science/article/pii/S1571066111001782>. Acesso em: 10 set. 2019.

PEREZ-PALACIN, D.; MIRANDOLA, R.; MERSEGUER, J. Software architecture adaptability metrics for qos-based self-adaptation. In: . [S. n.], 2011. p. 171–175. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-79960528311&doi=10.1145%2f2000259.2000288&partnerID=40&md5=c786e73bf93fba784406193c31459e77>. Acesso em: 10 set. 2019.

PEREZ-PALACIN, D.; MIRANDOLA, R.; MERSEGUER, J. On the relationships between qos and software adaptability at the architectural level. **Journal of Systems and Software**, v. 87, p. 1 – 17, 2014. ISSN 0164-1212. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0164121213001982>. Acesso em: 10 set. 2019.

PESSOA, L.; FERNANDES, P.; CASTRO, T.; ALVES, V.; RODRIGUES, G. N.; CARVALHO, H. Building reliable and maintainable dynamic software product lines: An investigation in the body sensor network domain. **Information and Software Technology**, v. 86, p. 54 – 70, 2017. ISSN 0950-5849. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0950584917301386>. Acesso em: 10 set. 2019.

PETERSEN, K.; FELDT, R.; MUJTABA, S.; MATTSSON, M. Systematic mapping studies in software engineering. In: . Swindon, GBR: BCS Learning I& Development Ltd., 2008. v. 8, p. 68–77.

POTENA, P. Optimization of adaptation plans for a service-oriented architecture with cost, reliability, availability and performance tradeoff. **Journal of Systems and Software**, v. 86, n. 3, p. 624 – 648, 2013. ISSN 0164-1212. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0164121212003226>. Acesso em: 10 set. 2019.

RAIBULET, C. Hints on quality evaluation of self-systems. In: **2014 IEEE Eighth International Conference on Self-Adaptive and Self-Organizing Systems**. London, UK: IEEE, 2014. p. 185–186. ISSN 1949-3673.

RAIBULET, C.; FONTANA, F. A.; CAPILLA, R.; CARRILLO, C. An overview on quality evaluation of self-adaptive systems. **Managing Trade-offs in Adaptable Software Architectures**, Elsevier, 2016.

RAIBULET, C.; FONTANA, F. A.; CAPILLA, R.; CARRILLO, C. Chapter 13 - an overview on quality evaluation of self-adaptive systems. In: MISTRICK, I.; ALI, N.; KAZMAN, R.; GRUNDY, J.; SCHMERL, B. (Ed.). **Managing Trade-Offs in Adaptable Software Architectures**. Boston: Morgan Kaufmann, 2017. p. 325 – 352. ISBN 978-0-12-802855-1. Disponível em: <http://www.sciencedirect.com/science/article/pii/B9780128028551000137>. Acesso em: 10 set. 2019.

RAIBULET, C.; MASCIADRI, L. Evaluation of dynamic adaptivity through metrics: an achievable target? In: **2009 Joint Working IEEE/IFIP Conference on Software Architecture European Conference on Software Architecture**. Cambridge, UK: IEEE, 2009. p. 341–344.

RAIBULET, C.; MASCIADRI, L. Towards evaluation mechanisms for runtime adaptivity: From case studies to metrics. In: **2009 Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns**. Athens, Greece: IEEE, 2009. p. 146–152.

RYAN, C.; ROSSI, P. Software, performance and resource utilisation metrics for context-aware mobile applications. In: **11th IEEE International Software Metrics Symposium (METRICS'05)**. [S. l.: s. n.], 2005. p. 10 pp.–12. ISSN 1530-1435.

SABATUCCI, L.; SEIDITA, V.; COSSENTINO, M. The four types of self-adaptive systems: A metamodel. **Smart Innovation, Systems and Technologies**, v. 76, p. 440–450, 2018. Disponível em: [https://www.scopus.com/inward/record.uri?eid=2-s2.0-85020384219&doi=10.1007%2f978-3-319-59480-4\\_44&partnerID=40&md5=28103bebecc3e0f4044fd4bfe7fcb729](https://www.scopus.com/inward/record.uri?eid=2-s2.0-85020384219&doi=10.1007%2f978-3-319-59480-4_44&partnerID=40&md5=28103bebecc3e0f4044fd4bfe7fcb729). Acesso em: 5 set. 2019.

SALEHIE, M.; TAHVILDARI, L. Self-adaptive software: Landscape and research challenges. **ACM transactions on autonomous and adaptive systems (TAAS)**, ACM, v. 4, n. 2, p. 14, 2009.

SANCHEZ, L.; DIAZ-PACE, J.; ZUNINO, A.; MOISAN, S.; RIGAULT, J.-P. An approach for managing quality attributes at runtime using feature models. In: **2014 Eighth Brazilian Symposium on Software Components, Architectures and Reuse**. Maceió, AL: IEEE, 2014. p. 11–20.

SANCHEZ, L. E.; DIAZ-PACE, J. A.; ZUNINO, A.; MOISAN, S.; RIGAULT, J.-P. An approach based on feature models and quality criteria for adapting component-based systems. **Journal of Software Engineering Research and Development**, v. 3, n. 1, p. 10, Jun 2015. ISSN 2195-1721. Disponível em: <https://doi.org/10.1186/s40411-015-0022-1>. Acesso em: 10 set. 2019.

SANISLAV, T.; MOIS, G.; MICLEA, L. An approach to model dependability of cyber-physical systems. **Microprocessors and Microsystems**, v. 41, p. 67 – 76, 2016. ISSN 0141-9331. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0141933115002057>. Acesso em: 10 set. 2019.

SEN, S.; ALESIO, S. D.; MARIJAN, D.; SARKAR, A. Evaluating reconfiguration impact in self-adaptive systems—an approach based on combinatorial interaction testing. In: **IEEE Software Engineering and Advanced Applications (SEAA), 2015 41st Euromicro Conference on**. [S. l.], 2015. p. 250–254.

SERRAL, E.; SERNANI, P.; DALPIAZ, F. Personalized adaptation in pervasive systems via non-functional requirements. **Journal of Ambient Intelligence and Humanized Computing**, v. 9, n. 6, p. 1729–1743, Nov 2018. ISSN 1868-5145. Disponível em: <https://doi.org/10.1007/s12652-017-0611-4>. Acesso em: 10 set. 2019.

SOLINGEN, R.; BERGHOUT, E. The goal/question/metric method: A practical guide for quality improvement of software development. 01 1999.

SOUSA, A. O. de; BEZERRA, C. I. M.; ANDRADE, R. M. C.; FILHO, J. M. S. M. Quality evaluation of self-adaptive systems: Challenges and opportunities. In: **Proceedings of the XXXIII Brazilian Symposium on Software Engineering**. New York, NY, USA: ACM, 2019. (SBES 2019), p. 213–218. ISBN 978-1-4503-7651-8. Disponível em: <http://doi.acm.org/10.1145/3350768.3352455>. Acesso em: 14 nov. 2019.

VILLEGAS, N.; MÜLLER, H.; TAMURA, G.; DUCHIEN, L.; CASALLAS, R. A framework for evaluating quality-driven self-adaptive software systems. In: . [S. n.], 2011. p. 80–89. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-79959550156&doi=>

10.1145%2f1988008.1988020&partnerID=40&md5=87328f5539e05de7c67c753109f03509.  
Acesso em: 10 set. 2019.

WEYNS, D.; CALINESCU, R. Tele assistance: a self-adaptive service-based system exemplar. In: IEEE PRESS. **Proceedings of the 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems**. [S. l.], 2015. p. 88–92.

WIEGERS, K.; BEATTY, J. **Software requirements**. New York City, New York: Pearson Education, 2013.

WOHLIN, C.; RUNESON, P.; HÖST, M.; OHLSSON, M. C.; REGNELL, B.; WESSLÉN, A. **Experimentation in software engineering**. Berlin, Heidelberg: Springer Science & Business Media, 2012.

YANG, J.; HUANG, G.; ZHU, W.; CUI, X.; MEI, H. Quality attribute tradeoff through adaptive architectures at runtime. **Journal of Systems and Software**, v. 82, n. 2, p. 319 – 332, 2009. ISSN 0164-1212. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0164121208001696>. Acesso em: 10 set. 2019.

ZHU, Y.; HUANG, G.; MEI, H. Quality attribute scenario based architectural modeling for self-adaptation supported by architecture-based reflective middleware. In: **11th Asia-Pacific Software Engineering Conference**. Busan, South Korea: IEEE, 2004. p. 2–9. ISSN 1530-1362.

## APÊNDICE A – PLANILHAS GQM DO PARTICIPANTE 1

<b>Objeto</b>	<b>Propósito</b>	<b>Foco de qualidade</b>	<b>Ponto de vista</b>
<i>Sistemas autoadaptativos</i>	<i>Avaliar</i>	<i>Confiabilidade (Sem sub característica)</i>	<i>Usuário final e desenvolvedor</i>
<b>Foco de qualidade</b>		<b>Fatores de variação</b>	
#Adaptação - tempo - corretude da adaptação #Consumo de energia		#Funcionamento do sistema como um todo #Desempenho	
<b>Hipótese de base (estimativas)</b>		<b>Impacto dos fatores de variação na hipótese de base</b>	
#Adaptação(Tempo) - Quanto menor, melhor #Adaptação (Corretude) - verificar se as funcionalidades atendem o contexto (SIM ou NÃO) - conclusão da tarefa: o usuário conseguiu concluir a tarefa? (SIM ou NÃO) - o sistema está consumindo a quantidade correta de recursos? (SIM ou NÃO) - existe alguma quebra de funcionalidade no sistema? (SIM ou NÃO)		#Funcionamento do sistema é afetado se a adaptação do sistema demorar para ser completada #O desempenho deve aumentar e diminuir de acordo com o exigido pela tarefa e/ou adaptação que o sistema deverá executar	

<b>Objeto</b>	<b>Propósito</b>	<b>Foco de qualidade</b>	<b>Ponto de vista</b>
<i>Sistemas autoadaptativos</i>	<i>Avaliar</i>	<i>Disponibilidade</i>	<i>Usuário final e desenvolvedor</i>
<b>Foco de qualidade</b>		<b>Fatores de variação</b>	
<p>#Tempo</p> <ul style="list-style-type: none"> <li>- Tempo que o sistema fica disponível para uso</li> <li>- Tempo que o sistema fica indisponível para uso</li> <li>- Tempo médio para falhar</li> </ul> <p>#Falhas</p> <ul style="list-style-type: none"> <li>- Número de falhas durante o uso do sistema</li> </ul>		<p>#Atributo de qualidade</p> <ul style="list-style-type: none"> <li>- Desempenho</li> </ul>	
<b>Hipótese de base (estimativas)</b>		<b>Impacto dos fatores de variação na hipótese de base</b>	
<p>#Estimativa para o tempo que o sistema fica disponível</p> <ul style="list-style-type: none"> <li>- Disponibilidade alta: &gt;99% do tempo</li> <li>- Disponibilidade media: entre 99% e 95% do tempo</li> <li>- Disponibilidade baixa: &lt;95% do tempo</li> </ul> <p>#Estimativa para o tempo em que o sistema fica indisponível</p> <ul style="list-style-type: none"> <li>- Quanto menor, melhor</li> </ul> <p>#Tempo médio para falhar</p> <ul style="list-style-type: none"> <li>- Quanto maior, melhor</li> </ul> <p>#Número de falhas</p> <ul style="list-style-type: none"> <li>- Quanto menor, melhor</li> </ul>		<p>#Tratamento de exceções e reconhecimento de padrões de falhas podem resultar em</p> <ul style="list-style-type: none"> <li>- menor número de falhas</li> </ul>	

## APÊNDICE B – PLANILHAS GQM DO PARTICIPANTE 2

<b>Objeto</b>	<b>Propósito</b>	<b>Foco de qualidade</b>	<b>Ponto de vista</b>
<i>Sistemas autoadaptativos</i>	<i>Avaliar</i>	<i>Disponibilidade</i>	<i>Usuário final e desenvolvedor</i>
<b>Foco de qualidade</b>		<b>Fatores de variação</b>	
#Tempo - entre o recebimento de estímulos (dados de contexto) #Estímulos - número de estímulos recebidos/monitorados		# Desempenho dos sensores - quanto menor o tempo de repasse de estímulos maior deve ser o desempenho	
<b>Hipótese de base (estimativas)</b>		<b>Impacto dos fatores de variação na hipótese de base</b>	
#Tempo entre o recebimento de estímulos - média do tempo entre o recimento, quanto menor, melhor #Estímulos - taxa de estímulos recebidos/monitorados em relação ao total de estímulos enviados para sensores. Quanto mais próximo de 100%, melhor		#Críticidade do sistema - quanto maior a criticidade do sistema, menor deve ser o tempo de repasse de estímulos	

### APÊNDICE C – PLANILHAS GQM DO PARTICIPANTE 3

<b>Objeto</b>	<b>Propósito</b>	<b>Foco de qualidade</b>	<b>Ponto de vista</b>
<i>Sistemas autoadaptativos</i>	<i>Avaliar</i>	<i>Disponibilidade</i>	<i>Usuário final e desenvolvedor</i>
<b>Foco de qualidade</b>		<b>Fatores de variação</b>	
#Interrupções - causadas por adaptação #Tempo - para responder uma requisição dada a ocorrência de uma adaptação em paralelo		#Qualidade da adaptação - quanto menos interrupções por adaptação maior a qualidade da adaptação e do sistema	
<b>Hipótese de base (estimativas)</b>		<b>Impacto dos fatores de variação na hipótese de base</b>	
#Interrupções causadas por adaptação - máximo 5 interrupções por utilização - quanto menos, melhor #Tempo para responder uma requisição dada a ocorrência de uma adaptação em paralelo - máximo 5s, pois já é perceptível ao usuários - quanto menor, melhor		#Otimização do processo de adaptação - quanto mais otimizado o processo de adaptação, menos tempo ela demorará pra responder a requisições mesmo com adaptações ocorrendo em paralelo	

<b>Objeto</b>	<b>Propósito</b>	<b>Foco de qualidade</b>	<b>Ponto de vista</b>
<i>Sistemas autoadaptativos</i>	<i>Avaliar</i>	<i>Tolerância a falhas</i>	<i>Usuário final e desenvolvedor</i>
<b>Foco de qualidade</b>		<b>Fatores de variação</b>	
#Falhas - por erro de adaptação toleradas			
<b>Hipótese de base (estimativas)</b>		<b>Impacto dos fatores de variação na hipótese de base</b>	
#Taxa de falhas por erro de adaptação toleradas - taxa de falhas toleradas em relação ao número total de falhas			

## APÊNDICE D – PLANILHAS GQM DO PARTICIPANTE 4

<b>Objeto</b>	<b>Propósito</b>	<b>Foco de qualidade</b>	<b>Ponto de vista</b>
<i>Sistemas autoadaptativos</i>	<i>Avaliar</i>	<i>Disponibilidade</i>	<i>Usuário final e desenvolvedor</i>
<b>Foco de qualidade</b>		<b>Fatores de variação</b>	
#Tempo - disponível - de adaptação #Requisições - requisições respondidas		#Tempo de adaptação	
<b>Hipótese de base (estimativas)</b>		<b>Impacto dos fatores de variação na hipótese de base</b>	
#Tempo disponível - tempo total de operação/tempo disponível; - Alto: maior ou igual a 95% - Médio: entre 70% e 95% - Baixo: menor que 70% #Tempo de adaptação - instante final da adaptação - instante inicial; quanto menor, melhor #Requisições respondidas - requisições respondidas/requisições recebidas; pelo menos 90% das requisições respondidas		#Tempo de adaptação - quanto maior o tempo de adaptação menor o tempo que o sistema permanece ativo	

<b>Objeto</b>	<b>Propósito</b>	<b>Foco de qualidade</b>	<b>Ponto de vista</b>
<i>Sistemas autoadaptativos</i>	<i>Avaliar</i>	<i>Tolerância a falhas</i>	<i>Usuário final e desenvolvedor</i>
<b>Foco de qualidade</b>		<b>Fatores de variação</b>	
#Requisições - atendidas na presença de uma falha #Falhas - ocorrendo sem "crashar"o sistema - ocorrendo nas regras de adaptação sem "crashar"o sistema - previstas		#Planos de adaptação - quantidade de execuções até a adaptação correta	
<b>Hipótese de base (estimativas)</b>		<b>Impacto dos fatores de variação na hipótese de base</b>	
#Requisições atendidas na presença de uma falha - requisições recebidas/requisições respondidas (considerando que existe uma falha "ativa") - quanto mais, melhor #Taxa de falhas toleradas - falhas toleradas/ falhas totais ; quanto maior, melhor #Taxa de falhas previstas - falhas previstas/falhas totais		#Disponibilidade - quanto mais falhas toleradas, maior a disponibilidade	

<b>Objeto</b>	<b>Propósito</b>	<b>Foco de qualidade</b>	<b>Ponto de vista</b>
<i>Sistemas autoadaptativos</i>	<i>Avaliar</i>	<i>Recuperabilidade</i>	<i>Usuário final e desenvolvedor</i>
<b>Foco de qualidade</b>		<b>Fatores de variação</b>	
<p>#Tempo - média do tempo de recuperação após falha</p> <p>#Funcionalidades - disponíveis após falha</p> <p>#Modelos internos (modelos de contexto) - resincronização de modelos internos após falha</p>		<p>#Tempo de recuperação - quanto maior , menor a disponibilidade do sistema</p> <p>#Funcionalidades disponíveis - quanto menor, menor a disponibilidade do sistema</p> <p>#Ressincronização de modelos internos - quanto mais atualizados, mais confiável é o mecanismo de adaptação em tempo real</p>	
<b>Hipótese de base (estimativas)</b>		<b>Impacto dos fatores de variação na hipótese de base</b>	
<p>#Tempo médio de recuperação após falha - quanto menor, melhor</p> <p>#Número de funcionalidades disponíveis após falha - quanto maior , melhor</p> <p>#Modelos internos - porcentagem que um modelo representa um sistema</p>		<p>#Mecanismos de tratamento de falhas - diminuem o tempo de recuperação de falhas</p>	

<b>Objeto</b>	<b>Propósito</b>	<b>Foco de qualidade</b>	<b>Ponto de vista</b>
<i>Sistemas autoadaptativos</i>	<i>Avaliar</i>	<i>Robustez</i>	<i>Usuário final e desenvolvedor</i>
<b>Foco de qualidade</b>		<b>Fatores de variação</b>	
<p>#Adaptação - corretas dado uma inconsistência de contexto</p> <p>#Funcionalidades - ativas quando há restrição de recursos</p>		<p>#Mecanismo de adaptação - tempo de adaptação</p> <p>#Lógica de negócio - requisições atendidas</p>	
<b>Hipótese de base (estimativas)</b>		<b>Impacto dos fatores de variação na hipótese de base</b>	
<p>#Taxa de adaptações corretas dada uma inconsistência de contexto - número de adaptações corretas/número total de adaptações - quanto mais perto de 100%, melhor</p> <p>#Funcionalidades ativas dada restrição de recursos - número de funcionalidades ativadas - quanto maior, melhor</p>		<p>#Quanto maior o número de inconsistências, maior o tempo de adaptação</p>	

<b>Objeto</b>	<b>Propósito</b>	<b>Foco de qualidade</b>	<b>Ponto de vista</b>
<i>Sistemas autoadaptativos</i>	<i>Avaliar</i>	<i>Maturidade</i>	<i>Usuário final e desenvolvedor</i>
<b>Foco de qualidade</b>		<b>Fatores de variação</b>	
#Tempo - sem falhar - com desempenho esperado		#Disponibilidade #Qualidade do serviço	
<b>Hipótese de base (estimativas)</b>		<b>Impacto dos fatores de variação na hipótese de base</b>	
#Tempo sem falhar - média do tempo sem falhar; quanto maior, melhor #Tempo com desempenho esperado - quanto maior, melhor		#Quanto maior o tempo sem falhar, maior a disponibilidade e a qualidade do serviço	

## APÊNDICE E – PLANILHAS GQM DO PARTICIPANTE 5

<b>Objeto</b>	<b>Propósito</b>	<b>Foco de qualidade</b>	<b>Ponto de vista</b>
<i>Sistemas autoadaptativos</i>	<i>Avaliar</i>	<i>Disponibilidade</i>	<i>Usuário final e desenvolvedor</i>
<b>Foco de qualidade</b>		<b>Fatores de variação</b>	
<p>#Tempo</p> <ul style="list-style-type: none"> <li>- médio para falhar</li> <li>- médio entre falhas</li> <li>- médio para se recuperar de uma falha</li> </ul>		<p>#Fatores Ligados ao produto</p> <ul style="list-style-type: none"> <li>- Decisões arquiteturais (arquitetura, tecnologias, padrões, estratégias)</li> <li>Hardware</li> <li>- Sistema Operacional, Midleware, etc.</li> <li>- Complexidade do design</li> </ul>	
<b>Hipótese de base (estimativas)</b>		<b>Impacto dos fatores de variação na hipótese de base</b>	
<p>#Tempo médio para falhar</p> <ul style="list-style-type: none"> <li>- média do tempo que o sistema costuma funcionar até que uma falha ocorra</li> </ul> <p>#Tempo médio entre falhas</p> <ul style="list-style-type: none"> <li>- média do tempo de funcionamento do sistema entre falhas (contar após a primeira falha)</li> </ul> <p>#Tempo médio de recuperação</p> <ul style="list-style-type: none"> <li>- média do tempo que o sistema demora para voltar ao normal após a ocorrência de uma falha</li> </ul> <p><i>As forma de interpretação deve ser definida de acordo com o contexto do sistema</i></p>		<p>Fatores ligados ao time</p> <p>Experiência do time -&gt; Diminui a probabilidade de falhas</p> <p>Processo de desenvolvimento -&gt; Diminui a probabilidade de falhas</p> <p>Conhecimento do negócio -&gt; Diminui a probabilidade de falhas e de necessidades de mudanças</p> <p>Fatores Ligados ao produto</p> <p>Decisões arquiteturais (arquitetura, tecnologias, padrões, estratégias) -&gt; pode melhorar a robustez do sistema e conseqüentemente diminuir a probabilidade de falhas</p> <p>Hardware -&gt; a escolha adequada pode minimizar a necessidade de implementar mecanismos de tolerância a falhas no sistema</p>	