



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

**MAIKE BEZERRA DA SILVA**

**UM CATÁLOGO DE MEDIDAS DE DESEMPENHO PARA AVALIAÇÃO DE  
SISTEMAS AUTOADAPTATIVOS**

**QUIXADÁ**

**2020**

MAIKE BEZERRA DA SILVA

UM CATÁLOGO DE MEDIDAS DE DESEMPENHO PARA AVALIAÇÃO DE SISTEMAS  
AUTOADAPTATIVOS

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

Orientadora: Prof<sup>ª</sup>. Dra. Carla Ilane Moreira Bezerra

QUIXADÁ

2020

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

---

S581c Silva, Maíke Bezerra da.  
Um catálogo de medidas de desempenho para avaliação de sistemas autoadaptativos / Maíke Bezerra da Silva. – 2020.  
53 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Engenharia de Software, Quixadá, 2020.  
Orientação: Profa. Dra. Carla Ilane Moreira Bezerra.

1. Sistema autoadaptativo. 2. Medidas. 3. Avaliação de desempenho. I. Título.

CDD 005.1

---

MAIKE BEZERRA DA SILVA

UM CATÁLOGO DE MEDIDAS DE DESEMPENHO PARA AVALIAÇÃO DE SISTEMAS  
AUTOADAPTATIVOS

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

Aprovado em: \_\_\_\_/\_\_\_\_/\_\_\_\_.

BANCA EXAMINADORA

---

Prof<sup>a</sup>. Dra. Carla Ilane Moreira  
Bezerra (Orientadora)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Paulo Henrique Mendes Maia  
Universidade do Estadual do Ceará (UECE)

---

Prof. Dr. Emanuel Ferreira Coutinho  
Universidade do Federal do Ceará (UFC)

## AGRADECIMENTOS

Ao Prof.<sup>a</sup>. Dra. Carla Ilane Moreira Bezerra por me orientar em meu trabalho de conclusão de curso.

Aos Prof.s. Drs. Paulo Henrique Mendes Maia e Emanuel Ferreira Coutinho, por fazerem parte da banca examinadora.

Ao Doutorando em Engenharia Elétrica, Ednardo Moreira Rodrigues, e seu assistente, Alan Batista de Oliveira, aluno de graduação em Engenharia Elétrica, pela adequação do *template* utilizado neste trabalho para que o mesmo ficasse de acordo com as normas da biblioteca da Universidade Federal do Ceará (UFC).

Aos bibliotecários da Universidade Federal do Ceará: Francisco Edvander Pires Santos, Juliana Soares Lima, Izabel Lima dos Santos, Kalline Yasmin Soares Feitosa e Eliene Maria Vieira de Moura, pela revisão e discussão da formatação utilizada neste *template*. Aos meus pais, irmãos e tios, que me apoiaram na decisão de estudar um curso superior, sempre fizeram entender que temos que correr atrás dos nossos objetivos.

A minha esposa e filha, pelo apoio que me deram para a conclusão deste curso.

Agradeço a todos os professores por me proporcionar 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.

Agradeço a turma de alunos do curso por me proporcionarem momentos únicos e me fazer acreditar que tudo é possível quando se acredita.

## RESUMO

Sistemas autoadaptativos (SAS) são sistemas capazes de avaliarem seu próprio comportamento em tempo de execução, e alterá-lo quando julgar necessário para evitar a presença de falhas. Muitos desses sistemas requerem um alto poder de processamento para efetuarem com sucesso suas operações dinâmicas e complexas, mantendo o nível de resposta aos seus usuários. Como forma de ajudar o SAS a garantir sua qualidade, podem ser realizadas avaliações de qualidade, podendo se tratar tanto de uma avaliação qualitativa quanto quantitativa baseadas em medidas de software, que buscam apresentar as perspectivas dos atributos do sistema. Por ser uma característica essencial e fazer parte da maioria dos *trade-offs* presentes em SAS, o desempenho se torna importante na avaliação geral dos SAS em níveis de adaptações, avaliando de maneira geral seus impactos. Dessa forma, o objetivo desse trabalho é construir um catálogo de medidas de desempenho para avaliar sistemas autoadaptativos. Para isso, foi realizado nesse trabalho uma revisão da literatura, identificando as subcaracterísticas e medidas de desempenho em 32 estudos primários selecionados nessa revisão. A maior parte das medidas identificadas nos estudos primários, são relacionadas as subcaracterísticas de comportamento em relação ao tempo, e algumas medidas relacionadas a utilização de recursos. No entanto, não foram identificadas medidas relacionadas a subcaracterística capacidade, que é fundamental em um SAS, uma vez que impacta diretamente o usuário final e deve ser preservada em suas adaptações. Para validar o catálogo construído, foram implementadas algumas medidas em um sistema autoadaptativo de *healthcare* onde foram avaliadas por dados extraídos de seus relatórios.

**Palavras-chave:** Sistemas autoadaptativos. Medidas. Desempenho.

## ABSTRACT

Self-adaptive systems (SAS) are systems capable of evaluating their behavior at runtime and changing it when deemed necessary to avoid the presence of failures. Many of these systems require high processing power to successfully carry out their dynamic and complex operations, maintaining the level of response to their users. As a way of helping SAS to guarantee its quality, quality assessments can be carried out, which can be both a qualitative and quantitative assessment based on measures of software, which seeks to present the perspectives of the system's attributes. Because it is an essential feature and is part of the majority of trade-offs present in SAS, performance becomes important in the general assessment of SAS in terms of adaptations and assessing their impacts. Thus, the goal of this work is to build a catalog of performance measures to evaluate self-adaptive systems. A literature review was carried out in this work, identifying the sub-characteristics and performance measures in 32 primary studies selected in this review. Most of the measures identified in the primary studies are related to the sub-characteristics of behavior over time, and some measures related to the use of resources. However, measures related to sub-characteristic capacity have not been identified, which is fundamental in a SAS. It directly impacts the end-user and must be preserved in its adaptations. To validate the built catalog, some measures were implemented in a self-adaptive healthcare system where they were evaluated by data extracted from their reports.

**Keywords:** Self-adaptive Systems. Measures. Performance.

## LISTA DE FIGURAS

Figura 1 – Características e atributos de qualidade da ISO/IEC 25010 (ISO/IEC, 2011)	17
Figura 2 – Etapas de execução dos procedimentos metodológicos . . . . .	23
Figura 3 – Processo de revisão da literatura estruturada . . . . .	25
Figura 4 – Tela principal de análise do sistema . . . . .	36
Figura 5 – Monitor do sistema para verificação de recursos . . . . .	42



## LISTA DE TABELAS

Tabela 1 – Comparação entre os trabalhos relacionados e o proposto . . . . .	22
Tabela 2 – Estudos por base de dados . . . . .	28
Tabela 3 – Conjunto final de artigos selecionados. . . . .	31
Tabela 4 – Classificação das medidas extraídas na revisão da literatura . . . . .	32
Tabela 5 – Distribuição da quantidade de medidas definidas por subcaracterística . . .	33
Tabela 6 – Catálogo de medidas de desempenho para avaliar sistemas autoadaptativos. .	34
Tabela 7 – Resultados da coleta e cálculo das medidas de comportamento em relação ao tempo. . . . .	41
Tabela 8 – Resultados da coleta e cálculo das medidas de utilização de recursos. . . . .	42

## LISTA DE ABREVIATURAS E SIGLAS

SAS	Sistemas Autoadaptativos
TAS	<i>Tele Assistance System</i>
CSV	<i>comma-separated values</i>
IDE	<i>Integrated Development Environment</i>
RAM	<i>Random Access Memory</i>

## SUMÁRIO

1	<b>INTRODUÇÃO</b>	11
2	<b>FUNDAMENTAÇÃO TEÓRICA</b>	14
2.1	<b>Sistemas autoadaptativos</b>	14
2.2	<b>Modelo de qualidade</b>	16
2.3	<b>Desempenho em sistemas autoadaptativos</b>	17
2.3.1	<i>Medidas de desempenho</i>	18
3	<b>TRABALHOS RELACIONADOS</b>	20
4	<b>PROCEDIMENTOS METODOLÓGICOS</b>	23
4.1	<b>Identificar medidas de desempenho para sistemas autoadaptativos</b>	23
4.2	<b>Identificar ferramentas que medem o desempenho de sistemas autoadaptativos</b>	23
4.3	<b>Selecionar sistema autoadaptativo</b>	24
4.4	<b>Implementar medidas para automatizar a avaliação do sistemas autoadaptativo</b>	24
4.5	<b>Avaliar o sistema autoadaptativo</b>	24
5	<b>UM CATÁLOGO DE MEDIDAS DE DESEMPENHO PARA SISTEMAS AUTOADAPTATIVOS</b>	25
5.1	<b>Metodologia de construção do catálogo</b>	25
5.2	<b>Catálogo de medidas de desempenho</b>	28
6	<b>AVALIAÇÃO DAS MEDIDAS DE DESEMPENHO</b>	35
6.1	<b>Sistema autoadaptativo</b>	35
6.2	<b>Ferramentas</b>	36
6.3	<b>Implementação das medidas de desempenho</b>	37
6.4	<b>Validação</b>	39
6.5	<b>Resultados e análise dos dados</b>	40
7	<b>CONCLUSÕES E TRABALHOS FUTUROS</b>	44
	<b>REFERÊNCIAS</b>	46

## 1 INTRODUÇÃO

Sistemas Autoadaptativos (SAS) estão ganhando espaço em nossos modelos de sistemas atuais. A busca pela automatização de tarefas vem diminuindo cada vez mais o contato humano na gestão do software, com o objetivo de acabar a espera do usuário final acarretada por manutenções nesses sistemas (WEYNS, 2019). Pode-se afirmar que ganhos são gerados tanto para equipes que trabalham na manutenção desses sistemas quanto aos usuários que dependem desse serviço.

De maneira geral, os SAS estão presentes em grupos de sistemas que precisam garantir vários aspectos de qualidade (e.g., disponibilidade, segurança, eficiência, desempenho), que são impostos sobre a alta demanda diária dos fluxos de informações trafegados por eles (e.g., sistemas críticos, cidades digitais). Muitos deles oferecem serviços 24 horas por dia 7 dias na semana ou 24/7 (WEYNS, 2019).

Por se adaptarem em tempo de execução, há necessidade dos SAS garantirem a qualidade na entrega de seus serviços comprometem o desempenho do sistema, seja ele causado, por falhas inseridas ao tentar se adaptarem enquanto atendem a quantidade massiva de usuários que estão utilizando simultaneamente o sistema, ou pela quantidade de dados trafegados nesse período, ou ainda pelas falhas mantidas no código (BOWERS *et al.*, 2018). Essas falhas podem gerar erros afetando diretamente o desempenho do sistema.

Nesse caso, é importante a priorização de uma avaliação de qualidade para garantir as necessidades de seus cenários atuais. Além disso, as propriedades *self*-\* (e.g., *self-adaptation*, *self-protection*, *self-management*) presentes nos SAS impõem desafios em sua avaliação de qualidade (RAIBULET, 2014). Para contornar essas dificuldades muitos métodos identificam abordagens induzidas por atributos de qualidade específicos (POTENA, 2013; SLIEM *et al.*, 2015; PERUMA; KRUTZ, 2018), outros ainda lidam com métodos que avaliam outra gama de sistemas buscando obter bons resultados em suas avaliações (NOBRE *et al.*, 2013; SALAÜN *et al.*, 2013; MCBRIDE; LEPMETS, 2016).

Segundo a norma ISO 25010 ISO/IEC (2011), o modelo de qualidade determina quais características de qualidade serão levadas em consideração ao avaliar as propriedades de um sistema de software. Quando essa característica é muito subjetiva ou considerada um atributo de qualidade não funcional, é dividida em subcaracterísticas para que possa ser medida. A norma apresentada define oito características de qualidade, e uma dessas características é a eficiência de desempenho.

As medidas de avaliação de um sistema atuam como medidores do seu grau de qualidade oferecido, cada medida pode compor determinadas características de qualidade. Quando não se pode garantir um grau satisfatório de qualidade a uma propriedade com apenas uma medida, um conjunto de medidas são propostas para sua avaliação. Não há como garantir todas as características de qualidade em um único sistema. Desse modo, uma avaliação pode ser guiada pelos atributos de qualidade que o sistema avaliado garante cumprir ISO/IEC (2011).

Para melhorar o desempenho dos SAS, as propriedades *self-\** evoluem em paralelo com as adaptações realizadas pelo sistema. Algumas dessas propriedades podem ter impacto direto no desempenho do sistema. Outro conjunto de propriedades são as que compartilham de *trade-offs* em relação ao desempenho, quando alguma dessas propriedades é melhorada, o desempenho do sistema diminui. Assim, se essa propriedade sofre alteração negativa a melhoria no desempenho do sistema pode ser identificada. Dessa forma, é importante minimizar o impacto negativo sobre as propriedades que representam *trade-offs* relacionado ao desempenho (FARAHANI *et al.*, 2016).

É importante ressaltar que na literatura pode-se identificar vários trabalhos que citam o desempenho em sistemas autoadaptativos como característica de qualidade importante (CORDY *et al.*, 2013; FARAHANI *et al.*, 2016; LEMOS *et al.*, 2017). No entanto, existem ainda poucos trabalhos que avaliam o desempenho dos SASs de forma eficiente, pelo uso reduzido de medidas de desempenho disponíveis para melhor avaliação desses sistemas.

Neste contexto, este trabalho tem como objetivo geral desenvolver um catálogo de medidas para avaliação do desempenho de sistemas autoadaptativos. Para isso, será realizada a extração de medidas de desempenho da literatura para criação de um *dataset*, contendo as especificações e suas medidas de desempenho avaliadas por cada estudo. Espera-se, dessa forma, encontrar um grande número de medidas que possam ser avaliadas em ambientes autoadaptativos, os quais são relevantes para essa pesquisa. É importante lembrar que algumas das medidas presentes no *dataset* podem sofrer mudanças de implementação devido ao ambiente de implantação do sistema. Por fim, serão realizadas avaliações dessas medidas de desempenho em conjunto com as subcaracterísticas de desempenho correspondentes. Com isso, podem se beneficiar deste trabalho pesquisadores da área de sistemas autoadaptativos e desenvolvedores que têm por finalidade explorar a característica de desempenho e suas subcaracterísticas.

O trabalho estrutura-se da seguinte forma: no Capítulo 2 são apresentados os conceitos principais sobre o tema deste trabalho; o Capítulo 3 apresenta os trabalhos relacionados se

que utilizam de medidas para avaliação de desempenho em SAS; no Capítulo 4 são descritos as etapas da metodologia do estudo; no Capítulo 5 é detalhado o processo de construção do catálogo de medidas; no Capítulo 6 detalha-se o processo de implementação e avaliação das medidas obtidas pela pesquisa; finalmente no Capítulo 7 são apresentadas as conclusões do trabalho e trabalhos futuros que podem se beneficiar desta pesquisa.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste Capítulo serão apresentados os conceitos importantes presentes neste trabalho. Na Seção 2.1 são introduzidos os conceitos referentes aos sistemas autoadaptativos. Na Seção 2.2 será abordado o modelo ISO/IEC 25010 no que corresponde à sua definição e modelo de avaliação. A Seção 2.3 aborda como a qualidade no desempenho está presente no SAS, detalhado na Subseção 2.3.1 as medidas de desempenho usadas na avaliação de desempenho dos SAS.

### 2.1 Sistemas autoadaptativos

Para Oreizy *et al.* (1999) os SASs são sistemas capazes de modificar seu próprio comportamento para se adaptarem as mudanças e imprevistos que ocorrem em seu ambiente durante sua execução sem a necessidade de uma intervenção humana. Weyns (2019) afirma que a garantia de incertezas é uma das preocupações fundamentais para os SAS devido à grande quantidade de estresse absorvida por eles. Na sociedade atual, percebe-se que a tecnologia evolui rapidamente, proporcionando softwares mais robustos, gerando uma necessidade de manutenções contínuas e usuários mais exigentes. O desenvolvimento de sistemas com adaptações em tempo real garante o mínimo de impacto sobre manutenções futuras, causando uma melhoria significativa na qualidade dos serviços, e experiência dos usuários.

A qualidade dos serviços oferecidos pelos SASs está diretamente comprometida pelas mudanças que ocorrem a todo momento no ambiente interno e externo ao qual o sistema está vinculado. Neste caso, um *framework* que identifica uma boa representação sobre operações de mudanças complexas poderá ser usado (GRASSI *et al.*, 2009). Outro fator importante sobre mudanças em SASs é a adaptatividade realizada para correções do software que geram impacto direto na qualidade dos serviços do sistema. Existem vários tipos de adaptações em um serviço que podem ser identificados na literatura (ZHU *et al.*, 2004; RAIBULET; MASCIADRI, 2009b; POTENA, 2013; LOUKIL *et al.*, 2014; PEREZ-PALACIN *et al.*, 2014).

A autoadaptação é um desafio para os SASs, pois precisa-se garantir o comprometimento da qualidade sistema. Tais garantias são direcionadas para vários aspectos entre as dimensões do usuários e do sistema (LEMOS *et al.*, 2013). Quando uma adaptação é requisitada, torna-se necessária uma análise abrangente sobre o impacto da mudança na qualidade do serviço garantida por propriedades relacionadas ao tempo e que apresentam *trade-offs* (e.g., tempo de resposta, disponibilidade, precisão, segurança) (SANCHEZ *et al.*, 2015). As propriedades

qualitativas podem ser definidas em críticas e não-críticas. Esta classificação é importante para visualização do mapeamento de *trade-offs* entre eles (FARAHANI *et al.*, 2016). Este mapeamento é uma abordagem eficaz que identifica as dependências entre as propriedades do sistema tornando mais fácil seu gerenciamento.

Para IBM Computing *et al.* (2006) o contexto adaptativo pode ser alcançado pelo modelo MAPE-K representando as etapas monitorar, analisar, planejar, executar e conhecer. A fase monitorar é realizada pelo colhimento de informações geral do sistema ou sua parte crítica, a parte de análise processa as informações coletadas pelo monitor e possíveis dados históricos. Essas informações são repassadas para o plano, que reconhece o que precisa ser mudado para atingir os objetivos. O plano contém a decisão de como o sistema será adaptado utilizando diferentes regras e algoritmos. Por fim, a fase de execução tem como objetivo aplicar o plano de adaptação sobre o elemento gerenciado.

Os ambientes de um sistema são projetados com base na suas necessidades. Uma característica em comum é que são projetados para garantir o comportamento da autoadaptatividade dos sistemas. Portanto, identificar qual característica de qualidade é prioridade do sistema torna-se fundamental para o planejamento prévio do ambiente. Essa decisão garantirá ganhos em decisões como (e.g., linguagem de programação, arquitetura, uso de *frameworks*) (PERUMA; KRUTZ, 2018). Nos sistemas atuais é bastante comum nos depararmos com uma grande complexidade em seus códigos fontes. Para diminuir os riscos que comprometem um serviço de qualidade, o mais comum é realizar a migração do sistema ou parte dele para um local mais robusto, geralmente na nuvem (SALAÜN *et al.*, 2013). Desafios maiores para sistemas autoadaptativos podem ser perceptíveis quando trabalhamos com, por exemplo, inteligência artificial, robótica, sistemas críticos, onde a adaptatividade fica refém das decisões do projeto, e da troca de informações entre a parte lógica do sistema com seus componentes e *hardwares* (STEINBAUER; WOTAWA, 2013).

Em SAS, além da presença dos atributos de qualidade, são encontrados também propriedades *self*-. Essas propriedades representam os atributos adaptativos de um sistema, assim garantindo uma percepção maior da autoadaptação de um sistema quando observadas. Dentre essas propriedades, as que são gerenciadas em tempo de execução apresentam mais relação ao desempenho do sistema (MARCO *et al.*, 2013). Em geral, essas propriedades apresentam impactos positivos ou negativos diretamente sobre os atributos de qualidades do sistema quando alterados. Quando nos deparamos com esse tipo de dependência estamos tratando



de *trade-offs* (FARAHANI *et al.*, 2016). Essas propriedades são responsáveis pelas mudanças do próprio sistema (RAIBULET; MASCIADRI, 2009a).

As características de qualidade estão ligadas diretamente com processo de definição arquitetural, requisitos de negócios, e de aplicação. Diferente do processo de desenvolvimento em qualquer tipo de sistemas, os SASs apresentam uma agregação de valor a mais no processo de garantia da autoadaptação. Este baseia-se principalmente nas características que são garantidas pelo sistema (ZHU *et al.*, 2004). No entanto, muitas vezes não há como garantir todas as características exigidas. Nesse caso uma priorização ocorre para garantia da autoadaptação (YANG *et al.*, 2009).

## 2.2 Modelo de qualidade

A qualidade de um produto é definida pelo grau de satisfação de um usuário ao usar tal produto. Assim, a qualidade de um sistema pode ser classificada de diferentes maneiras por perspectivas de usuários distintos (HIRAMA, 2012). Em software, se garante qualidade quando as funcionalidades do sistema são satisfeitas sem erros e com menor tempo gasto possível (KOSCIANSKI; SOARES, 2007). Na prática a garantia de qualidade em qualquer sistema de software é assegurada pelos vários tipos de testes de sistemas gerenciados por todo processo de desenvolvimento do software (SANTOS; OLIVEIRA, 2017). No entanto, ainda é comum encontrar projetos não orientados a testes.

O modelo de avaliação de qualidade de um sistema é baseado em características que são garantidas por ele. Inicialmente avaliados pela qualidade externa e interna através da ISO/IEC 9126-1 (ISO/IEC, 2001). Com o tempo os softwares evoluíram tanto que o modelo não cobria todos os pontos de avaliações, uma reforma realizada no modelo, surgiu a ISO/IEC 25010 (ISO/IEC, 2011) resolvendo as novas necessidades de avaliação. Uma visualização detalhada do modelo é ilustrado na Figura 1, apresentando oito características de qualidade.

Como o foco deste trabalho é a criação de um catálogo de desempenho, será utilizada apenas a característica de qualidade eficiência de desempenho. Para a ISO/IEC (2011) a eficiência de um *software* é a sua capacidade de fornecer desempenho apropriado, no que se trata de uso de recursos e de tempo para realização de uma tarefa. A eficiência de desempenho é dividida em três subcaracterísticas, conforme detalhado a seguir:

- **Comportamento em relação ao tempo.** Refere-se, ao tempo de resposta, processamento e taxas de rendimento gastos por um sistema de software para atender determinado

- requisito;
- **Comportamento em relação aos recursos.** Refere-se, a quantidade e tipos de recursos utilizados por um sistema de software para realização de uma tarefa; e
  - **Capacidade.** Refere-se, aos limites máximos que um sistema de software pode gerenciar para atender a um requisito.

Figura 1 – Características e atributos de qualidade da ISO/IEC 25010 (ISO/IEC, 2011)

Qualidade de Produto de Software							
Adequação Funcional	Confiabilidade	Usabilidade	Eficiência de Desempenho	Manutenibilidade	Portabilidade	Segurança	Compatibilidade
Completude Funcional	Maturidade	Conhecimento Adequado	Comportamento em Relação ao Tempo	Analisabilidade	Adaptabilidade	Confidenciabilidade	Coexistência
Corretude Funcional	Tolerância a Falhas	Apreensibilidade	Comportamento em Relação aos Recursos	Modificabilidade	Instabilidade	Integridade	Interoperabilidade
Funcionalidade Apropriada	Recuperabilidade	Operabilidade	Capacidade	Modularidade	Substituibilidade	Não-Repúdio	
	Disponibilidade	Acessibilidade		Testabilidade		Responsabilização	
		Proteção de Erro de Usuário		Reusabilidade		Autenticidade	
		Estética de Interface com o Usuário					

Fonte: adaptado da ISO/IEC 25010 (2019).

A avaliação de desempenho em um SAS pode ter algumas limitações. O ambiente influencia diretamente nas medidas de desempenho. Assim, em ambientes com configurações mais robustas tendem a indicarem melhores resultados (SIRONI *et al.*, 2012). Portanto, ao se obter resultados insatisfatórios, pode indicar uma necessidade de troca de ambiente ou a realização de melhoria no mesmo.

### 2.3 Desempenho em sistemas autoadaptativos

A avaliação da qualidade nos SAS é desafiadora. Além da sua complexidade existem uma grande variedade de propriedades *self*-\* (e.g., autoadaptação, auto-organização, autoproteção, autocura) que apresentam relação entre as demais, uma vez que um grande número de possíveis soluções podem ser apresentadas para cada propriedade *self*-\* (RAIBULET, 2014). É necessário tornar a avaliação dos SAS mais simplificada e alcançar uma confiabilidade maior nos resultados (KADDOUM *et al.*, 2010).

Para ajudar nessa questão, uma classificação inicial do estudo pode ser arquitetada. O trabalho de Villegas *et al.* (2011) identifica oito dimensões em análises qualitativas em sistemas autoadaptativos. Distribuídos em objetivo de adaptação, referências de entradas, medidas de

saída, ações calculadas de controle, estrutura do sistema, propriedades de adaptação observadas, avaliação proposta e medidas identificadas. Para identificação de medidas a sugestão é de usar medidas que avaliem as variáveis de adaptação de interesse da pesquisa.

Uma avaliação de qualidade do ponto de vista autoadaptativo, proposta em (FARAHANI *et al.*, 2016), identificou fatores qualitativos, sendo avaliados desde a fase de requisitos, e estendendo-se por todo o processo de desenvolvimento do sistema. Suas medições foram realizadas por meio de critérios qualitativos (e.g., desempenho, segurança, modificabilidade), nos quais, após mapeados, é possível visualizar a presença de fatores críticos, e não-críticos de acordo com a relação em comum identificada pelos *trade-offs* entre eles.

Por estar fortemente vinculado à qualidade dos sistemas autoadaptativos, e comprometida em tempo de execução, o desempenho conta com algumas dificuldades de melhoria, pois seu processo de adaptação está sujeito a varias mudanças no sistema, adicionando e removendo novas funções (WEYNS, 2019). No entanto, podemos avaliar o impacto gerado por essas mudanças durante a adaptação analisando as características de qualidade do sistema, como proposto por (CALINESCU *et al.*, 2011).

Nos SAS, algumas de suas propriedades *self*-\* tem como objetivo tratar do desempenho do sistema. As referentes à automação e autoconfiguração garantem uma superação no atendimento do volume de requisições atendidas pelo sistema, e proporcionam uma melhoria no gerenciamento de recursos proporcionando ganhos no desempenho. Um gerenciador contínuo coleta e analisa informações para garantir um melhor controle das mudanças ocorrentes durante a execução do sistema. Com isso, novas configurações são implantadas garantindo a qualidade dos serviços oferecidos (SLIEM *et al.*, 2015).

### **2.3.1 Medidas de desempenho**

A norma ISO/IEC 25010 estabelece três subcaracterísticas para avaliar a eficiência no desempenho em um sistema de *software*, são elas: comportamento em relação ao tempo, utilização de recursos, e capacidade (ISO/IEC, 2011). Todas as medidas de eficiência de desempenho são identificadas em ambientes controlados. No entanto, algumas medidas de desempenho mostram mais claramente a eficácia desse controle (i.e, medidas de comportamento temporal) (KOSCIANSKI; SOARES, 2007).

O desempenho de um SAS estará sempre em evolução durante o período de suas execuções proporcionando uma instabilidade significativa, levando a ser um critério de qualidade

essencial para sua avaliação (FARAHANI *et al.*, 2016). Assim, a preocupação em sua garantia sobressai em função do tempo e capacidade de recursos necessários para conclusão de suas tarefas. Em relação à capacidade, supõe-se que, com a evolução contínua dos *hardwares* apresentando sempre melhorias em sua capacidade de processamento e armazenamento de dados, deixa-se de ser um desafio comum para SAS.

Na avaliação de sistemas autoadaptativos é bastante comum nos depararmos com a medida de desempenho (tempo de resposta) presente nos mais diversos tipos de sistemas e abordagens de avaliação (ORIOLE *et al.*, 2015; LUNG *et al.*, 2016; RAFIQUE *et al.*, 2019). Sua avaliação se torna eficiente pelo fato de mostrar em uma alta amplitude o desempenho de um sistema. No cenário adaptativo, pode ser mostrado dados interessantes a partir do tempo de resposta, como se o nível de adaptação está proporcionando um *overflow* nas respostas do sistema, indicando possíveis falhas em suas adaptações. Em continuidade o tempo de resposta, ainda pode ser usado como medida comparativa entre um sistema simples e suas adaptações (RAIBULET; MASCIADRI, 2009a).

Na literatura é comum identificar trabalhos que apresentam medidas de desempenho em relação ao tempo para operações em um SAS. Rattani *et al.* (2011) apresentam um experimento com processamento de imagens em um sistema biométrico realizado por múltiplos ciclos de *uploads*, onde, análises constantes do nível de desempenho eram realizadas durante sua execução. Os resultados mostram que as medidas relacionadas ao tempo de captura, e processamento das imagens não apresentavam grandes perdas no desempenho conseguindo atender o valor limite do tempo das tarefas mesmo em ciclos maiores. Outro trabalho similar pode ser observado em (ROLI *et al.*, 2008), tem como objetivo avaliar o autotreinamento de um algoritmo biométrico, avaliando seu comportamento através de do processamento de novas de imagens que serão submetidas, a fim de identificar as imagens já encontradas na base de dados e verificar seus níveis de acurácia.

Com relação a subcaracterística de utilização de recursos, é atribuído para todo recurso que estiver disponível para prover desempenho a um sistema de *software*, com exceção do tempo do CPU (KOSCIANSKI; SOARES, 2007). Em um SAS é comum encontrar a medida de taxa de transferência (*throughput*). Esse tipo de medida pode estar alinhado com recursos disponíveis do tipo rede, memória e processador (RAIBULET *et al.*, 2017).

### 3 TRABALHOS RELACIONADOS

Foram identificados estudos que auxiliam o desenvolvimento deste trabalho que possuem similaridades e diferenças com o trabalho proposto. Estes estudos são apresentados neste Capítulo, assim como suas principais contribuições e relações com o presente trabalho.

Em Chen *et al.* (2009), os autores propõem um método que auxilia na diminuição do tempo de espera para taxas de transferências de dados para uma aplicação *web*. A princípio, um equilíbrio entre as transferências solicitadas e o acesso simultâneo de usuários é mantido por controles do sistema. Com essa garantia, um aumento no desempenho é proporcionado positivamente, vale lembrar que essa evolução é mantida dinamicamente pelo sistema através dos *feedbacks* recebidos, sem a necessidade de adaptações realizadas por parte de seus mantenedores. As principais fontes de garantias trabalhadas para aumento do desempenho são tempo de resposta e *throughput*. Embora o trabalho apresente medidas de desempenho em SAS com abrangência em duas das três subcaracterísticas para medidas de desempenho proposta pela ISO 25010 (ISO/IEC, 2011), seu número medidas apresentadas está restrita a 2 medidas.

Em Marzolla e Mirandola (2010), os autores desenvolveram um *framework* que lida com a reconfiguração de sistemas proporcionando melhores desempenhos. O algoritmo opera as mudanças conforme a necessidade do sistema em dois momentos. O primeiro, se o tempo de resposta do sistema for menor que o tempo de resposta total aceitável, notificações são mandadas de um monitor para um controle, que por sua vez, dispara uma *trigger* realizando ou notificando um possível aumento em taxas de transferências de um ou mais componentes, já no segundo, se o tempo de resposta do sistema for maior que o tempo total, a *trigger* notifica que alguns componentes precisam diminuir as suas taxas de transferências. As medidas de desempenho usadas pelos autores para prover melhorias no sistemas foram, tempo de resposta, excesso de tempo de resposta, utilidade total. Mesmo com um número maior de medidas presente neste trabalho relacionado, não podemos expandir muito nossos resultados.

Em Lemos *et al.* (2017) realizaram um estudo sobre os desafios de pesquisa na provisão de garantias em SASs, com o objetivo de mostrar estes desafios direcionados para área de desenvolvimento, implantação, operação, evolução e descomissionamento de sistemas autoadaptativos. Foram utilizadas quatro aspectos para prover o percentual da provisão de garantias, sendo eles: garantia de capacidades de abordagens, garantia de base de aferição, garantia de rigor e garantia de desempenho. Cada um dos aspectos de garantias apresentaram um conjunto de medidas para mensurá-las, para garantia de desempenho usaram de três medidas

das quais são, *timeliness* como tempo de requisição requer para alcançar a resposta necessária, *overhead* computacional referente ao poder computacional requerido para realizar uma tarefa e complexidade sendo aplicabilidade de uma abordagem em diferentes tipos de problemas. As medidas de desempenho proposta neste são bem interessantes por não aparecerem nos outros trabalhos citados acima, porém o número total de medidas ainda se encontra reduzido.

Em Mousa *et al.* (2019) os autores propõem uma abordagem adaptativa em ambientes software como serviço (SaaS), utilizando o padrão *master/slave*. Tendo os *slaves* monitorando e adaptando os constituintes do SaaS distribuídos no ambiente de nuvem, e enviando as informações de desempenho para o mestre que por sua vez adapta ao serviço composto para fornecer a qualidade do serviço (QoS) global esperada pelos outros. Para poder realizar as adaptações são monitorados os tempos de respostas do serviço, que quando estão abaixo do esperado uma adaptação é realizada como resposta de melhoria do serviço. Embora este trabalho apresente um sistema adaptativo em um ambiente de nuvem, que é bem explorado nos últimos anos, existe a limitação de da avaliação do sistema apenas por uma medida de desempenho para realizar as adaptações.

O trabalho Mertz e Nunes (2019) é proposto um *framework* para monitoramento de software com o objetivo de realizar adaptações em sistemas com base nas medidas relevantes para o sistema monitorado. O trabalho é dividido em duas fases, na primeira fase o *framework* é induzido a obter apenas resultados das medidas simples que garantem a execução do sistema. Durante a segunda fase o *framework* avalia a coleção de medidas de acordo com critérios e filtros de execução que são relevantes ao monitoramento, tendo como objetivo de reduzir a sobrecarga de monitoramento. Apesar de ser uma abordagem importante, visando o mapeamento do fluxo de execução de um SAS, seus critérios de avaliação compõem uma quantidade mínima para atributos de desempenho considerando apenas a latência como medida principal.

Apesar do números de medidas presentes nos trabalhos apresentados serem reduzidas, cada trabalho está restringindo a um ambiente específico, onde suas medidas podem não ser identificadas em outros tipos de ambientes. O presente trabalho não se restringe a um ambiente específico e espera-se identificar um número maior de medidas de desempenho para SAS. Os trabalhos (CHEN *et al.*, 2009; MARZOLLA; MIRANDOLA, 2010), apresentam abordagens que facilitam a medição e a identificação das medidas, as quais se parecem com abordagem proposta no presente trabalho, já que foram implementadas as medidas de forma automatizada, ou ainda, pela propostas de construção de *scripts* que facilitaram a automação da coleta das

medidas.

Tabela 1 – Comparação entre os trabalhos relacionados e o proposto

<b>Critérios</b>	<b>(CHEN <i>et al.</i>, 2009)</b>	<b>(MARZOLLA; MIRANDOLA, 2010)</b>	<b>(LE MOS <i>et al.</i>, 2017)</b>	<b>(MOUSA <i>et al.</i>, 2019)</b>	<b>(MERTZ; NUNES, 2019)</b>	<b>Trabalho Proposto</b>
<b>Subcaracterísticas de Desempenho</b>	Comportamento em relação ao tempo, e recursos	Comportamento em relação ao tempo	Comportamento em relação ao tempo, e recursos	Comportamento em relação ao tempo	Comportamento em relação ao tempo	Comportamento em relação ao tempo, e recursos, Capacidade
<b>Fórmulas de Cálculo</b>	Sim	Sim	Não	Sim	Não	Sim
<b>Uso de Ferramentas para medição</b>	Não	Não	Não	Não	Sim	Sim

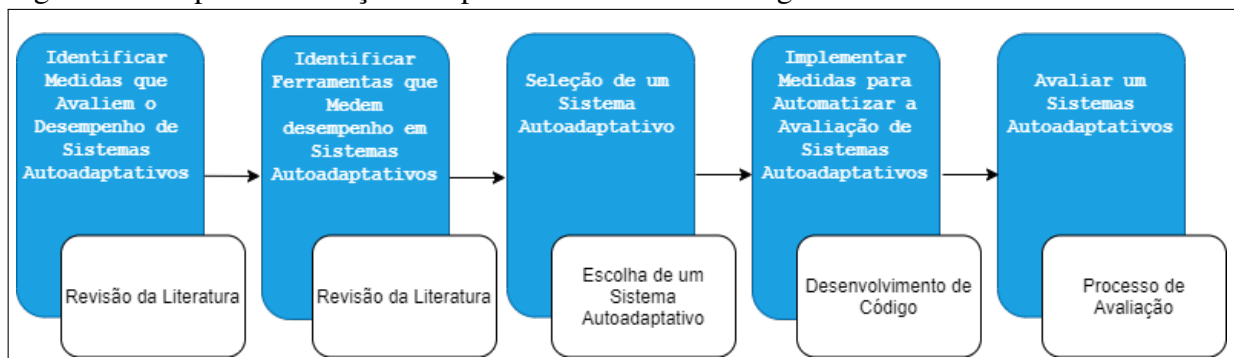
Fonte: elaborado pelo autor (2019).

A Tabela 1 resume as semelhanças e diferenças dos trabalhos citados neste Capítulo com o trabalho proposto, as mesmas são categorizadas em 3 níveis. O primeiro nível representa os tipos de subcaracterísticas de desempenho presente no trabalho, de acordo com a ISO 25010 (ISO/IEC, 2011). Pode-se observar que em todos trabalhos relacionados aparecem a subcaracterística de comportamento relacionado ao tempo, indicando grande contribuição nos trabalhos. A subcaracterística de utilização de recursos aparece em dois trabalho o que indica uma certa relevância para os estudos. Porém, a subcaracterística de capacidade não aparece nos estudos indicando uma deficiência em seu estudos e uma oportunidade de pesquisa. O segundo item indica se o trabalho proposto apresenta fórmulas de cálculo para as medidas identificadas, para maioria dos trabalhos os autores optaram por disponibilizar as fórmulas utilizadas em seus trabalhos como fonte de informação aos leitores e pesquisadores, outra parte prefere descrever como as medidas são utilizadas em seus estudos. Por fim, o terceiro item indica a utilização de algum tipo de ferramenta que auxilie na identificação de medidas ou cálculo automatizado, apresentando apenas um trabalho que dispõe do uso de ferramenta para auxiliar no cálculo ou identificação das medidas propostas, indicando uma possível lacuna a ser preenchida pelos trabalhos referentes a sistemas autoadaptativos.

## 4 PROCEDIMENTOS METODOLÓGICOS

Neste Capítulo serão apresentados os passos realizados para execução deste trabalho. Para melhor visualização pode-se observar a Figura 2, a qual reflete o caminho de execução deste trabalho, bem como a organização das seguintes seções. A Seção 4.1 aborda o processo de identificação das medidas de desempenho que serão utilizadas para avaliação deste trabalho. A Seção 4.2 explica o método de identificação de ferramentas que ajudaram no cálculo das medidas de desempenho identificadas. A Seção 4.3 mostra ações importantes na seleção do sistema autoadaptativo. A Seção 4.4 descreve a atividade de implementação das medidas em um sistema autoadaptativo. Por fim, a Seção 4.5 apresenta o planejamento da avaliação do sistema autoadaptativo para esta pesquisa, bem como as dificuldades enfrentadas durante esse processo.

Figura 2 – Etapas de execução dos procedimentos metodológicos



Fonte: elaborada pelo autor (2019).

### 4.1 Identificar medidas de desempenho para sistemas autoadaptativos

Neste passo será realizada uma busca na literatura por trabalhos que propõem uma avaliação sobre as subcaracterísticas e medidas de eficiência em desempenho para ambientes autoadaptativos. Espera-se, dessa forma, obter o maior número de medidas de desempenho possíveis para construção de um catálogo bem amplo e que possa ser utilizado em vários tipos de sistemas autoadaptativos.

### 4.2 Identificar ferramentas que medem o desempenho de sistemas autoadaptativos

Neste passo serão identificadas, na literatura e em outras fontes, ferramentas que se comprometem na avaliação de desempenho para SAS, as quais facilitem o processo de identificação de algumas medidas e automação de cálculos. Deve-se entender que, ao se tratar de



medidas de desempenho em SAS podem não haver ferramentas que realizem o cálculo de todas as medidas identificadas por este trabalho. Como forma de contornar isso, a Seção 4.4 propõe novas formas de mensurar medidas de desempenho.

### **4.3 Selecionar sistema autoadaptativo**

Será realizado a identificação de um sistema autoadaptativo que seja capaz de acolher modificações que possam surgir para realização do objetivo deste trabalho. Neste caso, torna-se indispensável a priorização por uma sistema de código aberto, uma vez que precisa-se de total acesso à documentação e codificação do sistema, necessários para o entendimento e execução do mesmo. Isso é praticamente indisponível por sistemas comerciais.

### **4.4 Implementar medidas para automatizar a avaliação do sistemas autoadaptativo**

Essa Seção tem como objetivo a criação de novas formas para atribuição e cálculo de medidas, (e.g., *scripts*, programas de rotineiros, implementação no código), ou utilizar de métodos presentes na literatura que podem mensurar medidas de desempenho não cobertas pelas ferramentas identificadas na Seção 4.2. Será analisada a viabilidade de implementação de cada medida do catálogo no sistema autoadaptativo.

### **4.5 Avaliar o sistema autoadaptativo**

O sistema autoadaptativo será avaliado com o objetivo de responder as questões de pesquisa deste trabalho. Importante lembrar que alguns desafios podem ser surgir durante essa etapa, podemos citar a adaptação do sistema para suportar as ferramentas descritas nas Seções 4.2 e os métodos descritos na Seção 4.4. A origem desse quadro torna-se negativa quando não atendida, gerando novas reavaliações na estratégia de análise das medidas. Fica evidente que uma padronização tem que ser aderida pelos ambientes de avaliação e validação.

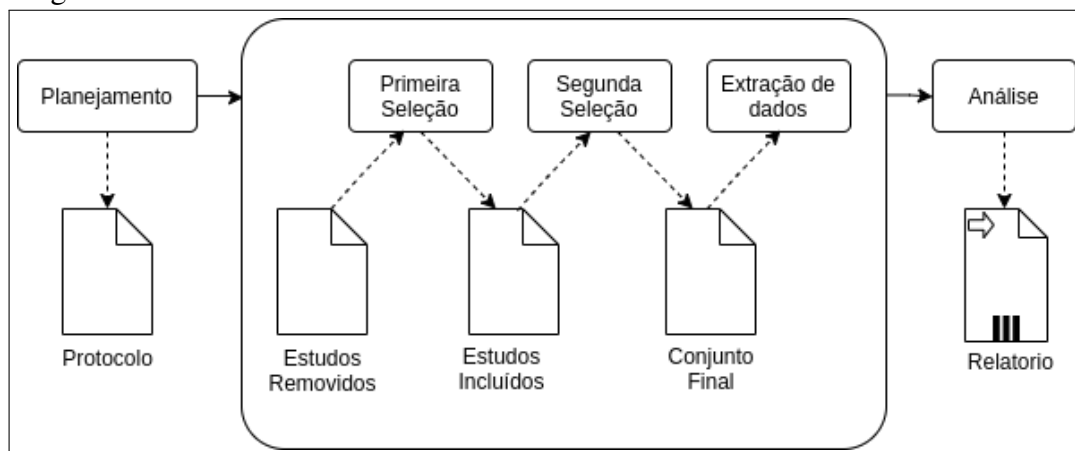
## 5 UM CATÁLOGO DE MEDIDAS DE DESEMPENHO PARA SISTEMAS AUTO-ADAPTATIVOS

Este Capítulo tem como objetivo explorar as fases que auxiliaram na construção de um catálogo para medidas de desempenho de sistemas autoadaptativos. A Seção 5.1 descreve a metodologia e etapas utilizadas na construção do catálogo realizada a partir de uma revisão da literatura; e a Seção 5.2 apresenta a organização das medidas de desempenho exploradas pelo estudo para construção do catálogo.

### 5.1 Metodologia de construção do catálogo

Para realização da construção do catálogo, foi adaptado um processo de revisão da literatura estruturada com objetivo de avaliar o estado da arte sobre subcaracterísticas e medidas relacionadas à eficiência em desempenho para avaliação de sistemas autoadaptativos. Para melhor visualização do planejamento da revisão a Figura 3 ilustra os passos da metodologia.

Figura 3 – Processo de revisão da literatura estruturada



Fonte: elaborada pelo autor (2020).

Na fase de planejamento, foi criado o plano de ações para a revisão da literatura. No protocolo produzido foram definidos os seguintes objetivos: identificar questões de pesquisa; filtro de palavras chaves para uma estratégia de busca; bibliotecas disponíveis para identificação dos estudos; critérios de inclusão e exclusão; e, extração e análise dos dados.

Dado o planejamento, foram estabelecidas as seguintes questões de pesquisa:

- $QP_1$  Quais as subcaracterísticas de eficiência de desempenho que são avaliadas em sistemas autoadaptativos?
- $QP_2$  Quais as medidas utilizadas para avaliar a eficiência de desempenho em sistemas

autoadaptativos?

Respondendo as questões de pesquisa, foram considerados os seguintes tipos de estudos:

- $QP_1$  Soluções que realizam avaliações em sistemas autoadaptativos usando subcaracterísticas, atributos de eficiência de desempenho.
- $QP_2$  Soluções que usam subcaracterísticas, atributos ou medidas de qualidade para avaliar a eficiência de desempenho em sistemas autoadaptativos.

Com isso, foram identificadas as palavras chaves “Sistemas autoadaptativos”, “Característica de desempenho”, “Atributos de desempenho” e “Medidas de desempenho”. Foram também destacados sinônimos e relações próximas com essas palavras, considerando seus plurais. Além dessas considerações foi utilizado dos operadores lógicos *OR* e *AND* para compor a *string* de busca. Para uma busca eficiente nos veículos de publicações as palavras foram implementadas apenas no idioma inglês. Assim a *string* construída é mostrada 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\*”) AND (“performance”)

Para atingir o maior número de trabalhos possíveis, foi considerado inicialmente quatro bases de dados: *ACM library*, *ScienceDirect*, *IEEE Xplore* e *Springer*. A confiabilidade dessas fontes para o tema desta pesquisa é confirmado em (DYBA *et al.*, 2007; KITCHENHAM; CHARTERS, 2007).

Por não ser compatível a todas bibliotecas, a *string* de busca sofreu uma adaptação para se adequar a base *ScienceDirect*, devido a sua restrição quanto ao uso do número de operadores lógicos permitindo no máximo oito dentro de uma mesma busca, e de não permitir o uso do caractere “\*”. Dada as limitações a *string* foi reformulada para a seguinte.

(“dynamic variability” OR “runtime variability” OR “reconfigura” OR “context-aware” OR “self-adaptive”) AND (“non-functional requirement”) AND (“metric” OR “measure”) AND (“performance”)

Ainda nesta fase foram determinados os critérios para seleção dos trabalhos utilizados na avaliação dos estudos retornados em cada base de dados. O uso dos critérios permitiu refinar a

identificação de estudos relevantes e que possam responder as questões de pesquisa, assim como realizar o descarte dos estudos irrelevantes. Dentre os critérios de inclusão (INC) e exclusão (EXC) estão:

- *INC*<sub>1</sub>: O resumo do trabalho menciona explicitamente a avaliação de desempenho no contexto de sistemas autoadaptativos.
- *INC*<sub>2</sub>: Estudos que descrevem abordagens que utilizam medidas de desempenho como forma de avaliarem resultados para pesquisa.
- *INC*<sub>3</sub>: Artigos publicados no período de 2000 a 2020.

Com o estabelecimento dos critérios de inclusão, o objetivo foi de identificar na literatura trabalhos que possuam estudos relacionados a avaliações de desempenho em sistemas autoadaptativos. O período de tempo teve por finalidade mostrar a evolução das soluções adaptativas ao longo dos anos, mostrando as tendências atuais e o modelo evolutivo na avaliação de desempenho.

- *EXC*<sub>1</sub>: O artigo está fora do escopo de avaliação de desempenho de sistemas autoadaptativos.
- *EXC*<sub>2</sub>: Avaliação de desempenho não é foco principal da pesquisa.
- *EXC*<sub>3</sub>: Artigos não escritos em inglês.
- *EXC*<sub>4</sub>: Artigos que possuem menos de 3 páginas.
- *EXC*<sub>5</sub>: Teses de doutorado ou dissertações de mestrado.
- *EXC*<sub>6</sub>: Estudos secundários.

Para os critérios de exclusão, o objetivo é realizar uma filtragem dentre os trabalhos selecionados, os que realizam uma avaliação fora do escopo do tema do presente trabalho e efetuar sua exclusão. Além disso, uma ideia da revisão literária foi identificar estudos com propostas de avaliação mais completas, o que levou a definição dos critérios de exclusão 4, 5 e 6.

A *string* de busca foi aplicada nas bibliotecas selecionadas com alguns filtros como adição. Foram selecionados trabalhos no período de 2000 e 2020, nas áreas de ciência da computação. Resultando em um conjunto de trabalhos que possivelmente seriam relacionados a este estudo. Neste momento, foi revisado primeiramente seus títulos, resumos e palavras-chaves recaindo sobre eles os critérios de inclusão e exclusão.

A revisão da literatura teve início em Agosto de 2020 e terminou em Setembro de 2020. Durante a fase de execução, foi utilizada a *string* de busca que foi adaptada para atender requisitos e restrições de algumas das bibliotecas. Na biblioteca *Springer* dois filtros foram

selecionados para garantir a assertividade da busca: ciência da computação e artigos científicos. A Tabela 2 ilustra o número de estudos retornados e incluídos por cada base de dados.

Tabela 2 – Estudos por base de dados

Base de Dados	Artigos Retornados	Artigos Incluídos
IEEE Xplorer	56	5
ScienceDirect	241	10
ACM Digital Library	84	6
Springer	223	11
<b>Total</b>	<b>604</b>	<b>32</b>

Fonte: bases de trabalhos científicos.

Do total de 604 artigos, foram subtraídos documentos que não eram artigos científicos (e.g., glossários, índices e capítulos de introdução de livros e outros temas). Resultando em 546 artigos para um conjunto primário. Durante a primeira seleção foi observado o resumo de cada estudo realizando a validação de que o trabalho se tratava de um estudo de com o objetivo verificar a eficiência de desempenho em um sistema autoadaptativo. Já na segunda seleção foi explorado no trabalho a identificação de subcaracterísticas e medidas de desempenho que avaliassem o estudo. Assim chegou-se ao conjunto final de 32 artigos que são apresentado na Tabela 3. Por limitação de espaço os nomes das bases foram abreviados: *ACM Digital Library* (ACM), *SpringerLink* (SL), *IEEE Xplorer* (IEEE) e *ScienceDirect* (SD).

## 5.2 Catálogo de medidas de desempenho

Independente da medida que se quer medir é necessário um planejamento prévio para realizar sua medição. Seja observar um sistema proposto, identificar como suas dependências estão distribuídas, monitorar as variações dos dados e como elas ocorrem. O processo de análise de uma medida pode ser categorizado nas seguintes etapas:

- **Desempenho:** característica do sistema que se deseja medir.
- **Medição:** objetivo principal.
- **Medidas:** representa as medidas de desempenho que serão expostas ao processo para que se tenha um resultado final.
- **Monitoramento:** processo que se observa as medidas identificando como estão distribuídas pelo sistema, e assim mostrar sua estratégia de medição.
- **Análise dos dados:** tem por objetivo mostrar os resultados obtidos podendo ser organizados para responder alguma questão de pesquisa.

Para definição do catálogo de medidas, primeiramente foi realizado um processo de identificação das medidas de desempenho representadas na literatura. Para encontrar tais medidas, foi realizada uma revisão da literatura sobre avaliação de desempenho em sistema autoadaptativos descrito na Seção 5.1. O processo de extração das subcaracterísticas e medidas de desempenho presentes no trabalho foi realizado a partir da sua identificação no trabalho e sua proposta de avaliação de desempenho para o sistema proposto pela pesquisa. Além disso, para uma avaliação mais qualificada das medidas extraídas, representando a resposta para  $QP_2$  indicando as medidas de desempenho utilizadas pelos estudos na área de SAS, foi proposto um modelo de classificação em três categorias mostradas a seguir conforme apresentado por Carvalho *et al.* (2018). Detalhes da extração na Tabela 4.

- **Bem definida:** a medida contém descrição e função de medição, assim um ou mais elementos de medição podem ser identificados.
- **Definida mas sem função de medição:** a medida é apresentada considerando apenas uma propriedade para quantificar
- **Não definida:** a medida não informa nenhuma informação sobre ela ou sua fórmula de cálculo, tornando difícil saber o que está sendo medido.

Ao final do processo de identificação das medidas, foi obtido um conjunto final de 24 medidas distribuídas entre as subcaracterísticas de desempenho que apresentam a resposta para  $QP_1$ , mostrando as subcaracterísticas de eficiência em desempenho utilizadas pelos estudos. A Tabela 5 apresenta a quantidade de medidas distribuídas pela subcaracterística equivalente.

Para realizar a documentação das medidas, foram listados atributos que podem descrever a medida com o máximo de detalhes possíveis, como propósito da medida, formula de cálculo e sua forma de interpretação. Com isso pode se cumprir o principal objetivo, permitindo que qualquer pessoa possa refletir a medida no sistema adaptativo que for abordado dado que cumpre as requisições. Para tanto as medidas que apresentaram uma classificação onde sua medição não é definida ou sua descrição é insuficiente, sera realizada sua exclusão da documentação uma vez que não podemos garantir sua assertividade para estudos que se basearem neste estudo. 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.

Para casos em que vários estudos utilizam da mesma medida, optou-se por usar no campo descrição e fórmula de medição a conversão mais aceita ou melhor interpretada. Já para os estudos que utilizam essa medida e não a descrevem em seu contexto, atribui-se também a descrição escolhida por este estudo. Para as medidas que foram categorizadas como não definida na Tabela 4, as mesmas foram removidas do catálogo por não apresentarem descrição ou fórmula de cálculo. Dessa forma, não há garantias de replicação dessas medidas para outros trabalhos.

Nas medidas apresentadas pelo conjunto final, pode-se observar que algumas medidas são novas. Isso pode indicar que para o domínio de sistemas autoadaptativos algumas medidas podem surgir para mesurar sua eficiência de desempenho. Outras medidas são comuns para outros tipos de sistemas, como tempo de resposta, *throughput* e latência. No entanto, há alguns desafios para suas medições em SAS podendo ocorrer adaptações no meio do processo, e assim provocando alterações nos resultados. O conjunto final de medidas pode ser visualizado na Tabela 6.

Tabela 3 – Conjunto final de artigos selecionados.

ID	Título	Referência	Base
E1	Evaluation of Self-Adaptive Systems: A Women Perspective	(RAIBULET; FONTANA, 2017)	ACM
E2	Synthesizing Self-Adaptive Connectors Meeting Functional and Performance Concerns	(MARCO <i>et al.</i> , 2013)	ACM
E3	Improving Context-Awareness in Self-Adaptation using the DYNAMICO Reference Model	(TAMURA <i>et al.</i> , 2013)	ACM
E4	On the Practical Feasibility of Software Monitoring: a Framework for Low-impact Execution Tracing	(MERTZ; NUNES, 2019)	ACM
E5	Optimal Planning for Architecture-Based Self-Adaptation Via Model Checking of Stochastic Games	(CÁMARA <i>et al.</i> , 2015)	ACM
E6	Combining Global Optimization with Local Selection for Efficient QoS-aware Service Composition	(ALRIFAI; RISSE, 2009)	ACM
E7	Dynamically Reconfigurable Slice Allocation and Admission Control within 5G Wireless Networks	(Perveen <i>et al.</i> , 2019)	IEEE
E8	FPGA-SDR Integration and Experimental Validation of a Joint DML SNR and Doppler Spread Estimator for 5G Cognitive Transceivers	(HAGGUI <i>et al.</i> , 2019)	IEEE
E9	Handover Decision in Heterogeneous Networks	(MAALOUL <i>et al.</i> , 2016)	IEEE
E10	Software, Performance and Resource Utilisation Metrics for Context-Aware Mobile Applications	(Ryan; Rossi, 2005)	IEEE
E11	Achieving functional and non functional interoperability through synthesized connectors	(NOSTRO <i>et al.</i> , 2016)	SD
E12	Container-based architecture for flexible industrial control applications	(GOLDSCHMIDT <i>et al.</i> , 2018)	SD
E13	Context-aware composite SaaS using feature model	(MOUSA <i>et al.</i> , 2019)	SD
E14	Context-aware RAON middleware for opportunistic network	(LAU <i>et al.</i> , 2017)	SD
E15	Dynamic adaptation of service compositions with variability models	(ALFÉREZ <i>et al.</i> , 2014)	SD
E16	Improving software performance and reliability in a distributed and concurrent environment with an architecture-based self-adaptive framework	(LUNG <i>et al.</i> , 2016)	SD
E17	Model-based system reconfiguration for dynamic performance management	(CAPORUSCIO <i>et al.</i> , 2007)	SD
E18	Monitoring the service-based system lifecycle with SALMon	(ORIOLO <i>et al.</i> , 2015)	SD
E19	SMADA-Fog: Semantic model driven approach to deployment and adaptivity in fog computing	(PETROVIC; TOSIC, 2020)	SD
E20	The ANTAREX domain specific language for high performance computing	(SILVANO <i>et al.</i> , 2019)	SD
E21	A dependability profile within MARTE	(BERNARDI <i>et al.</i> , 2011)	SL
E22	An approach based on feature models and quality criteria for adapting component-based systems	(SANCHEZ <i>et al.</i> , 2015)	SL
E23	Architecture-based resilience evaluation for self-adaptive systems	(CÁMARA <i>et al.</i> , 2013)	SL
E24	Autonomic fine-grained replication and migration at component level on multicloud	(PHAM; PHAM, 2017)	SL
E25	Experience with model-based performance, reliability, and adaptability assessment of a complex industrial architecture	(GOUVÊA <i>et al.</i> , 2013)	SL
E26	IPTV architecture for an IMS environment with dynamic QoS adaptation	(CRUZ <i>et al.</i> , 2011)	SL
E27	Plato: a genetic algorithm approach to run-time reconfiguration in autonomic computing systems	(RAMIREZ <i>et al.</i> , 2011)	SL
E28	Resource Provisioning Based Scheduling Framework for Execution of Heterogeneous and Clustered Workloads in Clouds: from Fundamental to Autonomic Offering	(GILL; BUYYA, 2019)	SL
E29	SCOPE: self-adaptive and policy-based data management middleware for federated clouds	(RAFIQUE <i>et al.</i> , 2019)	SL
E30	Self-adaptable media service architecture for guaranteeing reliable multimedia services	(KALAVATHY <i>et al.</i> , 2012)	SL
E31	Self-adaptation by coordination-targeted reconfigurations	(OLIVEIRA; BARBOSA, 2015)	SL
E32	Evaluation of dynamic adaptivity through metrics: an achievable target?	(RAIBULET; MASCIADRI, 2009a)	IEEE



Tabela 4 – Classificação das medidas extraídas na revisão da literatura

ID	Medida	Classificação	Tipos de SAS	Medição	Referência
M1	Tempo de resposta	Bem definida	Sistemas ( <i>Cloud</i> , WEB, IoT)	Modelo de <i>feature</i> , ferramentas (SAFCA, SCOPE), <i>framework</i> no contexto de elasticidade	(MOUSA <i>et al.</i> , 2019; LUNG <i>et al.</i> , 2016; PHAM; PHAM, 2017; RAFIQUE <i>et al.</i> , 2019)
M2	Throughput	Bem definida	Sistemas (rede, WEB, <i>cloud</i> ), <i>framework</i> autoadaptativo	Serviço de monitoramento, ferramentas (SAFCA, SALMon), implementação no código	(TAMURA <i>et al.</i> , 2013; MARCO <i>et al.</i> , 2013; LUNG <i>et al.</i> , 2016; ORIOL <i>et al.</i> , 2015; OLIVEIRA; BARBOSA, 2015)
M3	Latência	Bem definida	Sistemas (DSL, WEB, mobile, rede)	Sistema de monitoramento, serviço baseado em QoS, manual	(MERTZ; NUNES, 2019; ALRIFAI; RISSE, 2009; HAGGUI <i>et al.</i> , 2019; NOSTRO <i>et al.</i> , 2016)
M4	pLatencia	Bem definida	<i>Self-healing</i>	Realizada por especialistas	(RAIBULET; FONTANA, 2017; RAIBULET; MASCIADRI, 2009a)
M5	Desempenho da qualidade de resposta (pQoR)	Bem definida			
M6	Influência de desempenho na adaptatividade (pIA)	Bem definida			
M7	Tempo de Execução de Método (ET)	Bem definida	Aplicação mobile	Implementada no código fonte	(Ryan; Rossi, 2005)
M8	Tempo de Invocação de Método (IT)	Bem definida			
M9	Tempo de migração de uma instancia (MIT)	Bem definida			
M10	Tempo de migração de uma classe (MCT)	Bem definida			
M11	Taxas de sucesso de consultas	Bem definida	Aplicação mobile	Ferramenta: RAON	(LAU <i>et al.</i> , 2017)
M12	Atrasos de consulta	Bem definida			
M13	Taxa de requisição	Bem definida	<i>Framework</i> autoadaptativo	Ferramenta: SAFCA	(LUNG <i>et al.</i> , 2016)
M14	Queda da taxa de requisição	Bem definida			
M15	Utilização do <i>pool</i> de <i>thread</i>	Bem definida			
M16	Utilização de CPU	Bem definida			
M17	Leitura de <i>host</i>	Não definida	Sistema de <i>streaming</i>	Ferramenta: IPTV AS	(CRUZ <i>et al.</i> , 2011)
M18	Tempo de execução	Não definida			
M19	Tempo de conclusão de escrita (WCT)	Bem definida	SaaS e IoT	Ferramenta: SCOPE	(RAFIQUE <i>et al.</i> , 2019)
M20	Tempo de conclusão de leitura (RCT)	Bem definida			
M21	Tempo de inicialização	Bem definida	Sistema baseado em componentes	Gerenciador de componentes	(SANCHEZ <i>et al.</i> , 2015)
M22	Tempo de finalização	Bem definida			
M23	Tempo de processamento (execução sequencial)	Não definida			
M24	Tempo de processamento (execução paralela)	Não definida			

Fonte: dados da pesquisa.

Tabela 5 – Distribuição da quantidade de medidas definidas por subcaracterística

Subcaracterística	Medidas
Comportamento em relação ao tempo	M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M18, M19, M20, M21, M22, M23, M24
Utilização de recursos	M11, M12, M13, M14, M15, M16, M17
Capacidade	-

Fonte: dados da pesquisa.

Tabela 6 – Catálogo de medidas de desempenho para avaliar sistemas autoadaptativos.

ID	Nome	Descrição	Fórmula de Cálculo
M1	Tempo de resposta	Tempo médio para realizar uma operação e obter uma resposta	Tempo total gasto em milissegundos / número de operações
M2	<i>Throughput</i>	Número de eventos concluídos sem falhas em um determinado intervalo de tempo	Tempo total gasto em segundos / total de eventos concluídos sem falhas
M3	Latência	Custo de para uma requisição ser atendida	Tempo gasto por requisição
M4	pLatencia	Tempo de atraso das respostas do sistema na presença de adaptatividade em relação às respostas na ausência da adaptatividade	$pLatencia = (\text{tempo de resposta na presença da adaptatividade} / \text{tempo de resposta na ausência da adaptatividade}) * 100$
M5	Desempenho da qualidade de resposta (pQoR)	Representa o aumento da qualidade das respostas na presença da adaptatividade	$pQoR = (\text{qualidade de resposta na presença da adaptatividade} / \text{qualidade de resposta na ausência da adaptatividade}) * 100$
M6	Influência de desempenho na adaptatividade (pIA)	Indica se a adaptatividade é influenciada por outras medidas de desempenho	$pIA = 0$ se não influência e $1$ se influência
M7	Tempo de Execução de Método	Representa o custo para execução de um método do sistema	Tempo em milissegundos para execução de um método
M8	Tempo de Invocação de Método	Representa o custo para realizar uma ação, independente do processamento atual	Tempo em milissegundos gasto para realizar a invocação de um método
M9	Tempo de migração de uma instância	Custo para mover uma instância de objeto entre <i>hosts</i>	Tempo gasto em milissegundos para mover uma instância de objeto (e.g. <i>Java object</i> ) entre <i>hosts</i>
M10	Tempo de migração de uma classe	Custo para mover uma implementação de classe entre <i>hosts</i>	Tempo gasto em milissegundos para mover uma implementação de classe (e.g. <i>Java class</i> ) entre <i>hosts</i>
M11	Taxa de requisição	Solicitações recebidas por um servidor	Total de solicitações recebidas pelo servidor / tempo total gasto em segundos
M12	Queda da taxa de requisição	Solicitações recebidas, mas descartadas pelo servidor devido ao alto número de requisições	Total de solicitações descartadas / total de solicitações recebidas pelo servidor
M13	Utilização do <i>pool</i> de <i>thread</i>	Tarefas que são criadas pelas aplicações que são disparadas durante alguma condição específica	Total de <i>threads</i> em execução simultaneamente / total de <i>threads</i> criados no <i>pool</i> de <i>thread</i>
M14	Utilização de CPU	Uso agregado do processador de um <i>host</i>	Calculado pela ferramenta de alertas de Desempenho Microsoft Windows
M15	Tempo de conclusão de escrita (WCT)	Alinhado com operações de gravação em documentos	Tempo total para realizar uma operação de gravação
M16	Tempo de conclusão de leitura (RCT)	Alinhado com operações de leitura em documentos	Tempo total para realizar a operação de leitura
M17	Tempo de inicialização	Custo para iniciar um componente da aplicação	Tempo necessário para o gerenciador de componente leva para carregar um componente
M18	Tempo de desligamento	Custo para encerra um componente da aplicação	Tempo necessário para encerrar um componente

Fonte: dados da pesquisa.

## 6 AVALIAÇÃO DAS MEDIDAS DE DESEMPENHO

Neste Capítulo, é detalhada a aplicação das medidas de desempenho do catálogo em um sistema autoadaptativo. A Seção 6.1 apresenta o sistema autoadaptativo escolhido por este trabalho. A Seção 6.2 mostra o objetivo do uso de ferramentas para automatizar algum dos pontos da pesquisa, e quais se destacam na avaliação de sistemas autoadaptativos. A Seção 6.3 exhibe a abordagem adotada por este trabalho para a implementação das medidas de desempenho para avaliação de sistemas autoadaptativos identificadas na literatura. A Seção 6.4 apresenta o ambiente de coleta das medidas e o processo adotado em suas análises. Por fim, a Seção 6.5 exhibe a análise dos resultados relacionados as medidas de desempenho, quanto a sua subcaracterística, bem como dificuldades encontradas no processo de coleta no ambiente autoadaptativo.

### 6.1 Sistema autoadaptativo

Para esta pesquisa foi utilizado o sistema *Tele Assistance System (TAS)*. O TAS é um modelo de sistema autoadaptativo do domínio *healthcare* baseado no fornecimento de serviços, sendo eles fornecidos por terceiros contendo chances de serem compostos de forma dinâmica e dispondo de funcionalidades complexas (WEYNS; CALINESCU, 2015).

O conjunto de serviços apresentados pelo TAS são acionados por meio da seleção que variam de acordo com o atributo de qualidade selecionado como perfil de entrada em sua tela principal. É apresentado também, a escolha de desabilitar um serviço do perfil de entrada. Os perfis são classificados a seguir como: *ReliabilityQoS*, *CostQoS* e *PreferredQoS*. Quando selecionado um perfil de entrada, o sistema acionará o mecanismo de adaptação selecionado alterando seu trabalho e parâmetros para invocar os serviços recorrentes ao perfil selecionado (WEYNS; CALINESCU, 2015). Os mecanismos do TAS são divididos em quatro, o mecanismo de *No adaptation* não apresenta adaptações dos serviços quando executado, assim o fluxo normal do sistema é executado. Os mecanismos de *Model adaptation* e *Model Evolution* apresentam um modelo de recuperações de falhas, que é ativado quando uma falha acontece durante a execução dos serviços. Por fim, o mecanismo *Goal Management* tem por objetivo recuperar os serviços de todas que são encontradas pelos serviços, por este fator muita utilização de recursos pode ser observada até presenças de exceções.

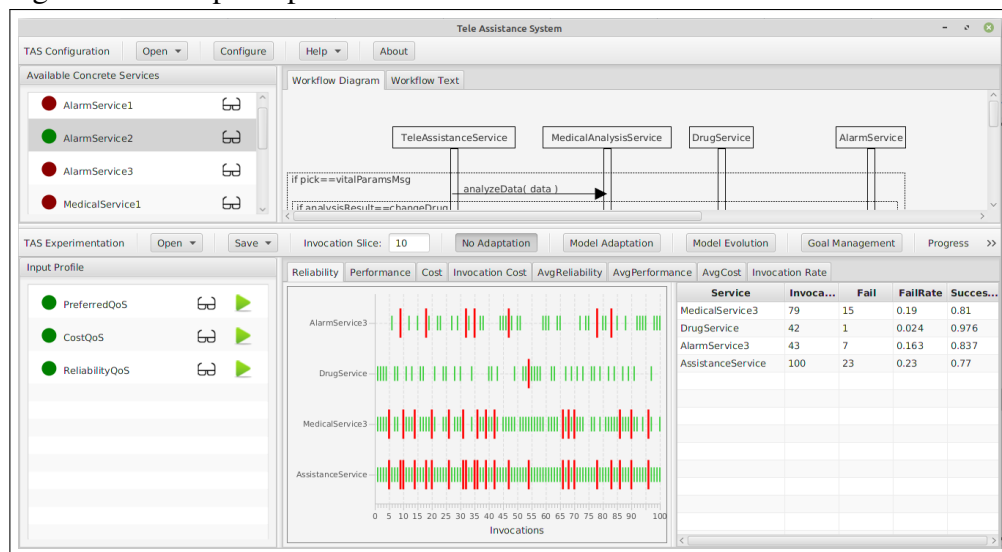
Esse sistema é bastante aceito pela comunidade de pesquisadores da área de sistemas autoadaptativos, e já foi explorado por avaliações de soluções nesse tema (BARESI *et al.*, 2007;

FILIERI *et al.*, 2012). Assim, por ser um sistema presente em estudos da comunidade e seu código ser acessível, o TAS foi escolhido para ser utilizado neste trabalho para avaliação do desempenho de sistema autoadaptativo.

O TAS pode ser apresentado como um sistema *desktop*, permitindo que funcione sem dependências externas, como acesso a internet. Seu código é desenvolvido na linguagem de programação Java em sua versão 8. Para *framework* de formatação e apresentação dos dados é utilizado o JavaFx presente no Java 8. O sistema também conta com a geração de relatórios, criados a cada verificação de perfil de entrega executada por seus usuários.

O sistema TAS possui duas versões, uma com mecanismos de adaptação mais simples e outra que os autores supõem ser mais eficaz, chamada de *ActivForms* (WEYNS; CALINESCU, 2015). Assim neste trabalho, foi optado por utilizar o projeto de versão com mecanismos de adaptação *ActivForms*. Pode-se visualizar uma visão da tela inicial do sistema na Figura 4.

Figura 4 – Tela principal de análise do sistema



Fonte: *Tele Assistance System* (TAS).

## 6.2 Ferramentas

O auxílio de ferramentas se faz necessário para pesquisa pela sua capacidade de agilizar algum processo que exige um certo esforço, diminuindo assim o tempo de conclusão de uma tarefa. Na área da computação é muito comum o uso de ferramentas para auxiliarem no processo monitoramento dos sistema fornecendo dados valiosos para construção de estratégia de melhoria dos softwares. Para SAS as ferramentas vão um pouco mais afundo, uma vez que a aplicação é monitoradas e seus dados podem ser avaliados mais rapidamente, permitindo a

montagem de uma estratégia mais eficaz e diminuindo os erros na adaptação de algum cenário crítico. Outra característica valiosa das ferramentas, são as finitas formas em que os dados podem ser distribuídos auxiliando no planejamento geral do sistema.

Para SAS é comum o uso de ferramentas para auxiliar no planejamento estratégico da adaptação e reconfiguração do sistema. No entanto, o conjunto de ferramentas que podem auxiliar SAS ainda é muito pequeno. Então, é comum a construção de um módulo ou implementação de uma ação adaptativa que é disparada quando um cenário crítico acontece, e são implementadas no próprio sistema. Outra abordagem aceita é uma adaptação de apenas de uma parte do sistema com mais valor de negócio ou mais acessada por parte dos usuários. Geralmente essa estratégia realiza operações de alocação de recursos do sistema.

No presente trabalho foi realizado uma pesquisa de ferramentas que possuem um processo de medição automatizado, ou que adotam uma adaptação do conjunto de medidas de desempenho identificadas no catálogo. Foram consideradas as ferramentas apresentadas nos estudos filtrados pela revisão da literatura apresentada na Seção 5.1. Dentre as ferramentas identificadas, a *Prometheus* apresentou um melhor plano de adaptação, permitindo inserir todo o número de medidas que fosse identificado pelo estudo fornecendo as informações para a medição de cada uma delas, além de utilizada por várias empresas.

Contudo, existe várias limitações do SAS escolhido para o estudo (descrito na Seção 6.1) em termos de desenvolvimento, organização do código, *frameworks* desatualizados e pouca utilização de padrões. No entanto, a principal limitação foi que o TAS por ser um sistema *desktop* restringiu o número de ferramentas para este tipo de ambiente. Assim como, o estudo de viabilidade da ferramenta *Prometheus* se destinava ao ambiente web, não sendo possível sua utilização neste trabalho. Como estratégia de remoção da dependência da ferramenta foi optado por implementar as medidas diretamente no código fonte do sistema.

### **6.3 Implementação das medidas de desempenho**

Como descrito na Seção 6.2, devido a falta de sucesso na identificação de ferramentas para análise das medidas de desempenho em sistemas autoadaptativos, para esta pesquisa optou-se pela implementação do conjunto de medidas diretamente no código do sistema. Para realização dessa abordagem, procurou-se sempre a melhor forma de se obter a medida dentro do sistema preservando a validade de seu resultado. Contudo, neste processo podem ser encontradas algumas dificuldades, como entendimento da lógica da aplicação em nível de código, organização do

projeto, falta de padrões, e trechos não refatorados.

Por ser um sistema presente na área de pesquisa, duas medidas de desempenho já estavam implementadas pelo TAS (e.g., tempo médio de resposta, tempo de invocação de método (no TAS é calculado por invocação de um serviço)). Para as medidas de desempenho que foram adicionadas no TAS, e as que deixaram de ser implementadas por algum motivo, foram criadas duas listas para apresentação. Para as medidas que foram adicionadas, será apresentada a medida seguida da descrição de como foi calculada ou interpretada dentro do sistema. Assim, para as medidas não implementadas pelo trabalho sua descrição indica o motivo de não ter sido adicionada ao sistema.

As medidas que foram adicionadas ao TAS podem ser visualizadas a seguir:

- M01 Tempo médio de resposta: embora seu valor seja calculado pelo sistema, foi observado o valor dessa medida foi fixada constante no momento da criação dos serviços. Para obter um resultado mais próximo do ideal, tanto para esta medida quanto para as que depende dela, foi implementado no sistema o tempo gasto em cada invocação de um serviço provido pelo sistema, e este valor atribuído a essa medida.
- M02 *Throughput*: é calculado pela média do tempo de resposta das invocações com sucesso de um serviço durante o processo de avaliação de um perfil de entrada, dividido pelo tempo de duração total das invocações em segundos.
- M03 Latência: é apresentada pelo valor da mediana do tempo de espera entre as chamadas para um mesmo serviço durante a avaliação de um perfil de entrada provido pelo TAS.
- M04 pLatencia: foi calculada após a execução dos módulos de adaptação do TAS, onde seu cálculo é realizado com a divisão dos tempos médios de respostas entre os modelos de adaptação e sem adaptação.
- M06 Influência do desempenho na adaptatividade: por ser calculada com um valor entre sim ou não, essa medida também é interpretada nos relatórios na alternância dos resultados dos modelos sem adaptação e o modelo com adaptação fornecido pelo TAS.
- M11 taxa de requisição, e M12 queda da taxa de requisição: podem ser avaliadas com base nas invocações realizadas aos serviços providos pelos perfis de entrada do TAS.

Para as medidas não implementadas pelo trabalho podem ser visualizadas a seguir:

- M05 Desempenho da qualidade de resposta: Não pode ser calculada por depender de um alto fluxo de execução dos perfis de entrada do TAS.
- M07 tempo de execução de método, e M08 tempo de invocação de método: por serem

baseadas em tempos relacionados a métodos de classes, não podemos garantir a validade dos dados uma vez que o TAS é um sistema orientado a serviço.

- M09 tempo de migração de uma instância, e M10 tempo de migração de uma classe: ambas são calculadas baseadas em tempo de migração dentro de uma rede. No entanto, o TAS não realiza transferências em rede.
- M13 utilização do *pool* de *thread*: A aplicação não apresenta uma maneira viável para o seu cálculo em seu código fonte.
- M15 tempo de conclusão de escrita, e M16 tempo de conclusão de leitura: o TAS apresenta apenas a criação e leitura de um arquivo *comma-separated values* (CSV), sendo que uma prioridade do sistema é que a manipulação de arquivos não se pode generalizar essas medidas em outros estudos.
- M17 tempo de inicialização, e M18 tempo de desligamento: ambas as medidas são calculadas em sistemas orientados a componentes, e sendo o TAS orientado a serviços não podemos garantir acurácia em seus resultados.

Para visualização das implementações realizadas foi disponibilizado o código do TAS com as mudanças realizadas, está disponível para o acesso na ferramenta *github*<sup>1</sup>, como forma de obter mais visualização o projeto está configurado como público, permitindo aos usuário examinarem e realizar o *download* do sistema.

## 6.4 Validação

Para validação das medidas do catálogo, o projeto TAS foi importado na *Integrated Development Environment* (IDE) Eclipse 2020 em uma máquina com o sistema operacional Linux Mint 20 Ulyana, com processador intel core i3-6100U e 8GB de *Random Access Memory* (RAM). O sistema é dividido em três projetos: “*ResearchServicePlataform*” contendo classes responsáveis pelo monitoramento dos serviços do sistema; “*TAS\_Gui*” contendo classes de visão e controle do sistema; e “*TeleAssistanceSystem*” contendo as classes de mecanismos de adaptação e os métodos de acionamento dos serviços.

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 teve duração média de 25 minutos, tendo seus perfis de entrada sendo executados 30 vezes seguidamente permitindo uma análise amplificada do desempenho dos serviços, após

<sup>1</sup> <https://github.com/MaikeBezerra/TAS-ActivFORMS>



várias invocações terem sido realizadas. Este número de execuções foi baseado em uma margem estatística onde aponta que 30 pode ser considerado uma margem aceitável para se alcançar acurácia do resultados. Para coleta dos dados, optou-se por escolher os serviços de “*MedicalService*” providos entre os perfis de entrada, por seus dados serem mais próximos dos serviços fornecidos em ambiente real. Além disso, um ponto importante a destacar é que não foi possível coletar as medidas do sistema utilizando o mecanismo *Goal Management*, pois sua execução disparou muitas exceções e o processo de adaptação entrou em *deadlock*.

Durante o processo de coleta foram realizados procedimentos específicos iguais para todos os mecanismos de adaptação, com o objetivo de manter um comportamento comum que permitisse a integridade na coleta dos dados necessários para o cálculo das medidas. Esses procedimentos foram:

- Acionamento do perfil de entrada *CostQoS*.
- Acionamento do perfil de entrada *CostQoS* em paralelo com o perfil de entrada. *ReliabilityQoS*.
- Acionamento do perfil de entrada *CostQoS* em paralelo com o perfil de entrada *PreferredQoS*.

Vale ressaltar, que durante o período da coleta foi priorizado o perfil de entrada *CostQoS*, uma vez que o propósito é maximizar o desempenho do sistema durante suas adaptações. Dessa forma, seria o perfil mais próximo para uma boa avaliação de desempenho. Assim, o perfil de entrada *CostQoS* está presente em todas as combinações das etapas de execução do sistema.

## 6.5 Resultados e análise dos dados

Para análise dos resultados obtidos serão distribuídos para cada subcaracterística de desempenho definido pela (ISO/IEC, 2011). Vale ressaltar que os resultados obtidos com as medidas não podem ser generalizados, já que a coleta foi realizada em um curto período de tempo.

Para verificar o comportamento relacionado ao tempo do TAS, pode-se visualizar a Tabela 7, na qual os valores apresentados para cada mecanismo serão reflexos da média total do processo de execuções dos serviços durante o período de avaliação. Foram utilizadas as medidas M01, M02, M3, M04, M05, M06. Para essas medidas as suas fórmulas de cálculo foram implementadas no próprio SAS. A seguir pode-se visualizar o processo de implementação das medidas e os valores obtidos no processo de verificação no sistema.

Tabela 7 – Resultados da coleta e cálculo das medidas de comportamento em relação ao tempo.

Subcaracterística	Medida	Resultados por Mecanismos de Adaptação		
		Sem adaptação	<i>Model Adaptation</i>	<i>Model Evolution</i>
Comportamento em relação ao tempo	M1	0,66 ms	10,83 ms	12,07 ms
	M2	1978,19 r/s	79,23 r/s	71,21 r/s
	M3	49,52 ms	0,98 s	1,95 s
	M4	-	2,56 s	2,75 s

Fonte: dados da avaliação de desempenho.

A medida M1 (Tempo de resposta) teve seu melhor resultado no mecanismo de adaptação *No Adaptation* com uma media de (0,66 ms), sendo uma medida interpretada como, quanto menor seu valor, mais rápido a aplicação realiza uma resposta ao usuário. Sendo ela mais alta nos mecanismos que sofrem alguma adaptação, pois realizam mais processamento em suas requisições aumentando o tempo para retornar uma resposta ao usuário. Outro fator que pode acrescentar no tempo de resposta, são as tentativas que recuperação de falhas implantadas por sistemas autoadaptativos.

A medida M2 (*Throughput*), mostra a proporção média de invocações com sucesso de um sistema em um determinado tempo de execução do serviço medido em segundos. Como o mecanismo de *No adaptation* obteve um valor muito além do esperado, foi considerado como melhor valor avaliado a execução do mecanismo *Model Adaptation* atingindo uma taxa media de (79,23 r/s) requisições com sucessos sobre o tempo total de sua execução. Assim, pode-se concluir que esse mecanismos de adaptação conseguiu se recuperar melhor das falhas que ocorrem em seu processo de execução.

A medida M3 (Latência) é avaliada pelo tempo que o TAS leva para atender uma requisição para o sistema. Como no ambiente de testes são realizados varias seções de execução para garantir a assertividade de desempenho, o tempo de latência é calculado como tempo total gasto sobre o número de execuções. Assim, o melhor resultado apresentado foi do mecanismos *No Adaptation* com (49,52 ms), e para os mecanismos de adaptação o *Model Adaptation* teve uma latência de (0,98 s) reafirmando que os modelos tendem a gastar mais tempo para atender as requisições por conta das tentativas de se readaptar em tempo de execução.

A medida M4 (pLatencia), baseada nos tempos de respostas entre um mecanismos sem adaptação e um com adaptação, reflete o valor de latência aproximadamente esperado para o mecanismo de adaptação comparado. Assim o resultado entre o mecanismo de adaptação *No Adaptation* e o mecanismo *Model Adaptation* obteve uma avaliação media de (2,56s), um pouco menor do que comparada ao mecanismo de *Model Evolution* que obteve (2,75s). O que nos

mostra que os mecanismos de adaptação estão bem próximos em suas avaliações.

A medida M6 (Influência de desempenho na adaptatividade) representa uma medida interpretativa entre sim, para responder que o desempenho é influenciado na presença da adaptatividade, e não para responder que não é influenciada. Nossa análise foi realizada baseada nos resultados das medidas M1, M2, M3 e M4 indicando que o desempenho é sim influenciado na presença da adaptatividade, porém nem sempre de forma positiva, como é o caso dos valores relacionados ao tempo. Geralmente tendem a sofrer algum acréscimo por conta das adaptações sofridas durante a execução do serviço.

Para análise relacionada a subcaracterística de utilização de recursos, foram utilizadas as medidas M11, M12 presentes na Tabela 8, e M18. Para validação da medida M18 (utilização de CPU) no trabalho ao qual foi extraída a medida, o autor utiliza a ferramenta de desempenho do sistema operacional Microsoft Windows. No entanto, como o ambiente de execução deste trabalho é o Linux Mint, foi utilizado a ferramenta de monitor do sistema mostrada na Figura 5. Durante a etapa de coleta de dados as três abordagens tiveram comportamento semelhante e nenhuma delas apresentou um alto consumo do processamento de CPU.

Tabela 8 – Resultados da coleta e cálculo das medidas de utilização de recursos.

Subcaracterística	Medida	Resultados por Mecanismos de Adaptação		
		Sem adaptação	<i>Model Adaptation</i>	<i>Model Evolution</i>
Utilização de recursos	M11	2.402,76	97,41	86,28
	M12	18,31%	18,53%	17,42%

Fonte: dados da avaliação de desempenho.

Figura 5 – Monitor do sistema para verificação de recursos



Fonte: Linux Mint 20 Ulyana.

A medida M11 (Taxa de requisição) consegue mostrar uma média de quantas requisições por segundo um servidor poderia suportar para prover seu serviço com uma certa qualidade. Pelo fato do mecanismo *No Adaptation* apresentar uma média de 2.402,76 requisições, seria injusto compararmos aos mecanismos de adaptação. Assim, foi considerado então o melhor resultado sendo do mecanismo *Model Adaptation* com uma média de 97,41 requisições por segundo. Conclui-se que os mecanismos de adaptação sempre irão prover menos requisições devido ao seus tempos de resposta serem maiores.

A medida M12 (Queda da taxa de requisição) irá refletir com relação a medida M11 o quão os mecanismos atendem as suas requisições e conseqüentemente gera menos falhas. Para esta medida se saiu melhor o mecanismo de *Model Evolution* com uma taxa de 17,42% de perda em suas requisições, um pouco menor em relação ao *Model Adaptation* com 18,53%. Para o mecanismo *No Adaptation* houve uma queda de 18,31%, chegando a ser menor que um modelo de adaptação. Porém, tem-se de levar em consideração que o número de invocações do mecanismo sem adaptação sempre será em maior que os outros. Assim, sempre o número de falhas tende a ser maior. Também pode-se concluir que os mecanismos de adaptação gerenciam melhor suas invocações e provocam menos falhas, mas requerem mais tempo de processamento.

Para a capacidade caracterizada como última subcaracterística de eficiência em desempenho, não foram encontradas medidas equivalentes na literatura. Existe uma lacuna na literatura de avaliação de SAS baseado na sua capacidade. Outro fator que também pode influenciar é a evolução tecnológica que vem alimentando constantemente a capacidade dos recursos computacionais, assim gerando sua despriorização.

Com base nos resultados das medidas são disponibilizadas em uma planilha <sup>2</sup>, foi possível identificar pontos fortes e fracos nos mecanismos de adaptação do TAS. Por exemplo, para as medidas referentes ao tempo o *Model Adaptation* apresentou melhores resultados, e para análise de recuperação de falhas o mecanismo *Model Evolution* alcançou melhores resultados.

De modo geral, considerando os resultados obtidos por meio da implementação de medidas de desempenho propostas neste trabalho, o sistema TAS é considerado confiável, apresentando valores próximos do cenário real de uma aplicação. Entretanto, é necessário o uso de mais medidas de desempenho principalmente para subcaracterísticas de utilização de recursos e capacidade.

---

<sup>2</sup> [https://docs.google.com/spreadsheets/d/1XprHLCNkC1W7DPkcalqCwrWcHU7oOBbfd9JUj18\\_8wg/edit#gid=1253987047](https://docs.google.com/spreadsheets/d/1XprHLCNkC1W7DPkcalqCwrWcHU7oOBbfd9JUj18_8wg/edit#gid=1253987047)

## 7 CONCLUSÕES E TRABALHOS FUTUROS

Sistemas autoadaptativos são sistemas de software que se adaptam em tempo de execução para garantirem a qualidade na entrega de seus serviços (BOWERS *et al.*, 2018). Os SAS são sistemas que precisam garantir vários aspectos de qualidade (e.g., disponibilidade, segurança, confiabilidade, desempenho), que são impostos sobre a alta demanda diária dos fluxos de informações trafegados por eles (e.g., sistemas críticos, cidades inteligentes) (WEYNS, 2019).

Pelo fato de conter essa característica adaptativa, os SAS são considerados sistemas complexos que, mesmo após suas mudanças dinâmicas, precisam garantir sua permanência de qualidade, para atender sempre futuras modificações que não foram identificadas em tempo de projeto, resultando no menor número de falhas.

Visando a preservação da qualidade do desempenho em sistemas autoadaptativos, neste trabalho foi desenvolvido primeiramente um revisão da literatura de forma estruturada com o objetivo de identificar as subcaracterísticas e medidas de desempenho que têm sido utilizadas na avaliação desses sistemas. Para essa revisão foi consideradas as subcaracterísticas de eficiência de desempenho apresentadas pela ISO (ISO/IEC, 2011). Entre os resultados da revisão se destacaram as subcaracterísticas de comportamento em relação ao tempo e utilização de recursos como as mais importantes pelos os estudos para SAS, identificadas em um total de 32 estudos primários.

Na análise dos estudos selecionados pela revisão, a subcaracterística de comportamento em relação ao tempo apresentou medidas responsáveis por medir tempos gastos em operações realizadas pelo sistema. Na utilização de recursos as medidas apresentadas tem o objetivo de mostrar tanto os recursos que são utilizados pelo SAS em seus ambientes, como o quanto podem ser eficiente o seu gerenciamento. Para a capacidade, não foram identificadas medidas.

Foi identificado um conjunto de medidas de desempenho para SAS contendo 24 medidas distribuídas nas subcaracterísticas: comportamento em relação ao tempo e utilização de recursos. O conjunto é composto por medidas existentes e medidas do contexto adaptativo.

O processo de filtragem das medidas teve o objetivo de descartar as medidas que não podiam ser implementadas nesta pesquisa. Obteve-se um conjunto final de 18 medidas de desempenho. Em seguida, foi avaliado o subconjunto de medidas que poderiam ser implementadas pelo TAS resultando em 7 medidas, e para aquelas que não puderam ser implementadas foi justificado o motivo.

A avaliação do TAS mostrou que seu mecanismo *No Adaption* tem as medidas relacionadas ao tempo, sendo executadas mais rapidamente, mas apresenta um grande número de falhas com relação ao total de invocações realizadas. Para os mecanismos com adaptação *Model Adaption* e *Model Evolution* as medidas de tempo recebem um acréscimo, relacionado diretamente com seu módulo de adaptações que precisam realizar mais processamento de dados, além de tentar se recuperar de falhas que podem ocorrer durante uma requisição ao serviço. Assim, é concluído que embora se espere mais tempo para os mecanismos de adaptação realizarem uma resposta, eles apresentaram um entrega de valor maior ao usuário final.

Durante o desenvolvimento do trabalho foram identificadas duas dificuldades principais: (i) o desenvolvimento de cálculo das fórmulas diretamente no código do TAS, que exigiu um grande estudo sobre a aplicação; e, (ii) a identificação de ferramentas que trabalhasse junto com o TAS para automação das medidas. Com base nas contribuições deste trabalho, espera-se que ele contribua com outros trabalhos relacionados na avaliação de eficiência em desempenho dos sistemas autoadaptativos, não limitando-se ao TAS como sistema. Tendo também as medidas que podem ser utilizadas em vários ambientes de sistemas de software.

Como trabalhos futuros pode-se sugerir: (i) desenvolver ou avaliar ferramentas que ajudem na avaliação de sistemas autoadaptativos, uma vez que implementá-las diretamente no sistema poderá influenciar no desempenho do sistema de maneira não intencional, e assim influenciando nos resultados; (ii) realizar avaliações de desempenho de sistemas autoadaptativos em outros ambientes aumentando o nível de generalização das medidas, e seguindo esse ponto pode ser interessante para a avaliação de mais de um sistema, permitindo fazer comparações entre eles assim gerando mais conteúdo para o trabalho; e, (iii) avaliar um conjunto de medidas de desempenho analisando as que não podem ser implementadas por este trabalho, bem como outras para as subcaracterísticas em apresentaram poucas medidas nesta pesquisa.

## REFERÊNCIAS

- ALFÉREZ, G. H.; PELECHANO, V.; MAZO, R.; SALINESI, C.; DIAZ, D. Dynamic adaptation of service compositions with variability models. **Journal of Systems and Software**, v. 91, p. 24–47, 2014. ISSN 0164-1212. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0164121213001465>. Acesso em: 8 set. 2020.
- ALRIFAI, M.; RISSE, T. Combining global optimization with local selection for efficient qos-aware service composition. In: **Proceedings of the 18th International Conference on World Wide Web**. New York, USA: Association for Computing Machinery, 2009. p. 881–890. ISBN 9781605584874. Disponível em: <https://dl.acm.org/doi/10.1145/1526709.1526828>. Acesso em: 26 set. 2020.
- BARESI, L.; BIANCULLI, D.; GHEZZI, C.; GUINEA, S.; SPOLETINI, P. Validation of web service compositions. **IET software**, Institution of Engineering and Technology, v. 1, n. 6, p. 219–232, 2007. ISSN 1751-8806. Disponível em: [https://digital-library.theiet.org/content/journals/10.1049/iet-sen\\_20070027](https://digital-library.theiet.org/content/journals/10.1049/iet-sen_20070027). Acesso em: 29 ago. 2020.
- BERNARDI, S.; MERSEGUER, J.; PETRIU, D. C. A dependability profile within marte. **Software & Systems Modeling**, Springer, v. 10, n. 3, p. 313–336, 2011. ISSN 1619-1374. Disponível em: <https://link.springer.com/article/10.1007/s10270-009-0128-1>. Acesso em: 14 set. 2020.
- BOWERS, K. M.; FREDERICKS, E. M.; CHENG, B. H. C. Automated optimization of weighted non-functional objectives in self-adaptive systems. In: **Search-Based Software Engineering**. Cham, Suíça: Springer, 2018. p. 182–197. ISBN 9783319992419. Disponível em: [https://doi.org/10.1007/978-3-319-99241-9\\_9](https://doi.org/10.1007/978-3-319-99241-9_9). Acesso em: 20 abr. 2019.
- CALINESCU, R.; GRUNSKÉ, L.; KWIATKOWSKA, M.; MIRANDOLA, R.; TAMBURRELLI, G. Dynamic qos management and optimization in service-based systems. **IEEE Transactions on Software Engineering**, IEEE, v. 37, n. 3, p. 387–409, 2011. ISSN 1939-3520. Disponível em: <https://ieeexplore.ieee.org/document/5611553>. Acesso em: 20 abr. 2019.
- CÁMARA, J.; GARLAN, D.; SCHMERL, B.; PANDEY, A. Optimal planning for architecture-based self-adaptation via model checking of stochastic games. In: **Proceedings of the 30th Annual ACM Symposium on Applied Computing**. Nova York, USA: Association for Computing Machinery, 2015. (SAC '15), p. 428–435. ISBN 9781450331968. Disponível em: <https://dl.acm.org/doi/10.1145/2695664.2695680>. Acesso em: 16 set. 2020.
- CÁMARA, J.; LEMOS, R. de; VIEIRA, M.; ALMEIDA, R.; VENTURA, R. Architecture-based resilience evaluation for self-adaptive systems. **Computing**, Springer, v. 95, n. 8, p. 689–722, 2013. ISSN 1436-5057. Disponível em: <https://link.springer.com/article/10.1007%2Fs00607-013-0311-7>. Acesso em: 15 set. 2020.
- CAPORUSCIO, M.; Di Marco, A.; INVERARDI, P. Model-based system reconfiguration for dynamic performance management. **Journal of Systems and Software**, v. 80, n. 4, p. 455–473, 2007. ISSN 0164-1212. Software Performance. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0164121206002068>. Acesso em: 13 set. 2020.
- CARVALHO, R. M.; ANDRADE, R. M. de C.; OLIVEIRA, K. M. de. 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. ISSN 0164-1212. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0164121217302728>. Acesso em: 11 set. 2020.

CHEN, B.; PENG, X.; ZHAO, W. Towards runtime optimization of software quality based on feedback control theory. In: **Proceedings of the First Asia-Pacific Symposium on Internetware**. New York, NY, USA: Association for Computing Machinery, 2009. (Internetware '09), p. 1–8. ISBN 9781605588728. Disponível em: <https://doi.org/10.1145/1640206.1640216>. Acesso em: 20 abr. 2019.

COMPUTING, A. *et al.* An architectural blueprint for autonomic computing. **IBM White Paper**, Citeseer, v. 31, n. 2006, p. 1–6, 2006.

CORDY, M.; CLASSEN, A.; HEYMANS, P.; LEGAY, A.; SCHOBENS, P.-Y. Model checking adaptive software with featured transition systems. In: **Assurances for Self-Adaptive Systems: Principles, Models, and Techniques**. Berlin, Heidelberg: Springer, 2013. p. 1–29. ISBN 9783642362491. Disponível em: [https://doi.org/10.1007/978-3-642-36249-1\\_1](https://doi.org/10.1007/978-3-642-36249-1_1). Acesso em: 20 abr. 2019.

CRUZ, R. A. S.; NUNES, M. S.; MENEZES, L.; DOMINGUES, J. Iptv architecture for an ims environment with dynamic qos adaptation. **Multimedia Tools and Applications**, Springer, v. 53, n. 3, p. 557–589, 2011. ISSN 1573-7721. Disponível em: <https://link.springer.com/article/10.1007%2Fs11042-010-0537-8>. Acesso em: 15 set. 2020.

DYBA, T.; DINGSOYR, T.; HANSEN, 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. ISBN 9780769528861. ISSN 1949-3770. Disponível em: <https://ieeexplore.ieee.org/document/4343750/>. Acesso em: 21 set. 2020.

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)**. Budapest, Hungary: IEEE, 2016. p. 2814–2820. ISBN 9781509018970. Disponível em: <https://doi.org/10.1109/SMC.2016.7844665>. Acesso em: 20 abr. 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. ISBN 9781447122975. Disponível em: [https://link.springer.com/chapter/10.1007%2F978-1-4471-2297-5\\_12](https://link.springer.com/chapter/10.1007%2F978-1-4471-2297-5_12). Acesso em: 10 set. 2020.

GILL, S. S.; BUYYA, R. Resource provisioning based scheduling framework for execution of heterogeneous and clustered workloads in clouds: from fundamental to autonomic offering. **Journal of Grid Computing**, Springer, v. 17, n. 3, p. 385–417, 2019. ISSN 1572-9184. Disponível em: <https://link.springer.com/article/10.1007%2Fs10723-017-9424-0>. Acesso em: 12 set. 2020.

GOLDSCHMIDT, T.; HAUCK-STATTELMANN, S.; MALAKUTI, S.; GRÜNER, S. Container-based architecture for flexible industrial control applications. **Journal of Systems Architecture**, v. 84, p. 28–36, 2018. ISSN 1383-7621. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1383762117304988>. Acesso em: 9 set. 2020.

GOUVÊA, D. D.; MUNIZ, C. d. A. A. D.; PINTO, G. A.; AVRITZER, A.; LEAO, R. M. M.; SILVA, E. d. S. e; DINIZ, M. C.; CORTELLESA, V.; BERARDINELLI,



L.; LEITE, J. C. *et al.* Experience with model-based performance, reliability, and adaptability assessment of a complex industrial architecture. **Software & Systems Modeling**, Springer, v. 12, n. 4, p. 765–787, 2013. ISSN 1619-1374. Disponível em: <https://link.springer.com/article/10.1007%2Fs10270-012-0264-x>. Acesso em: 14 set. 2020.

GRASSI, V.; MIRANDOLA, R.; RANDAZZO, E. Model-driven assessment of qos-aware self-adaptation. In: CHENG, B. H. C.; LEMOS, R. de; GIESE, H.; INVERARDI, P.; MAGEE, J. (Ed.). **Software Engineering for Self-Adaptive Systems**. Berlin, Heidelberg: Springer, 2009. p. 201–222. ISBN 9783642021619. 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: 25 abr. 2019.

HAGGUI, H.; AFFES, S.; BELLILI, F. Fpga-sdr integration and experimental validation of a joint da ml snr and doppler spread estimator for 5g cognitive transceivers. **IEEE Access**, IEEE, v. 7, p. 69464–69480, 2019. ISSN 2169-3536. Disponível em: <https://ieeexplore.ieee.org/document/8726401>. Acesso em: 16 set. 2020.

HIRAMA, K. **Engenharia de software: qualidade e produtividade com tecnologia**. Brasil: Elsevier, 2012.

ISO/IEC. **ISO/IEC 9126-1**: Software engineering - product quality. [S. l.]: ISO, 2001.

ISO/IEC. **ISO-IEC 25010**: Systems and software engineering-systems and software quality requirements and evaluation (square)-system and software quality models. [S. l.]: ISO, 2011.

KADDOUM, E.; RAIBULET, C.; GEORGÉ, J.-P.; PICARD, G.; GLEIZES, M.-P. Criteria for the evaluation of self-\* systems. In: **Proceedings of the 2010 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)**. Nova York, USA: ACM, 2010. p. 29–38. ISBN 9781605589718. Disponível em: <https://dl.acm.org/doi/10.1145/1808984.1808988>. Acesso em: 25 abr. 2019.

KALAVATHY, G. M.; RATHINAM, N. E.; SEETHALAKSHMI, P. Self-adaptable media service architecture for guaranteeing reliable multimedia services. **Multimedia Tools and Applications**, Springer, v. 57, n. 3, p. 633–650, 2012. ISSN 1573-7721. Disponível em: <https://link.springer.com/article/10.1007/s11042-010-0664-2>. Acesso em: 12 set. 2020.

KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing systematic literature reviews in software engineering**. [S. l.], 2007. v. 2.

KOSCIANSKI, A.; SOARES, M. dos S. **Qualidade de Software: Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software**. [S. l.]: Novatec Editora, 2007. v. 2. ISBN 9788575221129.

LAU, G.; AL-SABAH, M.; JASEEMUDDIN, M.; RAZAVI, H.; BHUIYAN, M. Context-aware raon middleware for opportunistic network. **Pervasive and Mobile Computing**, v. 41, p. 28–45, 2017. ISSN 1574-1192. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1574119216303558>. Acesso em: 6 set. 2020.

LE MOS, R. D.; GARLAN, D.; GHEZZI, C.; GIESE, H.; ANDERSSON, J.; LITOIU, M.; SCHMERL, B.; WEYNS, D.; BARESI, L.; BENCOMO, N. *et al.* Software engineering for self-adaptive systems research challenges in the provision of assurances. In: LEMOS, R. de; GARLAN, D.; GHEZZI, C.; GIESE, H. (Ed.). **Software Engineering for Self-Adaptive Systems III. Assurances**. Cham, Suíça: Springer, 2017. p. 3–30. ISBN 9783319741833.

Disponível em: [https://link.springer.com/chapter/10.1007/978-3-319-74183-3\\_1](https://link.springer.com/chapter/10.1007/978-3-319-74183-3_1). Acesso em: 14 set. 2020.

LEMOS, R. de; GIESE, H.; BENCOMO, N.; CUKIC, B.; MÜLLER, H.; WEYNS, D. Software engineering for self-adaptive systems: A second research road map. In: LEMOS, R. de; GIESE, H.; MÜLLER, H. A.; SHAW, M. (Ed.). **Software Engineering for Self-Adaptive Systems II: International seminar, dagstuhl castle, germany, october 24-29, 2010 revised selected and invited papers**. Berlin, Heidelberg: Springer, 2013. p. 1–32. ISBN 9783642358135. Disponível em: [https://link.springer.com/chapter/10.1007%2F978-3-642-35813-5\\_1](https://link.springer.com/chapter/10.1007%2F978-3-642-35813-5_1). Acesso em: 25 abr. 2019.

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, Suíça: Springer, 2014. p. 121–130. ISBN 9783319112985. Disponível em: [https://link.springer.com/chapter/10.1007/978-3-319-11298-5\\_13](https://link.springer.com/chapter/10.1007/978-3-319-11298-5_13). Acesso em: 25 abr. 2019.

LUNG, C.-H.; ZHANG, X.; RAJESWARAN, P. Improving software performance and reliability in a distributed and concurrent environment with an architecture-based self-adaptive framework. **Journal of Systems and Software**, v. 121, p. 311–328, 2016. ISSN 0164-1212. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0164121216300991>. Acesso em: 2 set. 2020.

MAALOUL, S.; AFIF, M.; TABBANE, S. Handover decision in heterogeneous networks. **2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)**, IEEE, Crans-Montana, Switzerland, p. 588–595, 2016. ISSN 1550-445X. Disponível em: <https://ieeexplore.ieee.org/document/7474142>. Acesso em: 12 set. 2020.

MARCO, A. D.; INVERARDI, P.; SPALAZZESE, R. Synthesizing self-adaptive connectors meeting functional and performance concerns. **Proceedings of the 8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)**, IEEE, San Francisco, California, p. 133–142, 2013. ISSN 2157-2321. Disponível em: <https://ieeexplore.ieee.org/document/6595500>. Acesso em: 12 set. 2020.

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, 2010. p. 51–66. ISBN 9783642157844. Disponível em: [https://link.springer.com/chapter/10.1007/978-3-642-15784-4\\_4](https://link.springer.com/chapter/10.1007/978-3-642-15784-4_4). Acesso em: 25 abr. 2019.

MCBRIDE, T.; LEPMETS, M. Quality assurance in agile safety-critical systems development. In: **2016 10th International Conference on the Quality of Information and Communications Technology (QUATIC)**. Lisboa, Portugal: IEEE, 2016. p. 44–51. ISBN 9781509035823. Disponível em: <https://ieeexplore.ieee.org/document/7814513>. Acesso em: 30 abr. 2019.

MERTZ, J.; NUNES, I. On the practical feasibility of software monitoring: A framework for low-impact execution tracing. **2019 IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)**, IEEE, Montreal, Quebec, Canada, p. 169–180, 2019. ISSN 2157-2321. Disponível em: <https://ieeexplore.ieee.org/document/8787058>. Acesso em: 13 set. 2020.

MOUSA, A.; BENTAHAR, J.; ALAM, O. Context-aware composite saas using feature model. In: **Future Generation Computer Systems**. [S. n.], 2019. v. 99, p. 376–390. ISSN 0167-739X. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167739X1832507X>.

NOBRE, J. C.; DUARTE, P. A. P. R.; GRANVILLE, L. Z.; TAROUCO, L. M. R. Self-\* properties and p2p technology on disruption-tolerant management. **2013 IEEE Symposium on Computers and Communications (ISCC)**, IEEE, Split, Croácia, p. 676–681, 2013. ISSN 1530-1346. Disponível em: <https://ieeexplore.ieee.org/abstract/document/6755026>. Acesso em: 30 abr. 2019.

NOSTRO, N.; SPALAZZESE, R.; GIANDOMENICO, F. D.; INVERARDI, P. Achieving functional and non functional interoperability through synthesized connectors. **Journal of Systems and Software**, v. 111, p. 185–199, 2016. ISSN 0164-1212. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0164121215002149>. Acesso em: 4 set. 2020.

OLIVEIRA, N.; BARBOSA, L. S. Self-adaptation by coordination-targeted reconfigurations. **Journal of Software Engineering Research and Development**, Springer, v. 3, n. 1, p. 1–31, 2015. ISSN 2195-1721. Disponível em: <https://link.springer.com/article/10.1186/s40411-015-0021-2>. Acesso em: 11 set. 2020.

Oreizy, P.; Gorlick, M. M.; Taylor, R. N.; Heimhigner, D.; Johnson, G.; Medvidovic, N.; Quilici, A.; Rosenblum, D. S.; Wolf, A. L. An architecture-based approach to self-adaptive software. **IEEE Intelligent Systems and their Applications**, IEEE, v. 14, n. 3, p. 54–62, 1999. ISSN 2374-9423. Disponível em: <https://ieeexplore.ieee.org/document/769885>. Acesso em: 10 set. 2020.

ORIOLO, M.; FRANCH, X.; MARCO, J. Monitoring the service-based system lifecycle with salmon. **Expert Systems with Applications**, v. 42, n. 19, p. 6507 – 6521, 2015. ISSN 0957-4174. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0957417415002213>. Acesso em: 5 set. 2020.

PEREZ-PALACIN, D.; MIRANDOLA, R.; MERSEGUER, J. On the relationships between qos and software adaptability at the architectural level. **Journal of Systems and Software**, Elsevier, v. 87, p. 1–17, 2014. ISSN 0164-1212. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0164121213001982>. Acesso em: 30 abr. 2019.

PERUMA, A.; KRUTZ, D. Security: a critical quality attribute in self-adaptive systems. **2018 IEEE/ACM 13th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)**, IEEE, Gotemburgo, Suécia, p. 188–189, 2018. ISSN 2157-2305. Disponível em: <https://ieeexplore.ieee.org/document/8595396>. Acesso em: 30 abr. 2019.

Perveen, A.; Patwary, M.; Aneiba, A. Dynamically reconfigurable slice allocation and admission control within 5g wireless networks. **2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)**, IEEE, Kuala Lumpur, Malásia, p. 1–7, 2019. ISSN 2577-2465. Disponível em: <https://ieeexplore.ieee.org/document/8746625>. Acesso em: 14 set. 2020.

PETROVIC, N.; TOSIC, M. Smada-fog: Semantic model driven approach to deployment and adaptivity in fog computing. **Simulation Modelling Practice and Theory**, Elsevier, v. 101, p. 102033, 2020. ISSN 1569-190X. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1569190X19301649>. Acesso em: 6 set. 2020.

PHAM, L. M.; PHAM, T.-M. Autonomic fine-grained replication and migration at component level on multicloud. **Vietnam Journal of Computer Science**, Springer, v. 4, n. 1, p. 39–49, 2017. ISSN 2196-8896. Disponível em: <https://link.springer.com/article/10.1007/s40595-016-0074-0>. Acesso em: 16 set. 2020.

POTENA, P. Optimization of adaptation plans for a service-oriented architecture with cost, reliability, availability and performance tradeoff. **Journal of Systems and Software**, Elsevier, v. 86, n. 3, p. 624–648, 2013. ISSN 0164-1212. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0164121212003226>. Acesso em: 5 mai. 2019.

RAFIQUE, A.; LANDUYT, D. V.; TRUYEN, E. *et al.* Scope: self-adaptive and policy-based data management middleware for federated clouds. **Journal of Internet Services and Applications**, Springer, 2019. ISSN 1869-0238. Disponível em: <https://link.springer.com/article/10.1186/s13174-018-0101-8>. Acesso em: 12 set. 2020.

RAIBULET, C. Hints on quality evaluation of self-systems. In: **2014 IEEE Eighth International Conference on Self-Adaptive and Self-Organizing Systems**. Londres, Inglaterra: IEEE, 2014. p. 185–186. ISSN 1949-3681. Disponível em: <https://ieeexplore.ieee.org/document/7001019>. Acesso em: 5 mai. 2019.

RAIBULET, C.; Arcelli Fontana, F.; CAPILLA, R.; CARRILLO, C. An overview on quality evaluation of self-adaptive systems. In: MISTRİK, 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 9780128028551. Disponível em: <http://www.sciencedirect.com/science/article/pii/B9780128028551000137>. Acesso em: 5 mai. 2019.

RAIBULET, C.; FONTANA, F. A. Evaluation of self-adaptive systems: A women perspective. In: **Proceedings of the 11th European Conference on Software Architecture: Companion proceedings**. Nova York, USA: Association for Computing Machinery, 2017. p. 23–30. ISBN 9781450352178. Disponível em: <https://dl.acm.org/doi/10.1145/3129790.3129825>. Acesso em: 24 set. 2020.

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, Inglaterra: IEEE, 2009. p. 341–344. ISBN 9781424449842. Disponível em: <https://ieeexplore.ieee.org/document/5290667>. Acesso em: 5 mai. 2019.

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**. Atenas, Grécia: IEEE, 2009. p. 146–152. ISBN 9781424451661. Disponível em: <https://ieeexplore.ieee.org/document/5359571>. Acesso em: 10 mai. 2019.

RAMIREZ, A. J.; KNOESTER, D. B.; CHENG, B. H.; MCKINLEY, P. K. Plato: a genetic algorithm approach to run-time reconfiguration in autonomic computing systems. In: **Cluster Computing: The Journal of Networks, Software Tools and Applications**. Springer, 2011. v. 14, n. 3, p. 229–244. ISSN 1573-7543. Disponível em: <https://link.springer.com/article/10.1007/s10586-010-0122-y>. Acesso em: 14 set. 2020.

RATTANI, A.; MARCIALIS, G. L.; ROLI, F. Self adaptive systems: An experimental analysis of the performance over time. **2011 IEEE Workshop on Computational Intelligence in Biometrics and Identity Management (CIBIM)**, IEEE, Paris, França, p. 36–43, 2011. ISSN 2325-4319. Disponível em: <https://ieeexplore.ieee.org/document/5949222>. Acesso em: 10 mai. 2019.

ROLI, F.; DIDACI, L.; MARCIALIS, G. L. Adaptive biometric systems that can improve with use. In: RATHA, N. K.; GOVINDARAJU, V. (Ed.). **Advances in Biometrics: Sensors, algorithms and systems**. Londres, Inglaterra: Springer, 2008. p. 447–471. ISBN 9781846289217. Disponível em: [https://link.springer.com/chapter/10.1007/978-1-84628-921-7\\_23](https://link.springer.com/chapter/10.1007/978-1-84628-921-7_23). Acesso em: 10 mai. 2019.

Ryan, C.; Rossi, P. Software, performance and resource utilisation metrics for context-aware mobile applications. **11th IEEE International Software Metrics Symposium**, IEEE, Como, Itália, p. 10–12, 2005. ISSN 1530-1435. Disponível em: <https://ieeexplore.ieee.org/document/1509290>. Acesso em: 15 mai. 2019.

SALAÜN, G.; ETCHEVERS, X.; PALMA, N. D.; BOYER, F.; COUPAYE, T. Verification of a self-configuration protocol for distributed applications in the cloud. In: CÂMARA, J.; LEMOS, R. de; GHEZZI, C.; LOPES, A. (Ed.). **Assurances for Self-Adaptive Systems: Principles, models, and techniques**. Berlin, Heidelberg: Springer, 2013. p. 60–79. ISBN 9783642362491. Disponível em: [https://link.springer.com/chapter/10.1007/978-3-642-36249-1\\_3](https://link.springer.com/chapter/10.1007/978-3-642-36249-1_3). Acesso em: 15 mai. 2019.

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**, Springer, v. 3, n. 1, p. 10, 2015. ISSN 2195-1721. Disponível em: <https://link.springer.com/article/10.1186/s40411-015-0022-1>. Acesso em: 15 mai. 2019.

SANTOS, L. D. V.; OLIVEIRA, C. V. de S. **Introdução à garantia de qualidade de software**. São Paulo: Cia do eBook, 2017. ISBN 9788555850882.

SILVANO, C.; AGOSTA, G.; BARTOLINI, A.; BECCARI, A. R.; BENINI, L.; BESNARD, L.; BISPO, J.; CMAR, R.; CARDOSO, J. M.; CAVAZZONI, C.; CESARINI, D.; CHERUBIN, S.; FICARELLI, F.; GADIOLI, D.; GOLASOWSKI, M.; LIBRI, A.; MARTINOVIČ, J.; PALERMO, G.; PINTO, P.; ROHOU, E.; SLANINOVÁ, K.; VITALI, E. The antarex domain specific language for high performance computing. **Microprocessors and Microsystems**, v. 68, p. 58 – 73, 2019. ISSN 0141-9331. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0141933119300353>. Acesso em: 5 set. 2020.

SIRONI, F.; BARTOLINI, D. B.; CAMPANONI, S.; CANCARE, F.; HOFFMANN, H.; SCIUTO, D.; SANTAMBROGIO, M. D. Metronome: operating system level performance management via self-adaptive computing. In: **Proceedings of the 49th Annual Design Automation Conference**. Nova York, USA: Association for Computing Machinery, 2012. p. 856–865. ISBN 9781450311991. Disponível em: <https://dl.acm.org/doi/10.1145/2228360.2228514>. Acesso em: 15 mai. 2019.

SLIEM, M.; SALMI, N.; IOUALALEN, M. Using performance modelling and analysis for self-adaptive resources allocation systems: A case study. In: **2015 12th International Symposium on Programming and Systems (ISPS)**. Algiers, Argélia: IEEE, 2015. p. 1–10. ISBN 9781479976997. Disponível em: <https://ieeexplore.ieee.org/document/7245006>. Acesso em: 7 mai. 2019.

STEINBAUER, G.; WOTAWA, F. Model-based reasoning for self-adaptive systems-theory and practice. In: CÂMARA, J.; LEMOS, R. de; GHEZZI, C.; LOPES, A. (Ed.). **Assurances for Self-Adaptive Systems: Principles, models, and techniques**.

Berlim, Alemanha: Springer, 2013. p. 187–213. ISBN 9783642362491. Disponível em: [https://link.springer.com/chapter/10.1007/978-3-642-36249-1\\_7](https://link.springer.com/chapter/10.1007/978-3-642-36249-1_7). Acesso em: 12 mai. 2019.

TAMURA, G.; VILLEGAS, N. M.; MÜLLER, H. A.; DUCHIEN, L.; SEINTURIER, L. Improving context-awareness in self-adaptation using the dynamico reference model. **Proceedings of the 8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)**, IEEE, San Francisco, California, p. 153–162, 2013. Disponível em: <https://ieeexplore.ieee.org/document/6595502>. Acesso em: 18 set. 2020.

VILLEGAS, N. M.; MÜLLER, H. A.; TAMURA, G.; DUCHIEN, L.; CASALLAS, R. A framework for evaluating quality-driven self-adaptive software systems. **Proceedings of the 6th international symposium on Software engineering for adaptive and self-managing systems**, ACM, Nova York, USA, p. 80–89, 2011. Disponível em: <https://dl.acm.org/doi/10.1145/1988008.1988020>. Acesso em: 19 mai. 2019.

WEYNS, D. Software engineering of self-adaptive systems. In: CHA, S.; TAYLOR, R. N.; KANG, K. (Ed.). **Handbook of Software Engineering**. Cham, Suíça: Springer, 2019. p. 399–443. ISBN 9783030002626. Disponível em: [https://link.springer.com/chapter/10.1007/978-3-030-00262-6\\_11](https://link.springer.com/chapter/10.1007/978-3-030-00262-6_11). Acesso em: 27 abr. 2019.

WEYNS, D.; CALINESCU, R. Tele assistance: A self-adaptive service-based system exemplar. **2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems**, IEEE, Florença, Itália, p. 88–92, 2015. ISSN 2157-2321. Disponível em: <https://ieeexplore.ieee.org/document/7194661>. Acesso em: 18 set. 2020.

YANG, J.; HUANG, G.; ZHU, W.; CUI, X.; MEI, H. Quality attribute tradeoff through adaptive architectures at runtime. In: **Journal of Systems and Software**. Elsevier, 2009. v. 82, n. 2, p. 319–332. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0164121208001696>. Acesso em: 17 mai. 2019.

ZHU, Y.; HUANG, G.; MEI, H. Quality attribute scenario based architectural modeling for self-adaptation supported by architecture-based reflective middleware. **11th Asia-Pacific Software Engineering Conference**, IEEE, Busan, Coreia do Sul, p. 2–9, 2004. ISSN 1530-1362. Disponível em: <https://ieeexplore.ieee.org/document/1371899>. Acesso em: 21 abr. 2019.