



**UFOP**

Universidade Federal  
de Ouro Preto

**Universidade Federal de Ouro Preto  
Instituto de Ciências Exatas e Aplicadas  
Departamento de Computação e Sistemas**

**Observabilidade em sistemas de  
Internet das Coisas: Um mapeamento  
sistemático**

**Guilherme Jannotti Arantes**

**TRABALHO DE  
CONCLUSÃO DE CURSO**

ORIENTAÇÃO:  
Igor Muzetti Pereira

**Agosto, 2021  
João Monlevade–MG**

**Guilherme Jannotti Arantes**

# **Observabilidade em sistemas de Internet das Coisas: Um mapeamento sistemático**

Orientador: Igor Muzetti Pereira

Monografia apresentada ao curso de Sistemas de Informação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

**Universidade Federal de Ouro Preto**

**João Monlevade**

**Agosto de 2021**

## SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

A662o Arantes, Guilherme Jannotti .  
Observabilidade em sistemas de Internet das Coisas [manuscrito]: um  
mapeamento sistemático. / Guilherme Jannotti Arantes. - 2021.  
39 f.

Orientador: Prof. Me. Igor Muzetti Pereira.  
Monografia (Bacharelado). Universidade Federal de Ouro Preto.  
Instituto de Ciências Exatas e Aplicadas. Graduação em Sistemas de  
Informação .

1. Conectividade (Computadores). 2. Engenharia de software. 3.  
Internet das coisas. I. Pereira, Igor Muzetti. II. Universidade Federal de  
Ouro Preto. III. Título.

CDU 004.41

Bibliotecário(a) Responsável: Flavia Reis - CRB6-2431



MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DE OURO PRETO  
REITORIA  
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS  
DEPARTAMENTO DE COMPUTAÇÃO E SISTEMAS



## FOLHA DE APROVAÇÃO

**Guilherme Jannotti Arantes**

**Observabilidade em Sistemas de Internet das Coisas: Um Mapeamento Sistemático**

Monografia apresentada ao Curso de Sistemas de Informação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação

Aprovada em 23 de Agosto de 2021

Membros da banca

MSc. Igor Muzetti Pereira - Orientador - Universidade Federal de Ouro Preto  
MSc. Euler Horta Marinho - Universidade Federal de Ouro Preto  
Dr. Diego Zuquim Guimarães Garcia - Universidade Federal de Ouro Preto  
MSc. Kattiana Fernandes Constantino

Igor Muzetti Pereira, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 25/08/2021



Documento assinado eletronicamente por **Igor Muzetti Pereira, PROFESSOR DE MAGISTERIO SUPERIOR**, em 25/08/2021, às 10:37, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [http://sei.ufop.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **0212117** e o código CRC **B474E440**.

**Referência:** Caso responda este documento, indicar expressamente o Processo nº 23109.008745/2021-47

SEI nº 0212117

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35400-000  
Telefone: - www.ufop.br

# Agradecimentos

Agradeço primeiramente a minha mãe e ao meu pai pelo incentivo ao estudo, por toda a base, educação e capital cultural. Ao meu irmão que foi a minha fonte de inspiração para o despertar da busca pelo conhecimento, aos meus tios e padrinho, pelos sábios conselhos e por todo suporte, ao meu primo Hugo, que teve papel fundamental na minha vinda à UFOP. Também gostaria de agradecer à todos os meus amigos que conheci nesta jornada, amizades nas quais levarei para a vida toda. Aos meus grandes professores, que parafraseando Isaac Newton, estão sentados sobre ombros de gigantes e me permitiram ver mais longe. Um agradecimento a todas as pessoas e todas as conexões que tive neste processo que me tornaram uma pessoa melhor. Também não poderia esquecer de agradecer ao sistema público e gratuito de ensino, que me possibilitou o acesso à educação de qualidade, em especial ao colégio Modesta Cravo, Carlos Lacerda e Estadual Central e as universidades federais UFMS e UFOP.

*“Um indivíduo tem a responsabilidade moral de desobedecer leis injustas.”*

— Martin Luther King Jr.

# Resumo

Nossa sociedade tende a se tornar cada vez mais conectada graças à conectividade de diferentes dispositivos entre si, conosco e com outras redes de comunicação. Como resultado, há uma demanda crescente para conectar esses objetos de vários tipos, finalidades, tecnologias e fabricantes. Essa demanda cria dilemas e desafios para garantir conectividade, segurança, disponibilidade, entre outros requisitos não funcionais para esses sistemas. Para que essas demandas sejam atendidas, é fundamental testar, monitorar e garantir o correto funcionamento desses sistemas. A crescente complexidade que envolve este cenário exige uma evolução no campo do monitoramento desses sistemas IoT. É aí que entra a capacidade de observação para ajudar a entender os fluxos, descobrir por que um sistema se comporta de determinada maneira e identificar problemas rapidamente. Realizamos um mapeamento sistemático da literatura para identificar o estado da prática na pesquisa de observabilidade em engenharia de software, em especial para sistemas IoT. Como principais contribuições, listamos alguns desafios encontrados na implementação da observabilidade em estudos científicos, e identificamos a necessidade de pesquisas científicas que abordem o conceito de observabilidade para sistemas IoT.

**Palavras-chaves:** Observabilidade. Internet das Coisas. Sistemas IoT.

# Abstract

Our society tends to become more and more connected thanks to the connectivity of different devices among themselves, with us, and with other communication networks. As a result, there is a growing demand for connecting these objects of various types, purposes, technologies, and manufacturers. This demand creates dilemmas and challenges to ensure connectivity, security, availability, among other non-functional requirements for these systems. For these demands to be met, it is essential to test, monitor, and guarantee the correct functioning of these systems. The growing complexity that involves this scenario requires an evolution in the field of monitoring these IoT systems. That's where observability comes in to help understand flows, discover why a system behaves a certain way, and identify problems quickly. We carried out systematic mapping of the literature to identify the state of practice in observability research in software engineering, especially for IoT systems. As main contributions, we list some challenges found in the implementation of observability in scientific studies, and we identify the need for scientific research that addresses the concept of observability for IoT systems.

**Keywords:** Observability. Internet of Things. IoT Systems.

# Lista de ilustrações

Figura 1 – Adaptada de Atzori . . . . .	13
Figura 2 – Ilustração de sistemas que permitem realizar observabilidade . . . . .	15
Figura 3 – Número de artigos filtrados por etapa . . . . .	23
Figura 4 – Distribuição dos estudos revisados por ano . . . . .	25

# Lista de tabelas

Tabela 1 – Critérios de Inclusão e Exclusão . . . . .	22
Tabela 2 – Compilação do resultado da busca utilizando as duas SB . . . . .	23
Tabela 3 – Artigos selecionados . . . . .	24
Tabela 4 – Area de domínio dos artigos. . . . .	26
Tabela 5 – Categoria de estudo dos artigos. . . . .	26
Tabela 6 – Desafios reportados. . . . .	29

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>10</b>
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>12</b>
<b>2.1</b>	<b>Fundamentação Teórica</b>	<b>12</b>
2.1.1	Internet das Coisas	12
2.1.2	Monitoramento	13
2.1.3	Observabilidade	14
2.1.4	Registros, Métricas e Rastreamento	15
<b>2.2</b>	<b>Trabalhos Correlatos</b>	<b>18</b>
<b>3</b>	<b>METODOLOGIA</b>	<b>21</b>
<b>3.1</b>	<b>Mapeamento Sistemático da Literatura (MSL)</b>	<b>21</b>
3.1.1	Planejamento	21
3.1.2	Execução	22
<b>4</b>	<b>RESULTADOS</b>	<b>25</b>
<b>4.1</b>	<b>Visão geral dos estudos primários</b>	<b>25</b>
<b>4.2</b>	<b>Análise dos dados coletados</b>	<b>27</b>
<b>5</b>	<b>AMEAÇAS A VALIDADE</b>	<b>32</b>
<b>6</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>33</b>
	<b>REFERÊNCIAS</b>	<b>34</b>

# 1 Introdução

A Internet das Coisas está cada vez mais difundida na sociedade, essas tecnologias permeiam diversas áreas que incluem automação residencial, cidades inteligentes, indústria aeroespacial<sup>1</sup> entre outros. Com o advento e difusão desses sistemas, a capacidade de monitoramento tradicionalmente utilizada em sistemas monolíticos se mostram insuficientes se aplicadas a essas arquiteturas distribuídas (KOETTER, 2019), dado a sua complexidade e dinâmica (LIMA, 2019).

A natureza centralizada dos sistemas monolíticos tornam a tarefa de monitoramento mais simples se comparado a sistemas IoT, criando uma barreira significativa para adoção dessas tecnologias de monitoramento (KNOCHE, 2019). Devido sua natureza distribuída, os sistemas IoT possuem muitos pontos de falha e são gerados um grande número de registros pelos múltiplos serviços e seus respectivos dispositivos interconectados, tornando o armazenamento e gerenciamento desses registros penosos (DEURSEN, 2021).

A escolha de como e o que registrar é muito importante e gera um dilema, pois o registro excessivo pode causar degradação do desempenho devido ao número de operações de gravação e pode ser caro em termos de armazenamento. Por outro lado, informações insuficientes prejudicam a utilidade dos dados para a equipe de operações (DEURSEN, 2021).

Essa lacuna entre a crescente complexidade dos sistemas e a capacidade das ferramentas de monitoramento tradicionais em gerenciar essa complexidade ainda é objeto de estudo na comunidade acadêmica, em especial em pesquisas utilizando um novo conceito chamado de observabilidade, que apesar de se confundir com o conceito monitoramento, possuem diferentes características, finalidades e objetivos.

Observabilidade é uma prática na qual utilizam-se ferramentas para instrumentalizar sistemas para coletar dados viáveis que proporcionem uma visão global sobre o estado do sistema, objetivando a descoberta de erros para entender os motivos que levaram a estes problemas, visando ganho de disponibilidade, estabilidade, integridade e outros conceitos fundamentais nos dias de hoje, que refletem diretamente em prejuízos financeiros dado a ampla utilização desses sistemas nos dias de hoje<sup>2</sup> (KINSELLA, 2015).

O objetivo deste estudo é verificar os desafios da observabilidade com a proposta de realizar um mapeamento sistemático da literatura com o intuito de classificar as evidências do uso da observabilidade em sistemas IoT do ponto de vista de praticantes e pesquisadores em engenharia de software no contexto da literatura científica disponível.

---

<sup>1</sup> <https://www.airbus.com/newsroom/stories/iot-aerospace-great-new-connector.html>

<sup>2</sup> <https://www.comparitech.com/internet-providers/iot-statistics/>

As principais contribuições deste trabalho incluem, (i) a identificação da necessidade de pesquisas científicas que abordem o conceito de observabilidade para sistemas IoT e (ii) discute os desafios encontrados para permitir a observabilidade de sistemas IoT.

O restante deste trabalho é organizado como segue. O Capítulo 2 apresenta a revisão bibliográfica, composta pela fundamentação teórica e trabalho correlatos. O Capítulo 3 apresenta a metodologia e mapeamento sistemática da literatura. O Capítulo 4 aborda os resultados, com a visão geral dos estudos primários e a resposta da questão de pesquisa. O Capítulo 5 apresenta as ameaças a validade e o Capítulo 6 apresenta as considerações finais.

## 2 Revisão Bibliográfica

### 2.1 Fundamentação Teórica

Esta sessão tem como finalidade abordar e contextualizar alguns assuntos discutidos no trabalho para maior entendimento do leitor, visando explicar estes conceitos teóricos.

#### 2.1.1 Internet das Coisas

As origens do termo Internet das Coisas datam de mais de vinte anos e foram atribuídas ao trabalho do Auto-ID Labs no Massachusetts Institute of Technology (MIT) em infraestruturas de identificação por radiofrequência em rede ([ATZORI, 2010](#)).

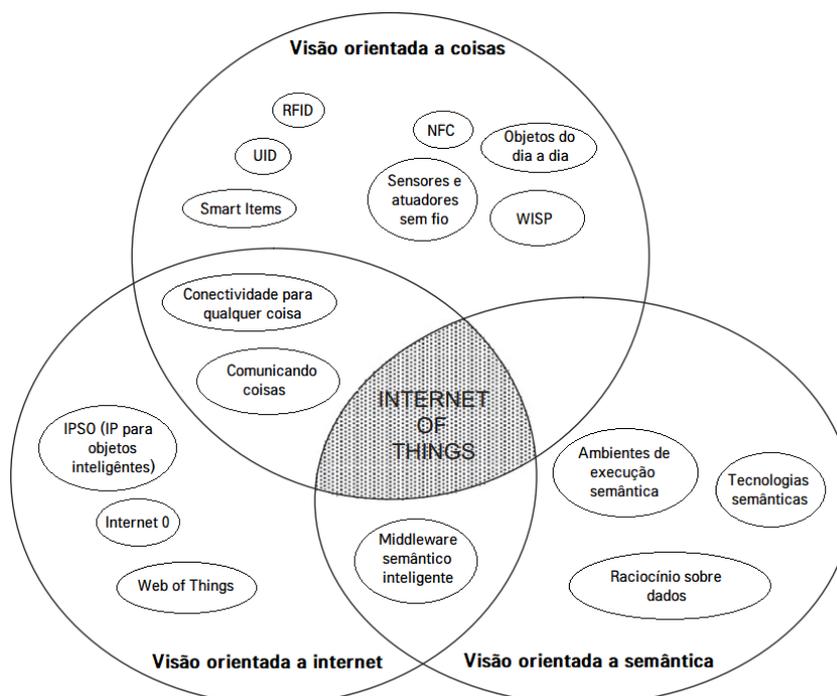
O conceito de Internet das coisas não é único e não há unanimidade em relação a sua definição na comunidade científica, havendo fortes divergências em relação ao seu conceito. O motivo da aparente imprecisão em torno do termo é uma consequência do próprio nome Internet das Coisas, que sintaticamente é composto de dois termos. O primeiro empurra para uma visão orientada para a rede IoT, enquanto o segundo move o foco em objetos genéricos para serem integrados em uma estrutura comum ([ATZORI, 2010](#)).

De maneira geral, IoT pode ser entendido como um conjunto de objetos e dispositivos que compõe um ambiente de computação onipresente e interconectado com a internet por meio de sensores e atuadores, que são utilizados para facilitar a vida das pessoas, introduzindo soluções para diversos processos do dia a dia ([MAGRANI, 2018](#)).

O que essas definições de IoT tem em comum é que elas se concentram em como computadores, sensores e objetos que interagem uns com os outros e processam informações em um contexto de hiper-conectividade. O que hoje é chamado de Internet das Coisas é um conjunto de tecnologias e protocolos associados que permitem que objetos se conectem a uma rede de comunicações e são identificados e controlados através desta conexão de rede.

Esse ambiente composto por um grande número de objetos (heterogêneos) envolvidos no processo trazem algumas consequências, o endereçamento exclusivo do objeto e a representação e armazenamento das informações trocadas tornam-se as questões mais desafiadoras, trazendo diretamente para uma outra perspectiva de IoT, orientada para a semântica. Na Figura 1, os principais conceitos, tecnologias e padrões são destacados e classificados com referência à(s) visão(ões) de IoT que melhor contribuem para sua caracterização, e apresenta o paradigma da IoT que resulta da convergência das três visões: coisas, internet e semântica ([ATZORI, 2010](#)).

Figura 1 – Adaptada de Atzori



Os campos de aplicação para tecnologias IoT são muito numerosas por serem diversificados, pois as soluções de IoT estão se estendendo cada vez mais a praticamente todas as áreas do nosso dia a dia (FLUCHTER, 2015).

### 2.1.2 Monitoramento

O monitoramento é uma solução técnica que permite aos profissionais de computação compreenderem o estado do sistema e sua utilização é fundamental para a manutenção e garantia de qualidade de software. A definição utilizada pelo Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE), diz que o monitoramento é definido como a supervisão, registro, análise ou verificação da operação de um sistema ou componente. (IEEE, 1990)

No contexto da computação, um dos objetivos do monitoramento é de observar e rastrear métricas de desempenho e integridade de sistemas e aplicativos, com a finalidade de detectar e diagnosticar comportamento indesejado e melhorar a confiabilidade do sistema (CANDIDO, 2019).

Uma pesquisa realizada pelo Google chamada DevOps Research and Assessment (DORA <sup>1</sup>), define o monitoramento como uma ferramenta ou solução técnica que permite às equipes monitorar e entender o estado dos sistemas, baseado na coleta de conjuntos predefinidos de métricas e registros.

<sup>1</sup> <https://cloud.google.com/architecture/devops/devops-measurement-monitoring-and-observability?hl=pt-br>

O monitoramento dá instrumentação às equipes de software, que coletam dados sobre seus sistemas, permitindo uma ação mais veloz para mitigar erros e problemas. Ou seja, é a preparação dos sistemas para coletarem dados, com o objetivo de saber quando algo dá errado, possibilitando uma resposta ágil (NEWRELIC, 2021).

Um exemplo da importância do monitoramento está no fato de como é desafiador entender, solucionar problemas e otimizar o comportamento de sistemas em implantações de grandes escalas, existem sistemas que fornecem ferramentas de monitoramento que observam o uso de recursos e as características de desempenho dos componentes individuais. Essas ferramentas normalmente empregam técnicas de monitoramento não intrusivas que controlam o uso de CPU, memória, disco, rede e as interações com o monitor de máquina virtual (hipervisor) (SUNEJA, 2014).

### 2.1.3 Observabilidade

O termo Observabilidade vem da engenharia elétrica, mais especificamente de sistemas dinâmicos lineares da teoria de sistemas de controle. Neste contexto, a definição de observabilidade diz que um sistema é dito ser observável no instante  $t^0$  se, com o sistema num estado  $x(t^0)$  qualquer, for possível determinar este estado a partir da observação da saída durante um intervalo de tempo finito (GOPAL, 1984). Em palavras menos formais, é uma medida que descreve quão bem podem os estados de um sistema serem inferidos a partir do conhecimento de suas saídas externas.

A observabilidade indica até que ponto a infraestrutura e os aplicativos e suas interações podem ser monitorados. As saídas utilizadas são, por exemplo, registros, métricas e rastreamentos (PICORETI et al., 2018). Assim como o monitoramento (porém de maneira mais ampla) também tem como objetivo garantir que os operadores responsáveis pelo sistema possam detectar comportamentos indesejáveis tendo as informações necessárias para determinar a causa raiz de maneira eficaz. Observabilidade significa reunir as informações (advindas de logs, métricas e rastreamento) de uma forma coordenada para encontrar problemas e gargalos.

Enquanto o monitoramento fornece informações como se um sistema está operando conforme o esperado, a observabilidade abrange a pesquisa de por que um aplicativo se comporta de determinada maneira. Semelhante à escalabilidade e resiliência, é uma propriedade do sistema. A capacidade de observação permite que os engenheiros explorem sistemas e respondam a perguntas ainda a serem formuladas quando o código foi implantado.

Na computação, o termo observabilidade ganha espaço na medida que se dissemina os sistemas baseados em arquiteturas de microserviços, superando as limitações de arquiteturas monolíticas. Entretanto, estas arquiteturas modernas estão cada vez mais complexas

(ACETO, 2013), flexíveis e modulares, caracterizadas pelo uso de serviços independentes, como microserviços e web services (HONEYCOMB, 2018).

Embora os sistemas que consistem em serviços da Web forneçam melhor observabilidade do que os sistemas monolíticos, os serviços têm o potencial de aprimorar a observabilidade e monitoramento, fornecendo informações relevantes sobre o comportamento interno (SUN, 2018). A base que da sustentação à observabilidade consiste em três pilares: registros, métricas e rastreamento (INDRASIRI, 2018).

### Monitoramento vs Observabilidade

Apesar de seus conceitos se misturarem em alguns quesitos dando a impressão de ser apenas um nome mais moderno para o monitoramento, na realidade os objetivos de monitoramento e observabilidade são diferentes, desta maneira, um não é substituto do outro e nem elimina a sua necessidade, eles são complementares. Apesar de ser um termo mais novo, a observabilidade não é uma ideia recente. Se a observabilidade é a capacidade de inferir os estados internos de um sistema, o monitoramento pode ser definido como as ações envolvidas no ato de praticar observabilidade. Sendo um pouco filosófico, enquanto o monitoramento permite responder questões e questionamentos previamente estabelecidos, a observabilidade permite responder novas questões que os profissionais da área talvez não saibam de antemão.

Figura 2 – Ilustração de sistemas que permitem realizar observabilidade



#### 2.1.4 Registros, Métricas e Rastreamento

Os três pilares da observabilidade são compostos por Registros (Logs), Métricas e Rastreamento. Cada um desempenha uma função específica no monitoramento do sistema e infraestrutura, a utilização em conjunto desses conceitos são fundamentais para que administradores de sistemas possam ter uma visão geral e holística do ambiente, auxiliando sobremaneira na sua manutenção e entendimento.

**Registros (Logs).** Log de dados é o termo utilizado para descrever o processo de registro de eventos num sistema computacional, esses registros são geralmente na forma de texto livre. Os logs podem expor partes do estado do sistema, como exceções e valores de variáveis e é frequentemente usado em diagnósticos de falhas e auditoria. Em uma mensagem de log, os desenvolvedores descrevem o evento registrado usando texto estático e, opcionalmente, registram os valores das variáveis relacionadas ao evento. (YUAN; PARK; ZHOU, 2012).

Segundo o guia foca linux avançado<sup>1</sup> um arquivo de log geralmente é composto dos seguintes campos:

*Data|Hora|Máquina|Daemon|Mensagem.*

O campo máquina é o nome do computador que registrou a mensagem. O campo daemon indica qual programa gravou a mensagem. A utilização da linguagem de programação **awk** pode ser muito útil na pesquisa e monitoramento dos logs.

Já o daemon de log é um programa que registra as saída de logs do kernel e do sistema. No Linux os daemon mais conhecidos são o syslogd, systemd e systemctl. A classificação de qual arquivo em */var/log* receberá qual tipo de mensagem é controlado pelo arquivo de configuração */etc/rsyslog.conf*. As mensagens de log possuem um nível de detalhamento e especificam a importância da mensagem. No arquivo de configuração *rsyslog.conf* são permitidos os seguintes níveis:

**emerg:** O sistema está inutilizável.

**alert:** Uma ação deve ser tomada imediatamente para resolver o problema.

**crit:** Condições críticas.

**err:** Condições de erro.

**warning:** Condições de alerta.

**notice:** Condição normal, mas significativa.

**info:** Mensagens informativas.

**debug:** Mensagens de depuração.

A concepção dos logs em um aplicativo envolvem várias decisões importantes que tem impacto direto na eficácia da análise futura. O log excessivo pode ser um problema dado a degradação de desempenho devido ao número de operações de gravação e pode ser caro em termos de armazenamento. Por outro lado, informações insuficientes prejudicam a utilidade dos dados para a equipe de operações (DEURSEN, 2021).

É importante mencionar que ambientes subjacentes também fornecem dados valiosos, que fornecem conhecimento sobre o uso de recursos (memória, CPU e rede) e esses

<sup>1</sup> <https://www.guiafoca.org/guiaonline/avancado>

dados podem ser correlacionados com os registros do aplicativo no processo de análise. Em contraste dos logs do aplicativo, os desenvolvedores não estão no controle dos logs do ambiente. Por outro lado, muitas vezes são altamente estruturados e são úteis como uma fonte de dados complementares que fornecem contextos adicionais (DEURSEN, 2021).

Os logs podem ser entendidos como provas digitais, dado sua utilização para auditorias e diagnóstico de problemas em sistemas computacionais. Os logs possuem grande importância para o Direito da Tecnologia da Informação, pois possibilita identificar a autoria de ações no ambiente virtual, permitindo a responsabilização dos autores.

**Métricas.** De acordo com o Padrão de Metodologia de Métricas de Qualidade de Software do IEEE, a métrica de software é uma função cuja entrada são os dados do software e a saída é um valor que pode decidir como um determinado atributo afeta o software (IEEE, 1990).

As métricas de software fornecem uma medida para o software e para o processo de produção de software e dão valores quantitativos aos atributos que envolvem o produto ou o processo de desenvolvimento de software. Esses atributos são extraídos do produto de software, do processo de desenvolvimento de software e dos recursos relacionados. A medição de software define, coleta e analisa os dados de processo mensuráveis, por meio dos quais facilitam o entendimento, avaliação, controle e aprimoramento do procedimento de produção de software (YANAN, 2009).

As métricas podem ser usadas para determinar o comportamento do sistema ou componente ao longo do tempo e compreendem um conjunto de atributos (por exemplo, nome, valor, rótulo) que transmitem informações. Com as diversas métricas é possível correlacionar os componentes da infraestrutura para obter uma visão holística da integridade e do desempenho do sistema.

Existem diversos tipos de métricas, como por exemplo sobre o tempo de atividade do sistema, o tempo de resposta, o número de solicitações por segundo, entre outras. As métricas fornecem visibilidade e percepções mais profundas que ajudam a entender a causa dos problemas. Além de fornecer visibilidade, pode-se usar as informações para detectar e determinar a gravidade dos problemas.

**Rastreamento.** Fundamentalmente a rastreabilidade é o potencial de relacionar dados armazenados em artefatos de algum tipo, junto com a capacidade de examinar esse relacionamento. A capacidade de obter rastreabilidade, portanto, depende da criação de links navegáveis entre os dados mantidos nos artefatos que, de outra forma, estão desconectados (ZISMAN, 2012). Em outras palavras, rastreabilidade é a habilidade de inter-relacionar artefatos de engenharia de software, mantendo os links necessários ao longo do tempo, e tais informações podem responder a perguntas sobre o produto de software e seu processo de desenvolvimento (GOTEL, 2014).

Rastreabilidade refere-se à criação de links entre diferentes tipos de artefatos de software. Existem duas maneiras principais de estabelecer a rastreabilidade: *recuperação de rastreamento*, que se refere à rastreabilidade após o fato e é definida como a abordagem para criar links de rastreamento após os artefatos que eles associam terem sido gerados ou manipulados e *captura de rastreamento*, que se refere à rastreabilidade em tempo real e é definida como a criação de links de rastreamento simultaneamente com a criação dos artefatos que eles associam (KAROUNTZOS, 2020).

O artefato de rastreamento representa qualquer dado passível de rastreamento, que pode variar em termos de granularidade (por exemplo, módulo de arquitetura, fragmento de código-fonte, etc.), e os links de rastreamento podem ser usados para analisar como as mudanças aplicadas aos artefatos existentes podem afetar outros artefatos no mesmo nível ou em níveis diferentes de abstração (VALE et al., 2017).

Um único rastreamento pode fornecer visibilidade tanto do caminho percorrido por uma solicitação quanto da estrutura de uma solicitação. O caminho de uma solicitação permite que engenheiros de software entendam os diferentes serviços envolvidos no caminho de uma solicitação, e a estrutura de uma solicitação ajuda a entender as conjunturas e os efeitos da assincronia na execução de uma solicitação.

Os benefícios da rastreabilidade incluem melhor compreensão do programa, atividades de manutenção mais fáceis, capacidade de avaliar a integridade do produto com base nos requisitos rastreados, análise de impacto e uma boa base para reutilização de software.

## 2.2 Trabalhos Correlatos

Para fornecer fundamentos para a pesquisa descrita neste artigo, os trabalhos relacionados investigam as práticas de pesquisa atuais sobre observabilidade, bem como pesquisas anteriores sobre monitoramento e observabilidade em sistemas IoT. Nosso trabalho até o momento ajuda a discussão sobre o uso da observabilidade em projetos de sistemas IoT. Entretanto, outros estudos que permeiam a área de observabilidade para sistemas IoT foram encontrados e analisados neste mapeamento.

A Revisão Sistemática da Literatura (RSL) de Bertolino et al. aborda testes em aplicações baseadas em nuvens e levanta, entre outras, questões como quais os principais objetivos dos testes dessas aplicações e quais são as técnicas, métodos e ferramentas utilizadas. Alguns desafios descritos por seu trabalho incluem alto custo, falta de padrões, complexidade, representação do ambiente real, elasticidade e segurança. Os tópicos de domínio mais citados incluem IoT, mas predomina Aplicações Web, Mobile e Service Oriented Architectures (SOA). O autor cita com entusiasmo e como sendo promissor o projeto ElasTest <sup>2</sup>, uma plataforma elástica que traz observabilidade para testar sistemas

---

<sup>2</sup> <https://elastest.io/>

de software distribuídos grandes e complexos ([BERTOLINO et al., 2019](#)).

Uma RSL, conduzida por Koetter et al., relata os desafios, requisitos e boas práticas no campo da observabilidade e monitoramento de sistemas distribuídos. É realizada uma análise dos desafios contemporâneos de monitoramento com o propósito de deduzir requisitos e mapear soluções e estratégias existentes em sistemas de monitoramento. Através de sua entrevista semiestruturada, observou uma complexidade crescente nesse campo e da forte necessidade de um bom planejamento organizacional, incluindo estratégia, funções e responsabilidades bem definidas. O estudo identificou a necessidade de mais trabalhos sobre boas práticas e exemplos do mundo real para alinhar objetivos de negócios com métricas e técnicas para quebrar barreiras e permitir desenvolvimento e operação eficientes ([KOETTER, 2019](#)).

Em sua RSL, Nguyem et al. estudam os avanços nas abordagens de orquestração e implantação para IoT. Seu trabalho elenca algumas lacunas encontradas na área, como por exemplo a dificuldade de se aproximar da indústria, devido a dominância acadêmica das pesquisas. Também levanta a necessidade de avanços nas abordagens de orquestração e implantação IoT para dar suporte ao processamento distribuído e comportamento coordenado em IoT, infraestruturas de borda e nuvem, para lidar com a vasta heterogeneidade, dinâmica e escalabilidade necessárias ([NGUYEN et al., 2019](#)).

O Mapeamento Sistemático da Literatura (MSL) de Avgeriou et al. trata da rastreabilidade de software, analisando o estado da arte e fazendo comparações entre diferentes abordagens. O autor não entra no mérito específico da observabilidade e de IoT, mas como a rastreabilidade é considerada um dos pilares da observabilidade, é um artigo que agrega valor para este mapeamento. O autor cita alguns benefícios do uso de abordagens de rastreabilidade, como na melhoria da compreensão, reutilização, correção e manutenção de software ([AVGERIOU, 2020](#)).

O MSL de Deursen et al., sobre monitoramento de software baseado em logs, elenca abordagens para resolver os desafios do ciclo de vida dos logs, infraestrutura de logs, análises de logs, entre outros. Os autores citaram como impeditivo para inovação a falta de dados representativos de logs de aplicações, dado a falta de acesso por parte de desenvolvedores, sugerindo uma aproximação da academia com a indústria visando fechar esta lacuna e propiciar a geração futura de ferramentas e padrões para logs. A análise dos autores identifica que, mesmo necessitando de uma investigação mais aprofundada, o aprendizado de máquina é uma abordagem promissora para permitir uma análise contextual do código-fonte para recomendação de logs. Eles também comentam que existem poucos estudos abordando persistência eficiente de dados de registro ([DEURSEN, 2021](#)).

Houve um esforço considerável para encontrar trabalhos correlatos que tratam especificamente da observabilidade para sistemas IoT. O artigo que chegou mais próximo disso foi o do Koetter et al., que aborda a observabilidade e monitoramento para sistemas distribuídos, no qual também foi possível observar uma crescente complexidade na área, além da necessidade de mais trabalhos sobre o tema.

Os MSL de Deursen e de Avgeriou, tratam de temas correlatos, temas que são pilares da observabilidade, mais especificamente o estudo de Deursen que fala de monitoramento de software baseado em logs e o MSL de Avgeriou que discorre sobre rastreabilidade de software, ambos citam benefícios e desafios encontrados nas pesquisas.

Já o estudo de Bertolino que apresenta uma RSL sobre testes em aplicações baseadas em nuvem e a RSL de Nguyem que estuda os avanços nas abordagens de orquestração e implantação para IoT, são artigos que possuem algumas correlações com este estudo, dado que abordam temas que circundam a observabilidade e a internet das coisas, mas não são profundamente relacionados, uma vez que o objetivo principal não discorre especificamente sobre observabilidade em sistemas IoT.

## 3 Metodologia

### 3.1 Mapeamento Sistemático da Literatura (MSL)

Um MSL é uma investigação científica que engloba estudos expressivos sobre um tópico de pesquisa específico. Os MSL identificam um conjunto de estudos já finalizados que abordam uma determinada questão de pesquisa e avaliam os resultados desses estudos para evidenciar conclusões sobre um corpo de conhecimento (CUMMINGS R.; BROWNER, 2015). Este trabalho é conduzido usando as diretrizes de Kitchenham (KITCHENHAM, 2007). Possui três etapas: (i) planejamento, (ii) execução e (iii) análise. O foco deste MSL é para analisar o estado atual da prática de pesquisa sobre observabilidade em engenharia de software, em especial para sistemas IoT. A seção 3.1.1 apresenta a fase de planejamento. A seção 3.1.2 descreve a execução e as etapas para a seleção dos estudos primários e secundários. A etapa de análise e relatos é apresentada no capítulo 4.

#### 3.1.1 Planejamento

Nessa fase, foi definido: (i) o tema a ser investigado; (ii) as bases de dados eletrônicas utilizadas para a busca de artigos; (iii) a string de pesquisa para identificar estudos relevantes; (iv) os critérios de inclusão e exclusão para obter estudos primários; (v) e data em que este trabalho foi realizado.

**Questão de Pesquisa (QP).** *Qual é o estado da prática da pesquisa sobre observabilidade em engenharia de software, em especial para sistemas IoT?*

Para identificar o atual estado da prática da pesquisa sobre observabilidade em sistemas IoT na comunidade de engenharia de software, considerando os desafios, ferramentas e estratégias para garantir a observabilidade em sistemas para IoT, definimos essa questão de pesquisa geral. Essa questão de pesquisa permite atingir o objetivo e orienta todas as atividades que conduzimos dentro deste mapeamento.

**Banco de dados eletrônicos (BDE).** Existem diferentes BDE que podem ser usados em revisões de literatura para a busca de estudos primários. Utilizamos quatro deles: IEEE Xplore Digital Library, ACM Digital Library, ScienceDirect e Springer, conforme mostra a Tabela 2. Estes BDE pertencem à categoria de sites de editores conhecidos como parte do grupo de fontes importantes para fácil recuperação de literatura relevante e publicada sobre engenharia de software (CHEN; BABAR; ZHANG, 2010). A primeira coluna da tabela mostra os nomes dos bancos de dados eletrônicos, a segunda apresenta seus respectivos sites e a terceira a quantidade de artigos retornados através da pesquisa realizada com as strings de busca.

**String de busca (SB).** A SB identifica os estudos relevantes nos bancos de dados eletrônicos selecionados, permitindo responder a questão de pesquisa proposta. Foram realizados testes com diferentes combinações de palavras chave para a composição da SB. Também foi avaliado os resultados em cada banco de dados eletrônicos com o objetivo de identificar quais das SB alcançaram o maior número possível de estudos na literatura neste contexto, após vários testes ficou definido a utilização de duas SB:

- 1) *((observability) AND ("internet of things"OR iot) AND ("software engineering"))*
- 2) *((logs) AND (traces) AND (metrics) AND (iot OR "internet of things"))*

**Critérios de inclusão e exclusão.** Temos quatro critérios de inclusão e dois critérios de exclusão para selecionar os estudos primários (ver Tabela 1). Esses critérios permitem classificar cada estudo em análise como candidato a ser incluído ou excluído do mapeamento.

Tabela 1 – Critérios de Inclusão e Exclusão

<b>Critérios de Inclusão</b>	<b>Critérios de Exclusão</b>
Escrito em Inglês.	Artigos duplicados.
Publicado em conferências, periódicos e workshops.	Tese, dissertação, tutorial e curso.
Disponível no formato eletrônico.	Artigos com menos de 3 páginas.
Publicações de engenharia de software	

**Período da busca.** Pesquisamos todos os estudos publicados de 2016 até 2021, apenas artigos encontrados através da técnica de Snowballing foram considerados com publicação que precede 2016. O processo de pesquisa foi realizado de 22 de Fevereiro a 10 de Junho de 2021.

### 3.1.2 Execução

Esta fase consiste em: (i) aplicar a string de busca nas bases de dados previamente selecionadas, (ii) identificar os estudos e (iii) selecionar os estudos relevantes encontrados por meio dos critérios de inclusão e exclusão. A Tabela 2 a seguir apresenta os resultados do passo (ii).

Tabela 2 – Compilação do resultado da busca utilizando as duas SB

Banco de dados eletrônicos	Endereço	Artigos Retornados
ACM Digital Library	https://dl.acm.org	499
IEEE Xplore Digital Library	https://ieeexplore.ieee.org	8
ScienceDirect	https://sciencedirect.com	524
Springer	https://link.springer.com	185
Total		1223

A Figura 3 apresenta o fluxo do processo de filtragem de artigos, o resultado de um passo serve como entrada para a próxima etapa. Os símbolos de verificação indicam os estudos que permaneceram na etapa, enquanto os números próximos aos símbolos vermelho indicam os estudos excluídos na etapa. Ao final do processo, identificamos treze artigos que se enquadram no escopo deste trabalho. Esses estudos foram analisados e resumidos a fim de coletar informações para responder a nossa QP.

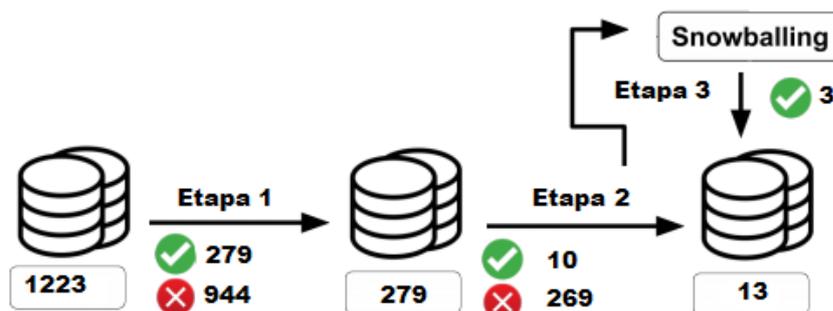


Figura 3 – Número de artigos filtrados por etapa

**Processo de seleção de estudos.** A Figura 3 apresenta as três etapas com foco na seleção de artigos relevantes de acordo com seu conteúdo. As três etapas são discutidas a seguir.

**Etapa 1** *Crítérios de inclusão e exclusão.* Após a aplicação dos critérios de inclusão e exclusão definidos na Seção 3.1.1, se o artigo se enquadrar em pelo menos um dos critérios de exclusão, ele é removido de nosso MSL.

**Etapa 2** *Leitura do título, abstract e palavras-chave.* Aqui, selecionamos os artigos que mostram alguma evidência de estarem vinculados ao contexto do estudo proposto neste artigo, por exemplo conceitos relacionados com observabilidade, tais como métricas, registros, rastreamento e monitoramento.

**Etapa 3** *Snowballing.* Nesta etapa executamos Backward Snowballing (WOHLIN, 2014). Esta técnica é executada usando a lista de referências bibliográfica para identificar novos artigos a serem incluídos. Basicamente, essa técnica consiste em fazer um loop que começa percorrendo a lista de referências, excluindo artigos que não atendam aos critérios

de inclusão e que já foram examinados anteriormente; os artigos restantes são os possíveis candidatos à inclusão após análise de todas as informações disponíveis. Ao acrescentar um novo artigo, verificou-se a lista de referências do mesmo e repetiu-se todo processo até que não fosse possível encontrar mais artigos relevantes.

**Extração de dados.** A questão de pesquisa é o principal direcionador dos quais informações precisam ser extraídas. Assim, o tema principal da questão de pesquisa foi identificado e resumido em uma tabela. Para cada artigo encontrado, foi documentado: (i) os desafios existentes no que diz respeito a observabilidade; e (ii) as tecnologias usadas para aplicação da observabilidade e reportadas nos estudos. Essas informações nos permitem identificar os maiores obstáculos e discutir as tecnologias usadas.

Tabela 3 – Artigos selecionados

<b>Estudo</b>	<b>Título</b>
[e1]	DCO analyzer: local controllability and observability analysis and enforcement of distributed test scenarios.
[e2]	Real-time Feedback in Node-RED for IoT Development: An Empirical Study.
[e3]	A model-based architecture for interactive run-time monitoring.
[e4]	On the Observability and Controllability of Large-Scale IoT Networks: Reducing Number of unmatched Nodes via Link Addition.
[e5]	Cyber-Physical Observability for the Electric Grid.
[e6]	A Testbed for Hardware-assisted Online Profiling of IoT devices.
[e7]	Automated Scenario-Based Integration Testing of Time-Constrained Distributed Systems.
[e8]	Securing Industrial Control Systems.
[e9]	Testing and deployment patterns for the internet-of-things.
[e10]	On Observability and Monitoring of Distributed Systems - An Industry Interview Study
[e11]	A Microservice Architecture for the Industrial IoT.
[e12]	On Performance Observability in IoT Systems using Active Measurements.
[e13]	Data-driven monitoring for cloud compute systems.

## 4 Resultados

Esta sessão responde a questão de pesquisa apresentada na Seção 3.1.1. A seção 4.1 descreve a visão geral e a caracterização dos artigos encontrados. As seção 4.2 apresentam as análises dos dados coletados para responder a questão de pesquisa.

### 4.1 Visão geral dos estudos primários

**Demografia dos estudos.** Os dados foram extraídos dos artigos e sintetizados para responder à questão de pesquisa. Esta seção começa com os metadados descritivos dos estudos e, em seguida, relata as descobertas importantes com base na síntese de dados em resposta à questão de pesquisa em vários formatos de representação.



Figura 4 – Distribuição dos estudos revisados por ano

Após o processo de busca e seleção do estudo, identificamos treze estudos relevantes para revisão. Dentre eles, oito foram publicados em sete conferências internacionais diferentes, três foram publicados em workshops e outros dois estudos foram publicados em periódicos. A Figura 4 mostra o número de estudos revisados publicados por ano. O número de estudos relevantes publicados nos últimos anos só vem aumentando e demonstram uma tendência de crescimento de pesquisas sobre a observabilidade em sistemas IoT no contexto de engenharia de software. Em relação as areas de domínio dos estudos, foi realizado uma categorização através de uma leitura completa de cada artigo visando entendê-lo e

definindo um tema principal para cada artigo. Como resultado foi obtido cinco principais áreas de domínio dos estudos, apresentadas na tabela 4.

Tabela 4 – Area de domínio dos artigos.

Área de domínio:	Artigos:
Testes	[e1], [e4], [e5], [e12], [e13]
Redes	[e9], [e10], [e11]
Arquiteturas	[e6], [e8]
Monitoramento	[e2], [e3]
Segurança	[e7]

Para estabelecer o conjunto de estratégias utilizadas na pesquisa, classificamos os estudos selecionados de acordo com as categorias de estudo de (WIERINGA et al., 2005) e sintetizamos na tabela 5. A classificação foi realizada pelo autor. A categoria com mais estudos (seis) foram propostas de soluções. Seguido de pesquisa de validação e pesquisa de avaliação com três cada. Finalmente, encontramos um artigo filosófico.

Tabela 5 – Categoria de estudo dos artigos.

Categoria de Estudo:	Artigos:	Total:
Proposta de solução:	[e1], [e4], [e5], [e6], [e8], [e13]	6
Pesquisa de avaliação:	[e3], [e10], [e12]	3
Pesquisa de validação:	[e2], [e7], [e9]	3
Artigo filosófico:	[e11]	1

A proposta de solução propõe uma solução técnica nova (ou pelo menos uma melhoria significativa de uma tecnologia existente) e defende sua relevância. Já a pesquisa de avaliação é a investigação de um problema ou uma implementação técnica na prática.

A pesquisa de validação investiga as propriedades de uma proposta de solução que ainda não foi implementada na prática, enquanto o artigo filosófico propõe uma nova visão sobre o assunto, uma nova estrutura conceitual.

Podemos perceber uma prevalência de artigos que se enquadram na categoria de propostas de solução, em especial propostas de solução de testes, que foi a área de domínio com mais casos de propostas de solução.

## 4.2 Análise dos dados coletados

Em seu estudo, Faria et al. atenta que para garantir a interoperabilidade e o comportamento correto de ponta a ponta de sistemas distribuídos heterogêneos em um número crescente de domínios, como IoT em cidades inteligentes. É apresentada a ferramenta DCO analyzer, que tem a função de analisar as propriedades de observabilidade e controlabilidade local de testes em sistemas distribuídos realizando o parsing de arquivos UML e checando sua conformidade com o meta-modelo UML esperado e então convertendo-os para as estruturas em VDM++ correspondente. Após isso, o sistema analisa o diagrama e julga se ele é localmente observável e controlável. Se não for, ele gera o menor conjunto de mensagens necessários para deixar o sistema de acordo com estas propriedades (FARIA, 2020).

Segundo Ferreira, apenas testar sistemas IoT usando ferramentas, métodos e abordagens tradicionais de software não são suficientes para as necessidades de testes em IoT com cenários, implantações e dispositivos reais, além de ser caro e ter lento tempo de feedback. Também não é possível implantar versões novas diretamente nos dispositivos de borda devido às restrições e compatibilidade de versões com outras camadas de sistema. É descrito duas estratégias de teste denominadas Testbed e Simulation-based Testing que desenvolvedores podem utilizar para testar sistemas IoT e uma estratégia de implantação (deployment) chamada Middleman Update, que ajudarão os desenvolvedores a manterem seus sistemas bem testados e os sistemas implantados atualizados (FERREIRA, 2019).

Lima et al. apresenta uma abordagem de teste de integração automatizada que visa auxiliar na garantia da interoperabilidade e integridade em sistemas distribuídos, através da verificação das interações com o ambiente e entre os componentes do sistema. Seu objetivo é desenvolver algoritmos para verificação de conformidade descentralizada e geração de entradas de teste, para verificar e fazer cumprir as condições que permitem a execução de testes descentralizados, no contexto de testes baseados em cenários de sistemas distribuídos com restrição de tempo (LIMA, 2019).

Restivo et al. desenvolvem uma prova de conceito criando um plugin para a ferramenta de desenvolvimento IoT chamada Node-RED, com a intenção de melhorar a capacidade de monitoramento e depuração, que segundo os autores são limitações das ferramentas que auxiliam no desenvolvimento de sistemas IoT atualmente. O plugin proposto permite aos usuários verificarem o estado do sistema em tempo de execução, observando a entrada e saída de um determinado nó, a usar mecanismos de debug como breakpoints e permite modificações em tempo de execução, injetando e modificando mensagens (RESTIVO, 2020).

Ma et al. apresentam uma plataforma de validação de rastreamentos (traces) obtidos nos dispositivos IoT. Seu diferencial é a de que ele é baseado em circuitos Field Programmable Gate Array (FPGA) e realiza análises online, permitindo uma melhora na capacidade de capturar os traces e analisá-los em tempo de execução de maneira mais completa. A arquitetura proposta consiste em três módulos principais, o Target (um dispositivo IoT a ser analisado), Observer (um hardware FPGA responsável por analisar os traces) e o Controller (servidor responsável por gerenciar todo processo, enviando os comandos e dados para os Observers e apresentando os resultados aos usuários) (MA, 2020).

O artigo de Rohner et al. explora o uso de medições ativas na camada IP visando ganho de observabilidade em IoT com foco em redes de sensores sem fio. Algumas das principais contribuições do seu estudo são a caracterização de recursos de time stamping e estimação de Round Trip Time (RTT) em dispositivos IoT, análise do equilíbrio entre frequência de medição, consumo de energia, RTT e perda auto-induzida em toda a rede (ROHNER, 2018).

Para ter sistemas bem gerenciáveis é necessário observar o comportamento do sistema. Dobaj et al. elencam alguns padrões que permitem a observação da aplicação e sua infraestrutura e visam fornecer mecanismos para identificar erros, falhas e bugs no sistema rapidamente. É proposto uma arquitetura de software baseada em microsserviços que combina ideias do domínio IoT, da automação industrial e de arquiteturas em nuvem com mecanismos que garantem interoperabilidade. Os autores concluem que a arquitetura proposta acompanhada de abordagens modernas de desenvolvimento (como containerização) a tornam pronta para implantação em ambientes IoT tanto industriais e domésticos (DOBAJ, 2018).

A pesquisa de Hili et al. demonstra a funcionalidade de uma arquitetura facilmente configurável e que contempla diversas ferramentas de monitoramento em tempo de execução. Argumenta-se sobre a importância do monitoramento em tempo de execução para a garantia de qualidade dos sistemas distribuídos e em tempo real, pois ajuda na identificação de bugs, problemas de performance e falhas de segurança. Resumidamente, é introduzido um Observador no sistema que fica responsável por encapsular as informações (como uso de memória) de cada evento chamado no sistema. Após isso, esta cápsula é enviada e consumida pelas ferramentas de monitoramento que reúnem as informações que julgarem relevantes e as apresentam ao usuário em tempo de execução (HILI, 2020).

Rabiee et al. estuda redes IoT de larga escala em termos de observabilidade e controlabilidade (RABIEE, 2020), já Krotofil et al. elencam várias questões relacionadas a segurança de IIoT (Industrial Internet of Things) e aborda temas como observabilidade e controlabilidade, detecção de intrusão, criptografia, computação confiável (trusted computing) e outras questões filosóficas acerca de segurança (KROTOFIL, 2019).

Matray et al. descrevem um método para monitorar os fluxos de dados em topologias de computação distribuída, a reconstruir relações causais e temporais, detectar degradações no desempenho, apontar as causas raízes para auxiliar na aplicação de ações corretivas antes que os requisitos sejam ameaçados (MATRAY, 2016).

Jacobs et al. levantam uma questão interessante em seu artigo que diz respeito ao elevado ritmo de penetração de dispositivos IoT nas redes elétrica, tecnologias como inversores e medidores inteligentes que são integrados à rede com conectividade e comunicações adicionais, tornando os sistemas cada vez mais ciberfísicos. Seu artigo propõe uma abordagem baseada na teoria dos grafos para definição observabilidade para sistemas ciberfísicos (JACOBS et al., 2020).

Investigamos que estão acontecendo pesquisas que tratam especificamente da observabilidade no contexto de IoT. Também percebemos que existe certo esforço da comunidade em pesquisar observabilidade sob a ótica dos testes de software. Pôde-se observar como um obstáculo importante, a dificuldade de simular ambientes reais para a finalidade de testar sistemas IoT.

*QP: Qual é o estado da prática da pesquisa sobre observabilidade em engenharia de software, em especial para sistemas IoT?*

Existem artigos de áreas correlatas que agregam conhecimento e nos auxiliam a ter um melhor entendimento desses conceitos relacionados a observabilidade. Analisando os artigos encontrados pelas strings de busca definidas neste estudo, podemos observar uma prevalência da prática de pesquisa de observabilidade com algumas ênfases que foram agrupadas nas áreas de domínio apresentadas na Tabela 4.

Com a leitura dos artigos filtrados por este MSL e de outros estudos correlatos, foi realizado uma análise de cada artigo e foi possível sintetizar na Tabela 6 alguns dos principais desafios e obstáculos reportados pelos autores.

Tabela 6 – Desafios reportados.

Desafios	Categorias
C1	Formatação de logs, análise de logs em tempo real e de múltiplas fontes
C2	Escala e elasticidade
C3	Vazamento de dados de monitoramento e segurança
C4	Simulação do ambiente real e custo monetário
C5	Heterogeneidade e falta de padrões
C6	Dinâmica e complexidade
C7	Planejamento organizacional

**Formatação de logs, análise de logs em tempo real e de múltiplas fontes.** Enquanto Deursen et al. tratam de alguns obstáculos envolvendo logs, em especial sobre o ciclo de vida, infraestrutura e análise dos logs, relatando desafios como a falta de padrão universal de logs e o seu ciclo de vida (DEURSEN, 2021), no artigo de Svacina, os autores elencam desafios como por exemplo o grande volume de dados, a dificuldade de derivar conhecimento a partir de arquivos de logs e obstáculos para análise de múltiplas fontes, que tem desafios como a propagação de erros entre sistemas, que dificulta o rastreamento (SVACINA, 2020).

**Escala e elasticidade.** Ghosh et al. relatam que existe uma grande dificuldade relacionada ao monitoramento de nuvem híbridas devido, entre outros fatores, a escala e complexidade, pois fica mais complexo de rastrear e detectar problemas (NATU et al., 2016). A elasticidade está entre os principais motivos que tornam a computação em nuvem uma tendência emergente. Os testes de elasticidade tem objetivos como controlar comportamentos diferentes, identificar os recursos a serem (des)alocados, coordenar eventos em paralelo e atingir escalabilidade. São encontrados desafios relativos a testes de capacidade de (des)provisionamento de recursos de nuvem, que impacta na execução dos testes e na sua perspectiva, uma vez que requer novas abordagens de testes específicas (BERTOLINO et al., 2019).

**Vazamento de dados de monitoramento e segurança.** Ghosh expõe desafios em seu artigo que englobam mais de uma categoria apresentada neste MSL, como vazamento de dados, segurança e custo monetário. Ele cita como exemplo, que para evitar vazamento de dados, é importante passar os dados (muitas vezes em grandes volumes) para uma nuvem privada (quando está alocado em nuvem pública), o que acarreta altos custos monetários. Também elencam desafios importantes como escala e complexidade, conhecimento incompleto e impreciso do sistema, entre outros (NATU et al., 2016). Já o estudo de Krotofil et al discutem questões relacionadas a segurança de sistemas IoT industriais (KROTOFIL, 2019).

**Simulação do ambiente real e custo monetário.** Bertolino destaca em seu trabalho a complexidade de simular adequadamente um ambiente IoT real para testes, uma vez que diversos fatores externos e internos influenciam nos resultados. A alta complexidade dos sistemas em nuvem e suas muitas configurações possíveis e o alto custo monetário para monitoramento em ambiente de nuvem impactam fortemente nas áreas de design, execução e avaliação de teste, e também afeta a perspectiva do teste (BERTOLINO et al., 2019).

**Heterogeneidade e falta de padrões.** Os sistemas distribuídos consistem de várias camadas, de aplicativos a tecnologias de infraestrutura como contêineres e VMs. Essas camadas são desenvolvidas e operadas por equipes heterogêneas (KOETTER, 2019). Ademais, a diversidade de tecnologias, de diferentes fabricantes, geram desafios relacionados

a falta de padrões como de registros e protocolos. No estudo de Bertolino, foi relatado que a falta de padrões também está relacionado à novidade da disciplina de computação em nuvem. Segundo o autor, foi encontrado esse desafio em estudos sobre execução e avaliação de texto, bem como sobre a portabilidade entre domínios de teste (BERTOLINO *et al.*, 2019).

**Dinâmica e complexidade.** A tendência emergente de arquiteturas de microsserviço, implantações em nuvem e DevOps aumentam a complexidade dos sistemas distribuídos. Enquanto a complexidade individual de um microsserviço é reduzida, a complexidade das interdependências dos microsserviços e dos componentes dinâmicos em um sistema distribuído causam mais esforços operacionais (KOETTER, 2019). Além disso, a complexidade também tem relação com a heterogeneidade, escala, elasticidade, falta de padrões, simulação de ambientes real para testes, entre diversos outros fatores.

**Planejamento organizacional.** Através da entrevista semiestruturada realizada pela RSL de Koetter, observou-se uma complexidade crescente no campo de observabilidade de sistemas distribuídos e da forte necessidade de um bom planejamento organizacional, incluindo estratégia, funções e responsabilidades bem definidas. Também foi relatado em seu estudo desafios relacionados a falta de experiência, tempo e recursos, além da dependência de profissionais especialistas e a cultura e mentalidade da empresa (KOETTER, 2019).

## 5 Ameaças a validade

Discutimos a ameaça em termos de quatro categorias (WOHLIN, 2012).

**Validade de Construção** A principal ameaça à generalização de nossas conclusões diz respeito à representatividade de nosso conjunto de dados. Nosso procedimento para descobrir artigos relevantes consiste em consultar bibliotecas digitais populares. Embora coletamos dados de quatro fontes diferentes, não está claro como cada biblioteca indexa as entradas. Com isso é possível que possa ter perdido algum artigo relevante porque nenhuma das bibliotecas digitais contempladas por este estudo o relatou. Portanto, o procedimento de pesquisa pode não ser capaz de produzir resultados completos.

**Validade Interna** Possíveis limitações podem estar relacionadas ao entendimento das informações apresentadas, que expressa o ponto de vista dos autores. No entanto, foram realizadas reuniões frequentes entre os autores para discutir a relevância dos artigos que deveriam continuar nas demais fases de nossa análise.

**Validade Externa** Esta ameaça está relacionada à representatividade dos estudos selecionados publicados entre 2016 e 2021 para os principais objetivos deste MSL. Nossas descobertas sobre o estado da prática da pesquisa sobre observabilidade realizadas durante o período de estudo selecionado são precisas até onde sabemos.

**Validade de Conclusão** O resumo do estado da prática de pesquisa e desafios apresenta o ponto de vista dos autores e pode não fornecer a definição real publicada pelos artigos.

## 6 Considerações Finais

Este mapeamento sistemático da literatura veio com a proposta de pesquisar o estado da prática da pesquisa sobre observabilidade em engenharia de software, em especial para sistemas IoT. Foi possível verificar que existem artigos de áreas como de testes, monitoramento, registros, rastreamento e métricas, entre outros, que agregam conhecimento e nos auxiliam a ter um melhor entendimento da observabilidade em IoT.

Destacamos a necessidade de pesquisas futuras que abordem a observabilidade voltada a sistemas IoT, já que tem peculiaridades e especificidades que demandam uma atenção particular. Como por exemplo, na simulação de ambientes reais de IoT para testes. Ademais, devido a heterogeneidade do ambientes IoT, com os mais diversos tipos de dispositivos dos mais variados fabricantes, são produzidos desafios como a falta de padrões de protocolos e registros. Podemos observar na literatura, uma considerável complexidade para observar sistemas IoT. Entretanto, observamos uma tendencia crescente de publicações envolvendo este conceito e acreditamos que a observabilidade será mais pesquisada.

Pretendemos como trabalho futuro fazer um survey com profissionais atuantes no mercado para identificar e analisar os benefícios e desafios da aplicação da observabilidade em projetos que desenvolvem sistemas IoT.

# Referências

ACETO, G. Cloud monitoring: a survey. International Conference on Service-Oriented Computing., 2013. Citado na página 15.

ATZORI, L. The internet of things: A survey. *Computer Networks*, v. 54, n. 15, p. 2787–2805, 2010. ISSN 1389-1286. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1389128610001568>>. Citado na página 12.

AVGERIOU, S. Empirical studies on software traceability: A mapping study. *Journal of Software: Evaluation and Process*, 2020. Citado na página 19.

BERTOLINO, A. et al. A systematic review on cloud testing. *ACM Comput. Surv.*, Association for Computing Machinery, New York, NY, USA, v. 52, n. 5, set. 2019. ISSN 0360-0300. Disponível em: <<https://doi.org/10.1145/3331447>>. Citado 3 vezes nas páginas 19, 30 e 31.

CANDIDO, J. Log-based software monitoring: a systematic mapping study. Delft University of Technology, 2019. Citado na página 13.

CHEN, L.; BABAR, M. A.; ZHANG, H. Towards an evidence-based understanding of electronic data sources. In: *Proceedings of the 14th International Conference on Evaluation and Assessment in Software Engineering*. Swindon, GBR: BCS Learning Development Ltd., 2010. (EASE'10), p. 135–138. Citado na página 21.

CUMMINGS R.; BROWNER, S. Delineando a pesquisa clínica. Artmed Editora., 2015. Citado na página 21.

DEURSEN, A. Log-based software monitoring: a systematic mapping study. 2012 34th International Conference on Software Engineering (ICSE), 2021. Citado 5 vezes nas páginas 10, 16, 17, 19 e 30.

DOBAJ, J. A Microservice Architecture for the Industrial Internet-Of-Things. European Conference on Pattern Languages of Programs, 2018. Citado na página 28.

FARIA, J. DCO analyzer: local controllability and observability analysis and enforcement of distributed test scenarios. International Conference on Software Engineering, 2020. Citado na página 27.

FERREIRA, H. Testing and deployment patterns for the internet-of-things. European Conference on Pattern Languages of Programs, 2019. Citado na página 27.

FLUCHTER, F. W. K. Internet of things Technology and Value Added. BISE - Business Information Systems Engineering, 2015. Citado na página 13.

GOPAL, M. *Modern Control System Theory*. USA: Halsted Press, 1984. ISBN 0470274247. Citado na página 14.

GOTEL, O. Software Traceability: Trends and Future Directions. FOSE 2014: Future of Software Engineering Proceedings., 2014. Citado na página 17.

- HILI, N. A model-based architecture for interactive run-time monitoring. *International Journal Software and Systems Modeling (SoSyM)*, 2020. Citado na página 28.
- HONEYCOMB. Observability For Developers. 2018. Disponível em: <<https://www.honeycomb.io/wp-content/uploads/2018/12/Observability-for-Developers-ebook-Dec-7-2018-.pdf>>. Citado na página 15.
- IEEE. Standard Glossary of Software Engineering Terminology. 1990. Citado 2 vezes nas páginas 13 e 17.
- INDRASIRI, P. S. K. *Microservices for the Enterprise - Designing, Developing, and Deploying*. 2018. Citado na página 15.
- JACOBS, N. et al. Cyber-physical observability for the electric grid. In: *2020 IEEE Texas Power and Energy Conference (TPEC)*. [S.l.: s.n.], 2020. p. 1–6. Citado na página 29.
- KAROUNTZOS, S. C. P. A. E. Empirical studies on software traceability: A mapping study. *Journal of Software Evolution and Process.*, 2020. Citado na página 18.
- KINSELLA, J. The cloud complexity gap: making software more intelligent to address complex infrastructure. 2015. Disponível em: <<https://cloudcomputing-news.net/news/2015/jun/17/cloud-complexity-gap-making-software-more-intelligent-address-complex-infrastructure/>>. Citado na página 10.
- KITCHENHAM, A. Guidelines for performing systematic literature reviews in software engineering. 2007. Citado na página 21.
- KNOCHE, W. H. H. Drivers and barriers for microservice adoption a survey among professionals in Germany. *Enterp. Model. Inf. Syst. Architect.*, 2019. Citado na página 10.
- KOETTER, F. On Observability and Monitoring of Distributed Systems – An Industry Interview Study. *International Conference on Service Oriented Computing.*, 2019. Citado 4 vezes nas páginas 10, 19, 30 e 31.
- KROTOFIL, M. Securing Industrial Control Systems. *Security and Privacy Trends in the Industrial Internet of Things*, 2019. Citado 2 vezes nas páginas 28 e 30.
- LIMA, B. Automated Scenario-Based Integration Testing of Time-Constrained Distributed Systems. 2019 12th IEEE Conference on Software Testing, Validation and Verification (ICST), 2019. Citado 2 vezes nas páginas 10 e 27.
- MA, J. Y. S. A testbed for hardware-assisted online profiling of iot devices. In: *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*. New York, NY, USA: Association for Computing Machinery, 2020. (ICSEW'20), p. 627–630. ISBN 9781450379632. Disponível em: <<https://doi.org/10.1145/3387940.3392247>>. Citado na página 28.
- MAGRANI, E. A internet das coisas. Fundação Getúlio Vargas., 2018. Citado na página 12.
- MATRAY, P. OData-Driven Monitoring for Cloud Compute Systems. *IEEE/ACM International Conference on Utility and Cloud Computing*, 2016. Citado na página 29.

- NATU, M. et al. Holistic performance monitoring of hybrid clouds: Complexities and future directions. *IEEE Cloud Computing*, v. 3, n. 1, p. 72–81, 2016. Citado na página 30.
- NEWRELIC. A Era da Observabilidade. 2021. Disponível em: <<https://newrelic.com/resources/ebooks/what-is-observability-pt>>. Citado na página 14.
- NGUYEN, P. et al. Advances in deployment and orchestration approaches for iot - a systematic review. In: *2019 IEEE International Congress on Internet of Things (ICIOT)*. [S.l.: s.n.], 2019. p. 53–60. Citado na página 19.
- PICORETI, R. et al. Multilevel observability in cloud orchestration. In: *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*. [S.l.: s.n.], 2018. p. 776–784. Citado na página 14.
- RABIEE, H. On the Observability and Controllability of Large-Scale IoT Networks: Reducing Number of Unmatched Nodes via Link Addition. *IEEE Control Systems Letters (L-CSS)*, 2020. Citado na página 28.
- RESTIVO, A. Real-time Feedback in Node-RED for IoT Development: An Empirical Study. *Symposium on Distributed Simulation and Real-Time Applications (DS-RT)*, 2020. Citado na página 27.
- ROHNER, C. On performance observability in iot systems using active measurements. In: *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*. [S.l.: s.n.], 2018. p. 1–5. Citado na página 28.
- SUN, C. ai. Constraint-based model-driven testing of web services for behavior conformance. *International Conference on Service-Oriented Computing*, 2018. Citado na página 15.
- SUNEJA, S. Non-intrusive, out-of-band and out-of-the-box systems monitoring in the cloud. *The 2014 ACM International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS*, 2014. Citado na página 14.
- SVACINA, J. On Vulnerability and Security Loganalysis: A Systematic Literature Review on Recent Trends. *Proceedings of the International Conference on Research in Adaptive and Convergent Systems*, Korea, 2020. Citado na página 30.
- VALE, T. et al. Software product lines traceability: A systematic mapping study. *Information and Software Technology*, v. 84, p. 1–18, 2017. ISSN 0950-5849. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0950584916304463>>. Citado na página 18.
- WIERINGA, R. et al. Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. *Requir. Eng.*, Springer-Verlag, Berlin, Heidelberg, v. 11, n. 1, p. 102–107, dez. 2005. ISSN 0947-3602. Disponível em: <<https://doi.org/10.1007/s00766-005-0021-6>>. Citado na página 26.
- WOHLIN, C. *Experimentation in Software Engineering*. Springer Publishing Company, 2012. Citado na página 32.

WOHLIN, C. Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. International Conference on Evaluation and Assessment in Software Engineering (EASE), 2014. Citado na página 23.

YANAN, Z. The Research on Software Metrics and Software Complexity Metrics. 2009 International Forum on Computer Science-Technology and Applications., 2009. Citado na página 17.

YUAN, D.; PARK, S.; ZHOU, Y. Characterizing logging practices in open-source software. In: *2012 34th International Conference on Software Engineering (ICSE)*. [S.l.: s.n.], 2012. p. 102–112. Citado na página 16.

ZISMAN, C. O. G. Software and Systems Traceability. Springer Publishing Company., 2012. Citado na página 17.