



# Sobre Observabilidade e Monitoramento de Sistemas Distribuídos – Uma Indústria

## Estudo de entrevista

Sina Niedermaier<sup>1</sup>, Falko Koetter<sup>2(B)</sup>, Andreas Freymann<sup>2</sup>,  
e Stefan Wagner<sup>1</sup>

<sup>1</sup> Instituto de Tecnologia de Software, Universidade de Stuttgart, Stuttgart, Alemanha  
{sina.niedermaier,stefan.wagner}@iste.uni-stuttgart.de Instituto Fraunhofer

<sup>2</sup> de Engenharia Industrial IAO, Fraunhofer IAO,  
Estugarda, Alemanha  
{falko.koetter,andreas.freymann}@iao.fraunhofer.de

**Resumo.** O sucesso empresarial das empresas depende muito da disponibilidade e do desempenho de seus aplicativos clientes. **Devido à modernidade paradigmas de desenvolvimento como DevOps e arquitetura de microserviços estilos, as aplicações são desacopladas em serviços com interações complexas e dependências. Embora esses paradigmas permitam ciclos de desenvolvimento individuais com tempos de entrega reduzidos, eles causam vários desafios para gerenciar os serviços em sistemas distribuídos. Um grande desafio é para observar e monitorar tais sistemas distribuídos.** Este artigo fornece um estudo qualitativo para compreender os desafios e as boas práticas no campo da observabilidade e monitoramento de sistemas distribuídos. Em 28 entrevistas semiestruturadas com profissionais de software que descobrimos crescente complexidade e dinâmica nesse campo. Especialmente a observabilidade torna-se um pré-requisito essencial para garantir serviços estáveis e desenvolvimento de aplicações cliente. No entanto, os participantes mencionaram uma discrepância na conscientização sobre a importância do tópico, tanto da perspectiva da gerência quanto da perspectiva do desenvolvedor. **Além dos desafios técnicos, identificamos uma forte necessidade de um conceito organizacional incluindo estratégia, papéis e responsabilidades.** Nossos resultados apoiar os profissionais no desenvolvimento e implementação de observabilidade e monitoramento sistemáticos para sistemas distribuídos.

**Palavras-chave:** Monitoramento · Observabilidade · Sistemas distribuídos · Nuvem · Indústria

## 1 Introdução

Nos últimos anos, muitos departamentos de TI migraram com sucesso seus serviços para a computação em nuvem [21]. Ainda assim, os desafios para a adoção da nuvem permanecem em relação à operação e monitoramento holístico de tais serviços [16]. **Enquanto as TI convencionais a infraestrutura pode ser monitorada com soluções de monitoramento convencionais, nuvem os ambientes são mais dinâmicos e complexos [1], resultando numa lacuna [12] entre a complexidade dos sistemas distribuídos e a capacidade das ferramentas de monitoramento para gerenciar essa complexidade.**

© Springer Nature Suíça AG 2019

Português S. Yangui et al. (Eds.): ICSSOC 2019, LNCS 11895, pp. 36–52, 2019.

[https://doi.org/10.1007/978-3-030-33702-5\\_3](https://doi.org/10.1007/978-3-030-33702-5_3)

Tendências emergentes como a Internet das Coisas (IoT) e os microsserviços aumentam ainda mais a complexidade, tornando a monitorização uma barreira significativa para a adoção destas tecnologias [13].

Enquanto as ferramentas de software recém-surgidas tentam preencher essa lacuna de complexidade, o caminho a seguir para muitas empresas não está claro. Descobrimos que não há pesquisas que combinem novas soluções e tecnologias com diferentes áreas de aplicação, problemas e desafios. Embora exista uma infinidade de novas tecnologias e abordagens, as empresas precisam ser capazes de relacionar essas tecnologias aos desafios que enfrentam. Processos e boas práticas são necessários para incorporar novas soluções em arquiteturas empresariais existentes, bem como em arquiteturas de nuvem emergentes.

Para atender a essa necessidade, conduzimos um estudo de entrevistas da indústria entre diferentes stakeholders envolvidos no monitoramento, incluindo gerentes de serviço, engenheiros de DevOps, provedores de software e consultores. Das entrevistas semiestruturadas, extraímos desafios, requisitos e soluções contemporâneos.

## 2 Trabalhos relacionados

Para fornecer contexto à pesquisa descrita neste trabalho, o trabalho relacionado investiga (1) abordagens atuais para preencher a lacuna entre a complexidade do sistema distribuído e a capacidade de monitoramento, bem como (2) pesquisas anteriores sobre monitoramento e observabilidade (ver Fig. 1).

O IEEE define monitoramento como a supervisão, registro, análise ou verificação da operação de um sistema ou componente [10].

O termo Observabilidade se origina na teoria do sistema de controle e mede o grau em que o estado interno de um sistema pode ser determinado a partir de sua saída [7]. Em ambientes de nuvem, a observabilidade indica em que grau a infraestrutura e os aplicativos e suas interações podem ser monitorados. As saídas usadas são, por exemplo, logs, métricas e rastros [18].

Yang et al. [24] investigam a captura de caminhos de execução de serviços em sistemas distribuídos. Embora capturar o caminho de execução seja desafiador, pois cada solicitação pode cruzar muitos componentes de vários servidores, eles introduzem uma metodologia genérica de ponta a ponta para capturar a solicitação inteira. Durante nossas entrevistas, descobrimos uma necessidade de transparência dos caminhos de execução, bem como interdependências mais gerais entre os serviços.

A tendência atual em direção a sistemas distribuídos mais flexíveis e modulares é caracterizada pelo uso de serviços independentes, como microsserviços ou serviços web. Embora sistemas que consistem em serviços web forneçam melhor observabilidade do que sistemas monolíticos, os serviços têm o potencial de melhorar sua observabilidade e monitoramento, fornecendo informações relevantes sobre seu comportamento interno. Sun et al. [23] lidam com o desafio de que as definições de serviços web não têm nenhuma informação sobre seu comportamento. Eles estendem a definição de serviços web adicionando uma descrição lógica de comportamento com base em uma abordagem de teste orientada a modelos baseada em restrições. Durante nossas entrevistas, identificamos que o comportamento, especialmente de serviços de terceiros, precisa ser comunicado de forma mais clara para avaliar o impacto nos níveis de serviço e para detectar e diagnosticar falhas.

Além de monitorar chamadas de serviço individuais, é importante prever o desempenho do tempo de execução de sistemas distribuídos. Johng et al. [11] mostram que dois

técnicas, benchmarking e simulação, apresentam deficiências se forem utilizadas separadamente e introduzir e validar uma abordagem complementar. Sua abordagem apresenta um processo que mapeia ontologias de referência de simulações. Isso provar ser barato, rápido e confiável. Da mesma forma, Lin et al. [14] propõem uma nova maneira de detecção de causa raiz em arquiteturas de microserviços utilizando causal gráficos. Em nossas entrevistas, descobrimos que o desempenho geralmente só é conhecido quando um sistema entra em operação, à medida que as interdependências entre os diferentes serviços e seus o desempenho individual não é avaliado previamente.

Gupta et al. [8] aborda o monitoramento de tempo de execução em implantação contínua no desenvolvimento de software como uma tarefa crucial, especialmente em soluções de software que mudam rapidamente. Embora as abordagens atuais de monitoramento de tempo de execução de anteriores e versões recém-implantadas não conseguem capturar e monitorar diferenças em tempo de execução, eles apresentam uma abordagem que descobre automaticamente um comportamento de execução modelo por logs de execução de mineração. Abordagens como esta que reúnem informações automaticamente em vez de necessitar de definição manual são cruciais com o crescimento complexidade e dinâmica de sistemas distribuídos.

Esses trabalhos mostram que para a pesquisa sobre o fechamento da lacuna de complexidade entre ambientes de nuvem e seu monitoramento é contínuo. No entanto, essas soluções são ainda não amplamente adotada na prática. Ao adotar novas tecnologias para a indústria aplicação, requisitos não funcionais como usabilidade, configurabilidade e adaptabilidade aumenta em importância.

Nos inquéritos anteriores no contexto da monitorização e observação habilidade (2) são descritas.

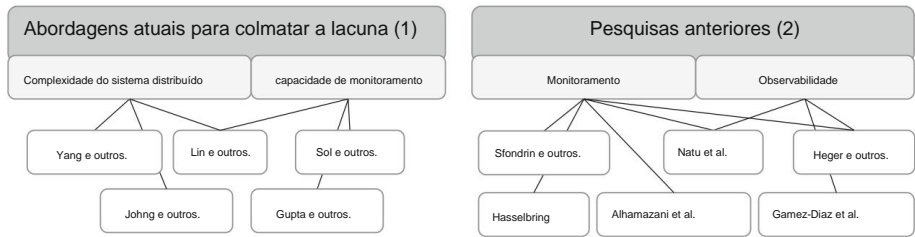


Fig. 1. Visão geral do trabalho relacionado

Aceto et al. [1] conduziram uma pesquisa abrangente de monitoramento de nuvens em 2013, detalhando motivações, suporte de ferramentas então atual e desafios abertos. Eles identificaram a necessidade de escalabilidade, robustez e flexibilidade. A pesquisa previu corretamente o aumento da complexidade e dinâmica das arquiteturas de nuvem e propôs ações para lidar com isso, como detecção de causa raiz, filtragem/resumo de dados e monitoramento entre camadas/plataformas. Da mesma forma, outra pesquisa inicial de ferramentas de monitoramento de nuvem por Fatema et al. [5] identifica capacidades como escalabilidade, robustez, interoperabilidade e personalização para encontrar uma lacuna entre capacidades necessárias e ferramentas existentes.

Sfondrin et al. [21] conduziram uma pesquisa com 62 empresas multinacionais sobre a adoção da nuvem pública. Embora o uso da infraestrutura de nuvem pública esteja aumentando, barreiras como segurança, conformidade regulatória e monitoramento permanecem. Em relação ao monitoramento, a pesquisa mostrou que metade das empresas depende exclusivamente do painel de monitoramento de seus provedores de nuvem. Os participantes notaram uma necessidade crucial de monitoramento da qualidade do serviço integrado com sua ferramenta de monitoramento.

Da mesma forma, Knoche e Hasselbring [13] conduziram uma pesquisa com especialistas alemães sobre adoção de microsserviços. Os motivadores para adoção de microsserviços são escalabilidade, manutenibilidade e velocidade de desenvolvimento. Por outro lado, as barreiras para adoção são principalmente de natureza operacional. O departamento de operações resiste aos microsserviços devido à mudança em suas tarefas. No nível técnico, executar aplicativos distribuídos propensos a falhas parciais e monitorá-los é um desafio significativo.

Gamez-Diaz et al. [6] realizaram uma análise de APIs RestFUL de provedores de nuvem, identificando requisitos para governança de API e observando uma falta de padronização.

Embora não seja um estudo empírico, Natu et al. [16] mostram desafios de monitoramento de aplicações holísticas de nuvem. Escala e complexidade de aplicações são identificadas como um desafio principal. Relacionados à observabilidade, visões incompletas e imprecisas do sistema total, bem como localização de falhas são outros desafios identificados.

Heger et al. [9] dão uma visão geral do estado da arte em monitoramento de desempenho de aplicativos (APM), descrevendo capacidades típicas e software APM disponível. Eles descobriram que o APM é uma solução para monitorar e analisar ambientes de nuvem, mas observam desafios futuros na detecção de causa raiz, esforço de configuração e interoperabilidade. O APM não pode mais ser entendido como um tópico puramente técnico, mas precisa incorporar aspectos comerciais e organizacionais também.

Alhamazani et al. [2] dão uma visão sobre ferramentas comerciais de monitoramento de nuvem, mostrando recursos de última geração, identificando deficiências e, conectado a isso, futuras áreas de pesquisa. Agregação de informações em diferentes camadas de abstração, uma ampla gama de métricas mensuráveis e extensibilidade são vistas como fatores críticos de sucesso. As ferramentas foram consideradas carentes de padronização em relação aos processos e métricas de monitoramento.

Comparando pesquisas anteriores sobre monitoramento e observabilidade (2) com nosso trabalho, essas pesquisas se concentram em drivers e desafios ou em soluções disponíveis (em ferramentas científicas e comerciais). Em comparação, nosso estudo adota uma abordagem holística. Fornecemos pesquisa empírica focada na indústria com entrevistas em profundidade, onde combinamos diferentes perspectivas para descobrir quais soluções e estratégias emergentes são usadas pelas empresas e em que grau elas superaram os desafios existentes. Isso é necessário para avaliar a adoção de novas tecnologias na prática. Além disso, compreende quais desafios essas tecnologias abordam e quais desafios surgem na adoção ou não são resolvidos.

### 3 Escopo e Método de Pesquisa

Desenho do estudo: Para estruturar nossa pesquisa, aplicamos o processo de pesquisa de estudo de caso de cinco etapas, conforme descrito por Runeson e Hoest [19]. Nosso objetivo de pesquisa

pode ser definido como segue: Análise dos desafios contemporâneos de monitoramento e operação de sistemas distribuídos com o propósito de deduzir requisitos e mapear soluções e estratégias existentes do ponto de vista de diferentes partes interessadas de sistemas de monitoramento e provedores de ferramentas. A Tabela 1 resume as questões de pesquisa:

Tabela 1. Visão geral das questões de pesquisa

RQ1	Quais desafios contemporâneos existem no monitoramento de sistemas distribuídos?
RQ2	Quais requisitos as partes interessadas têm para um conceito de monitoramento e observabilidade para sistemas distribuídos?
RQ3	Quais são as estratégias e soluções técnicas e organizacionais nas empresas?

Para responder às nossas questões de pesquisa, aplicamos o método qualitativo de entrevistas semiestruturadas. Elas nos permitem explorar os desafios individuais declarados pelos participantes e analisar as relações subjacentes, fornecendo uma agenda básica. Ao mesmo tempo, as entrevistas permitem uma interação dinâmica com base no histórico de nossos especialistas e suas respostas [22].

No total, realizamos 28 entrevistas semiestruturadas com duração média de 45 minutos. As entrevistas foram concluídas entre fevereiro e abril de 2019. Para atingir uma distribuição equilibrada dos entrevistados, primeiro, consideramos usuários que usam soluções de monitoramento e provedores de ferramentas que oferecem soluções de monitoramento (consulte a Tabela 2). Em segundo lugar, garantimos que os provedores de soluções e usuários estejam relacionados a diferentes domínios e se concentrem em obter perspectivas diversas de soluções de monitoramento. Além dos provedores de ferramentas, cobrimos outros domínios, como software e serviço de TI, IoT, telecomunicações, seguros e consultoria de TI. Os usuários foram selecionados de diferentes pontos de vista na pilha de aplicativos e diferentes funções, como DevOps e engenheiros de suporte, juntamente com proprietários e gerentes de produtos. O recrutamento de participantes foi obtido por meio de contatos pessoais no setor, bem como por aquisição em conferências de desenvolvedores.

Preparação para coleta de dados: Para conduzir as entrevistas semiestruturadas, criamos um guia de entrevista [17]. O guia é estruturado em diferentes blocos temáticos para agrupar as perguntas individuais. Os entrevistados foram pré-informados sobre o escopo e o procedimento das entrevistas. Além das informações para tratar suas transcrições como confidenciais, pedimos para gravar as entrevistas para criar transcrições, se permitido. Além disso, os informamos sobre a possibilidade de revisar suas transcrições para concordar com as informações fornecidas na entrevista.

Coleta de dados: Das 28 entrevistas, 15 foram conduzidas 'face a face' e 13 por meio de comunicação remota. As entrevistas foram realizadas em alemão, exceto por duas entrevistas em inglês. Enquanto 21 entrevistas foram gravadas em áudio, para as entrevistas restantes dois pesquisadores criaram protocolos para reduzir o viés do pesquisador. Durante as entrevistas, seguimos vagamente o guia de entrevista de acordo

Tabela 2. Visão geral sobre os participantes e empresas

Domínio CID	Funcionários	Função de especialista em EID	Foco
C1 IoT	>100T	E1 Proprietário do produto	Solução APM
		E2 Arquiteto Líder	Infraestrutura de Nuvem
		Proprietário do produto E3	Backend de conectividade
		Gerente de serviço E4	Suporte e Operação de Solução IoT
		Arquiteto de Nuvem E5	Backend de IoT
C2 IoT	100-1T	E6 Consultor/Arquiteto de IoT	Consultoria de Projetos de IoT
		Serviço de nuvem para desenvolvedores de código aberto E7	
		Engenheiro DevOps E8	Serviço em Nuvem
		Arquiteto DevOps E9	Serviço em Nuvem
		Proprietário do produto E10	Serviço em Nuvem
		Líder do Projeto E11	Projeto IoT
C3 IoT	Gerente	10T-100T E12	Plataforma IoT
C4 IoT		Proprietário da solução IoT 10T-100T E13	Serviço em Nuvem
C5 Telecomunicações		Proprietário do produto 10T-100T E14	Plataforma de Monitoramento
Software C6 e Serviços de TI	>100T	E15 Ex-Chefe de Tecnologia Oficial	Desenvolvimento de Software e Ferramenta de Operações
		E16 Líder Técnico	Solução de Operação e
		Operações de TI	Gestão de Eventos
C7 Aplicado Pesquisar	Engenheiro DevOps	100-1T E17	Serviço de Seguros
		Desenvolvedor E18	Front- e Backend de Gestão de frotas
Fornecedor de ferramentas	C8 1T-10T	E19 Engenheiro de vendas	Solução APM
Fornecedor de ferramentas	C9 10T-100T	E20 Diretor estratégico	Monitoramento de Infraestrutura <small>Ferramenta</small>
		Suporte E21	Monitoramento de Infraestrutura <small>Ferramenta</small>
C10 Tool Provider 100-1T E22 Desenvolvedor e Arquiteto Monitoramento de Infraestrutura <small>Ferramenta</small>			
C11 Provedor de ferramentas 100-1T E23 Desenvolvedor <small>Ferramenta de monitoramento</small>			
Serviço de TI C12 Seguro	Diretor de Divisão	1T-10T E24 Monitoramento	Monitoramento de desempenho
C13 Consultoria de TI 1-25		E25 Desenvolvedor e Arquiteto	Consultoria de TI Monitoramento
C14 Consultoria de TI 100-1T E26 Diretor Executivo Processo de Negócios  Monitoramento			
C15 Software e Serviços de TI	Arquiteto de soluções	10T-100T E27	Tecnologia de código aberto Provedor
C16 Software e Serviços de TI	Desenvolvedor	10T-100T E28	Empilhamento cruzado
			Instrumentação para
			Monitoramento e Depuração

\*CID = ID da empresa, \*EID = ID do especialista

às respostas e aos focos dos participantes. Após transcrever manualmente as entrevistas, enviamos as transcrições aos participantes para revisão, onde eles tiveram a possibilidade de corrigir declarações não intencionais ou remover dados sensíveis.

**Análise de dados:** Para a análise das transcrições individuais, codificamos o material para extrair categorias importantes em relação ao nosso objetivo de pesquisa. Para esse propósito, seguimos a abordagem de análise de conteúdo qualitativa de Mayring [15]. Nós codificamos abertamente as transcrições aplicando desenvolvimento de categoria indutiva, onde analisamos as transcrições no nível da frase. Normalmente, um código era atribuído a diferentes frases em uma transcrição e, além disso, uma frase poderia ser atribuída a mais de um código. Durante a análise, formamos hierarquias de códigos e subcódigos. Em várias iterações, os códigos foram revisados, divididos ou mesclados.

## 4 Resultados e Discussão

Esta seção apresenta as descobertas agregadas da análise da entrevista com foco em nossas questões de pesquisa definidas na Seção 3. Criamos uma hierarquia de categorias como uma abstração dos códigos definidos durante a análise das transcrições. Este artigo apresenta a hierarquia de nível superior dos desafios, requisitos e soluções identificados. A seguir, descrevemos os diferentes códigos gerados de acordo com as questões de pesquisa e ilustramos as respostas dadas para os códigos com algumas declarações exemplares dos especialistas (consulte a Tabela 2 do EID).

### 4.1 Desafios

A primeira questão de pesquisa (RQ1) visa entender os desafios com os quais nossos participantes lidam no campo de sistemas distribuídos e quais implicações estão mais relacionadas a esses desafios. Identificamos um conjunto de nove desafios (Cx) e suas implicações correspondentes, que são descritas a seguir.

Dinâmica e complexidade crescentes (C1): A tendência emergente de arquiteturas de microsserviços, implantações em nuvem e DevOps aumenta a complexidade dos sistemas distribuídos. Enquanto a complexidade individual de um microsserviço é reduzida, a complexidade das interdependências de microsserviços e os componentes dinâmicos dentro de um sistema distribuído causam mais esforço operacional. Este ambiente dinâmico não é gerenciável manualmente e abordagens tradicionais como o Configuration Management Database (CMDB) [4] não são mais suficientes: “CMDBs são frequentemente baseados em pesquisas e obtêm o estado do sistema uma vez por semana. Em uma semana, muita coisa aconteceu no sistema em nuvem, o que um CMDB não pode cobrir.” (E16). Este problema não inclui apenas arquiteturas de microsserviços nativas da nuvem, mas também sistemas historicamente desenvolvidos, onde uma visão geral das dependências do serviço está faltando. Além disso, alguns participantes declararam uma subestimação da complexidade dinâmica de seus sistemas. Isso causou

em caso de problema (especialmente para o diagnóstico de falhas dependentes do contexto ou não permanentes), a duração média para detecção e recuperação demorava muito.

**Heterogeneidade (C2):** Os sistemas distribuídos de hoje consistem em várias camadas: de aplicação a tecnologias de infraestrutura como contêineres, VMs ou mesmo ambientes sem servidor. Essas camadas são desenvolvidas e operadas por equipes. Além disso, como afirmaram nossos participantes, os sistemas geralmente contêm legados e tecnologia de serviço moderna em paralelo, onde ferramentas adicionais são necessárias para integrar componentes legados. Com relação aos sistemas multilocatários, alguns os participantes experimentaram um efeito de vizinho barulhento, onde um inquilino monopoliza recursos e afetar negativamente outros inquilinos na mesma infraestrutura. No entanto, neste caso, os participantes não conseguiram separar as visões entre diferentes inquilinos. Em termos de heterogeneidade tecnológica e velocidade de inovação, os participantes tiveram opiniões divergentes. Para desenvolvedores de sistemas distribuídos, pode-se escolher o tecnologia mais adequada por um lado, mas por outro lado, a heterogeneidade tecnológica dificulta a aplicação consistente de ferramentas de monitoramento.

Além disso, outros participantes criticaram a velocidade da inovação e alguns exigem uma desaceleração dos modismos tecnológicos ao definir regulamentações. A heterogeneidade nessas diferentes áreas está levando a uma visão geral ausente do sistema geral, são componentes individuais e as solicitações processadas.

**Cultura e mentalidade da empresa (C3):** A maioria dos participantes acredita que aspectos culturais e de mentalidade referentes ao monitoramento são essenciais. Vários até afirmou que este aspecto é mais desafiador do que os aspectos técnicos. Além disso, alguns entrevistados também mencionaram que uma transparência holística para aplicar o monitoramento muitas vezes não é pretendida. Isso dá origem, por exemplo, ao perigo de ser responsabilizado em retrospecto por um fracasso. Muitas vezes, os participantes descreveram que as equipes não ter uma visão geral fora de sua própria área, por exemplo, do contexto empresarial de seu serviço. Isso causou conceitos isolados de monitoramento e operação sem contexto para soluções de clientes e requisitos relacionados. No geral, a colaboração e a comunicação entre as equipes e a perspectiva a partir da qual elas se desenvolvem e operar seus serviços são frequentemente fracamente pronunciados. Isso ilustra a seguinte declaração (E22): “Normalmente não é a ignorância ou a incapacidade das pessoas na empresa, mas o ponto de vista errado. Muitas vezes os desenvolvedores estão tão enterrados em seu ambiente problemático, tão absorvidos em suas tarefas diárias que não conseguem não podem mais se dar ao luxo de mudar a si mesmos.”

**Falta de ponto de vista central (C4):** Os participantes declararam possibilidades limitadas em termos de visibilidade e dependências de outros serviços e equipes. Isso resulta por sua vez, em uma visão geral ausente de todo o sistema. E6 descreve, por exemplo, tal situação: “Se chegar a problemas tais que haja muitas reclamações de usuários porque o sistema não está funcionando corretamente, todos foram para solução de problemas. Devido a a falta de uma visão geral, era difícil diagnosticar as falhas. Foram necessários vários rodadas de escalada e teleconferências para discutir onde a falha está localizada.” ao mesmo tempo, identificamos a falta de uma pessoa responsável para gerar visões gerais e, assim, permitir que equipes individuais colaborem. o ponto mencionado é a falta de transparência sobre o impacto na disponibilidade e



desempenho de componentes integrados de terceiros que frequentemente fazem parte de um sistema distribuído. Devido ao fato de que para tais componentes os parâmetros de serviço geralmente não são acessíveis, pontos cegos permanecem e impedem um monitoramento geral.

**Inundação de dados (C5):** Os participantes mencionaram a inundação avassaladora de dados provenientes do sistema distribuído, que está em constante mudança. O desafio identificado é criar conclusões significativas a partir de alertas do cliente e como priorizá-los. Isso mostra a seguinte declaração (E17): "O volume e a quantidade de alertas são desafiadores no momento, não conseguimos priorizar os que impactam o cliente." Além disso, para problemas em que uma solicitação precisa ser tratada por vários componentes desenvolvidos por equipes independentes, é muito complexo identificar a localização das falhas, incluindo as responsabilidades para corrigi-las. Mais detalhadamente, muitos participantes descreveram a complexidade na correlação de métricas e logs com registro de data e hora de vários serviços, o que geralmente é acompanhado por metadados insuficientes. Além disso, os participantes declararam a ausência de um painel compreensível que permita visualizações navegáveis pelos dados.

**Dependência de especialistas (C6):** O processo de detecção e diagnóstico de falhas, que geralmente são realizados manualmente, parece ser altamente dependente do conhecimento de especialistas individuais sobre o design e o comportamento dos sistemas. Como a seguinte declaração de E11 ilustra, esses especialistas aparecem como uma 'fonte para depurar': "Essa forma de solução de problemas depende muito do conhecimento especializado dos membros da equipe [...]. Na maioria das vezes, o conhecimento sobre a estrutura do serviço é atualmente mais crucial do que um monitoramento que indique especificamente 'pesquisar neste ponto'." Esse desafio novamente descreve o desenvolvimento sistemático ausente de sistemas de monitoramento no suporte a humanos na detecção e diagnóstico de falhas.

**Falta de experiência, tempo e recursos (C7):** Muitos participantes descreveram o desafio de dominar tecnologias de microserviços e o paradigma DevOps, que exigem esforço adicional para operação, mas, ao mesmo tempo, faltam engenheiros DevOps qualificados. Especialmente, o curto tempo de comercialização resulta em uma priorização de recursos e em uma desconsideração de requisitos não funcionais, como disponibilidade ou desempenho. A maioria dos participantes mencionou o tempo limitado como razão para um desenvolvimento iterativo, muitas vezes reativo, de observabilidade e monitoramento do sistema.

**Requisitos não funcionais pouco claros (C8):** De acordo com os entrevistados, requisitos não funcionais como disponibilidade e desempenho, também conhecidos como Qualidade de Serviço (QoS) ou Níveis de Serviço (SL), geralmente não são definidos e controlados ou são insuficientemente definidos. Além disso, alguns participantes comentaram que as equipes geralmente não estão cientes de seus principais QoSs, bem como de sua importância no contexto do que precisa ser medido e monitorado. As razões relacionadas a isso são devido a requisitos do cliente ausentes ou imprecisos ou devido à falta de conscientização sobre a importância dos requisitos não funcionais (E6): "É muito importante pensar sobre níveis de serviço ou KPIs e defini-los de uma certa maneira. Isso geralmente é subestimado. Em muitos projetos, pode-se determinar que os gerentes de projeto têm apenas uma visão puramente técnica do sistema, sem estar cientes da disponibilidade e do desempenho necessários." Outra razão

identificamos a complexidade de definir a disponibilidade e o desempenho geral objetivos, que então devem ser convertidos em objetivos para serviços de componentes. Isto é ainda mais intensificada por restrições de tempo que levam a implementações reativas conforme consta na seguinte declaração (E6): "Muitas equipes de desenvolvimento estão sob pressão para levar o serviço ao mercado o mais rápido possível. Então, as equipes geralmente começam a desenvolver sem requisitos específicos do cliente e acabam em produção sem quaisquer requisitos sistematicamente derivados." A falta de requisitos definições leva, por sua vez, à falta de ciclos de feedback (E4): "[...] onde a qualidade 'falta controle na prestação de serviços'.

Implementação reativa (C9): Conforme afirmado em várias entrevistas, a falta de clareza requisitos e a falta de indicação e controle suficientes muitas vezes leva ao fracasso. Nestes casos, o desenvolvimento do monitoramento foi desencadeado por uma falha na produção, onde as equipes reconheceram uma falta de observabilidade para diagnosticar falha do cliente ou mesmo detectá-los. Na verdade, os clientes frequentemente recebiam níveis de serviço inadequados. Em vários exemplos, as equipes estavam ocupadas apenas com solução de problemas, o que por sua vez resultou em soluções ad hoc, em vez de criar soluções de monitoramento sistematicamente derivadas. Além disso, identificamos que as equipes deparam-se com os mesmos problemas, onde o desenvolvimento intensivo de mão-de-obra para monitorização serviços individuais são criados e os efeitos de sinergia do compartilhamento de conhecimento, expertise e boas práticas não são usados. Uma outra razão para a implementação reativa é que durante o desenvolvimento, os desenvolvedores não tinham conhecimento suficiente sobre as interações complexas na produção e, portanto, os pontos cegos permaneceram até a operação.

## 4.2 Requisitos e Soluções

Em relação a RQ 2 e RQ 3, os seguintes requisitos (Rx) e possíveis soluções (Sx) são listados. Rx e Sx foram extraídos das entrevistas.

mapeamento dos desafios descritos anteriormente, os requisitos correspondentes assim como as soluções são mostradas na Fig. 2.

Abordagem holística (R1): Juntamente com C4, os participantes da entrevista declararam que eles caracterizam o monitoramento "[...] como um problema holístico e tentar chegar a uma solução como uma abordagem holística para garantir a observabilidade [...]" (E28). Portanto, é necessário para permitir a colaboração e a comunicação entre diferentes camadas do sistema e equipes. Descobrimos que é necessária uma visão comum e central que auxilie a implementação de um diagnóstico e detecção de falhas em todo o sistema. Uma solução indicada pelos entrevistados é um sistema de gerenciamento de eventos (S1), também conhecido como 'gerente de gerentes', que permite uma visão geral do estado do sistema. Isso permite correlacionar eventos para redução de eventos. Outras soluções mencionadas são gerenciadores de topologia e modelagem de descoberta de arquitetura (S2). Eles permitem mapeie dinamicamente as transações para os componentes de infraestrutura subjacentes. Além disso, o rastreamento distribuído (S3) foi enfatizado como solução, que registra o caminho de execução de uma solicitação em tempo de execução, propagando IDs de solicitação [20]. Isso a solução permite capturar a relação causal entre eventos na execução caminho. Permite criar uma "[...] visão aérea, para descobrir o que está acontecendo com

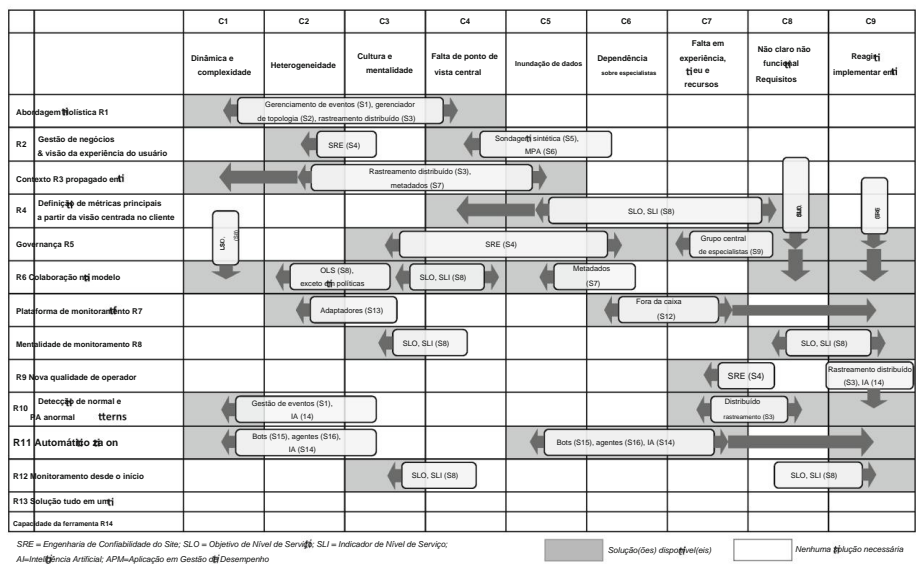


Fig. 2. Visão geral dos desafios, requisitos e soluções

uma solicitação do usuário [...]” (E28). O rastreamento distribuído também pode ser aplicado para diagnóstico de componentes de terceiros onde o código fonte não está disponível. Portanto, os participantes mencionaram que a instrumentação para propagar IDs de rastreamento através de serviços individuais, desenvolvidos por diferentes equipes, não está no momento consistentemente assegurado.

Gestão sob a ótica do negócio e da experiência do usuário (R2): Vários os participantes descreveram a tendência de passar do monitoramento isolado de indivíduos serviços para uma visão dependente do contexto da perspectiva de um cliente ou aplicação empresarial. Além disso, alguns declararam aplicar a abordagem do Google de Site Engenharia de Confiabilidade (SRE) (S4) (E28): “Com nossa abordagem SRE em mente, nos importamos com a experiência do usuário e esses são os caminhos de ouro que queremos melhorar. Não me importo necessariamente com o que está acontecendo por baixo, assim que o usuário não está apresentando erros, latência ou indicadores improváveis”. SRE [3] leva aspectos da engenharia de software e os aplica à infraestrutura e às operações problemas com foco na experiência do cliente. Além disso, vários entrevistados declarou realizar sondagem sintética. Isso é conhecido como monitoramento de ponta a ponta (S5) que permite a emulação do comportamento real do usuário para medir e comparar seu disponibilidade e desempenho. Em geral, muitos participantes referiram-se a aplicar Application Performance Management (APM) (S6) que compreende métodos, técnicas e ferramentas para monitorar continuamente o estado de um sistema de um visão centrada na aplicação. O APM permite diagnosticar e resolver especialmente problemas relacionados ao desempenho.

Propagação de contexto (R3): Para fornecer uma abordagem holística e ser capaz de detectar eventos que impactam o cliente, a propagação de contexto é necessária. O sistema propaga o contexto relevante na forma de metadados, como IDs ou tags ao longo do fluxo de execução de uma solicitação por meio de serviços. Junto com o rastreamento distribuído (S3), adicionar metadados (S7) a métricas e logs são outros exemplos. Isso ajuda a localizar a falha subjacente no fluxo de dados gerados pelos sistemas, fornecendo informações contextuais (E28): “Com a propagação de contexto, você pode facilmente apontar para a causa raiz dos problemas. Por exemplo, há alguma latência adicional e você pode ver que essa outra chamada de banco de dados específica está causando a latência adicional. Você pode inspecionar e executar ping automaticamente no momento e componente certos [...]”.

Definição de métricas principais da visão centrada no cliente (R4): Uma maneira de definir sistematicamente métricas para serviços individuais é o conceito de estabelecer métricas de nível de serviço. Alguns participantes aplicam Objetivos de Nível de Serviço (SLO) e Indicadores de Nível de Serviço (SLI) (S8) como parte de uma abordagem SRE. Enquanto os SLOs descrevem objetivos de negócios definindo o tempo de inatividade aceitável de um serviço da perspectiva do usuário, o SLI, por sua vez, permite vincular métricas aos objetivos de negócios. Os entrevistados expressam uma necessidade essencial de uma definição sistemática dessas métricas, mas, ao mesmo tempo, lutam com sua implementação (E28): “Dá muito trabalho descobrir o SLO certo, o que é um processo muito longo.

Nem todo mundo está interessado nisso. [...] É difícil introduzir esse conceito mais tarde. Isso está criando tensão nas equipes, porque elas estão dizendo: “Isso não é o que vocês nos prometeram”. Mas o problema é que, se ninguém realmente formulou qual era a promessa.”

Governança (R5): Para promover os requisitos anteriores, vários participantes descreveram a necessidade de uma governança que defina uma estratégia incluindo funções, responsabilidades, processos e tecnologias para monitoramento e observabilidade. Isso deve compreender formulações claras de um conjunto mínimo de indicadores que devem ser monitorados de cada serviço. Um outro requisito é reivindicar a observabilidade de um serviço como um critério de aceitação para desenvolvimento e operação. Os participantes mencionaram que o desenvolvimento e a aplicação da governança precisam de várias iterações e devem ser continuamente adaptados em termos da estratégia da empresa. Especialmente para serviços executados na nuvem, as diretrizes devem ser definidas porque “A nuvem é padronização” (E16). Ao mesmo tempo, alguns participantes criticaram a introdução de padrões de ferramentas e uma desaceleração do desenvolvimento por regulamentações de governança superdimensionadas. Algumas empresas já têm um departamento próprio e um grupo de especialistas com a responsabilidade central do monitoramento (S9).

Para fornecer uma estratégia, os participantes mencionaram alinhar sua governança aos princípios e diretrizes do SRE (S4), que em alguns exemplos já evoluíram para um sistema autorregulador. Uma questão comumente mencionada aborda a criação de uma comunidade de uma prática (S10) para compartilhar boas práticas e lições aprendidas.

Modelo de colaboração (R6): Alguns participantes descreveram um modelo de colaboração como uma base essencial para comunicação e processos de diagnóstico eficientes entre diferentes equipes. Como parte disso, políticas de exceção e taxonomias para

anomalias (S11) precisam ser definidas. Para trabalhar eficientemente em conjunto durante o diagnóstico, metadados (S7), capturar as relações causais e fornecer contexto são necessários. Além disso, alguns participantes declararam que uma (E28): “ [...] linguagem comum, chamada SLO e SLI” (S8) é a base para sua colaboração em equipe.

Plataforma de monitoramento (R7): Vários entrevistados exigiram uma plataforma de monitoramento unificada para aumentar a eficiência operacional. Isso inclui componentes prontos para uso (S12) e padronizados que podem ser usados modularmente e são personalizáveis para necessidades específicas dos serviços individuais. O monitoramento e sua configuração padrão também precisam permitir a “democratização de dados” (E19), por exemplo, para oferecer uma API padrão para implantar adaptadores (S13) para diferentes tecnologias. Esse padrão pode, portanto, “ [...] criar um tipo de governança que não é estrita.” (E20).

Mentalidade de monitoramento (R8): O pré-requisito é aumentar a importância da observabilidade e do monitoramento de sistemas distribuídos. Sem uma conscientização crescente, soluções ad-hoc isoladas permanecerão, o que não permite o fornecimento de serviços e diagnósticos suficientes. Uma solução mencionada para aumentar a importância de atributos não funcionais, como disponibilidade e desempenho, pode ser alcançada definindo e controlando SLOs (S8). Dessa forma, a estabilidade e o desenvolvimento de recursos podem ser controlados da perspectiva de gerenciamento. Além disso, os participantes delinearam a necessidade de equalizar os requisitos funcionais e não funcionais.

Nova qualidade do operador (R9): Muitos participantes mencionaram uma falta de educação acadêmica dos operadores. Consequentemente, os operadores precisam aumentar suas habilidades, especialmente de serem capazes de lidar com tarefas de automação. Para atingir isso, a empresa precisa aumentar a conscientização sobre monitoramento e promover operadores responsáveis adequados. Alguns participantes destacaram a necessidade de Engenheiros de Confiabilidade do Site (SRE) (S4), que sejam capazes de trabalhar em tarefas de operação e infraestrutura, bem como em aspectos de engenharia de software.

Deteção de padrões normais e anormais (R10): Quase todos os entrevistados falaram sobre a detecção de anomalias como uma tarefa importante para o monitoramento para diferenciar entre o que é comportamento normal e anormal. Diferentes soluções são mencionadas, como gerenciamento de eventos (S1), para correlacionar eventos de diferentes partes do sistema. Para correlacionar eventos, análises preditivas e inteligência artificial (IA) (S14) estão em uso. Alguns participantes discutiram o problema de diferenciar o comportamento normal de um serviço. Nesse contexto, alguns participantes consideraram o rastreamento distribuído (S3) para indicar medidas de desempenho e desenvolver iterativamente garantias para seus serviços definindo níveis de serviço. O método mais avançado para detecção de anomalias é a IA (S14). Quase todos os participantes apreciaram seu enorme potencial para dominar a complexidade e a enxurrada de dados gerados por sistemas distribuídos. No entanto, muitos entrevistados apontaram que ainda faltam pré-condições suficientes para o uso da IA na prática. Primeiramente, os dados corretos devem ser coletados, a qualidade dos dados deve ser garantida, o contexto precisa ser propagado e os dados devem ser armazenados centralmente. Permanecem preocupações em termos da relação custo-benefício das abordagens de IA e sua confiabilidade (E28): “Não sei se algum dia teremos uma solução na qual possamos confiar com confiança”.

Automação (R11): A dinâmica e a complexidade crescentes dentro de sistemas distribuídos, causadas pela futura arquitetura de microsserviços e pelos ciclos de vida mais curtos dos componentes, não são mais controláveis manualmente como um todo.

Portanto, a automação é indispensável para observar e monitorar um sistema distribuído. Especialmente problemas recorrentes podem ser resolvidos automaticamente e técnicas básicas de monitoramento podem ser implementadas automaticamente. Bots (S15) e agentes (S16) podem realizar automação, pois agem por si mesmos. Em combinação com IA (S14), bots e agentes podem ser mais eficientes e mais precisos em suas tarefas, como coleta e análise de dados de tráfego.

Monitoramento desde o início (R12): O monitoramento é um pré-requisito para qualquer desenvolvimento e operação. Muitos entrevistados indicaram considerar o monitoramento desde o início. Deve ser parte de qualquer design. **Alguns participantes citam a integração de SLOs e SLIs (S8) "[...] no tempo de design. Assim que há um novo serviço, você tem uma seção no documento de design, onde você pode ver que essas são as promessas, elas podem mudar ao longo do tempo para reduzir o trabalho. Começamos a conversa muito cedo."** (E28). Isso pode aumentar a conscientização sobre o monitoramento e pode mudar a cultura da empresa para uma mentalidade de monitoramento. Embora os departamentos de TI devam ver o monitoramento como uma parte integrada, a gerência precisa ser o principal impulsionador para implementar essa mentalidade.

**Solução tudo-em-um (R13):** Soluções que abrangem todas as funcionalidades de monitoramento em uma solução foram mencionadas nas entrevistas. No entanto, a realidade mostra que tais soluções não existem. No melhor dos casos, o mercado oferece soluções que fornecem funcionalidades básicas para monitoramento, como medição de desempenho ou registro. Mais detalhadamente, elas geralmente fornecem a capacidade de expandir facilmente as soluções, por exemplo, combinando e integrando outras soluções de software.

Isso também pode compreender novos padrões, tecnologias e outras soluções já existentes. Portanto, soluções all-in-one representam, neste contexto, uma combinação de várias soluções e tecnologias. No entanto, para concretizar uma solução tão abrangente, é preciso evitar ou substituir soluções isoladas por software de monitoramento padrão, padrões abertos e tecnologias modernas para atingir uma solução abrangente no futuro.

Capacidades da ferramenta (R14): Neste parágrafo, resumimos diferentes capacidades da ferramenta, principalmente requisitos não funcionais, mencionados pelos participantes.

Um requisito frequentemente declarado é o monitoramento em tempo real, onde mudanças e impactos estão sendo monitorados diretamente sem demora. Portanto, as informações necessárias podem ser fornecidas para uma resposta apropriada (por exemplo, alertas em tempo real para reduzir o tempo de reação). Um outro requisito aborda o uso de padrões abertos (por exemplo, JSON ou funcionalidades de monitoramento padrão), que é motivado por ser adaptável e flexível devido a novas tecnologias e padrões. Isso também promove a manutenção e portabilidade de soluções de monitoramento por serem facilmente transferíveis para outros sistemas distribuídos. Associado a isso, a escalabilidade é de particular importância para lidar com sistemas distribuídos grandes e dinâmicos. Embora o gerenciamento da dinâmica dentro de um sistema distribuído precise ser abordado, a confiabilidade e a disponibilidade do monitoramento são altamente exigidas. Por exemplo,

funcionalidades de saúde, como o status atual do sistema, precisam estar disponíveis o tempo todo. Além disso, as ferramentas precisam dar suporte ao gerenciamento multi-tenant. Este requisito aborda especificamente a capacidade de visualizações específicas de locatários e gerenciamento de permissões individuais. Um aspecto adicional mencionado é a importância da segurança das próprias ferramentas de monitoramento. Quanto mais agentes forem usados e quanto maior for a profundidade da integração, mais "backdoors" podem ser abertas e maior será o possível impacto negativo. Portanto, aspectos de segurança, como ações de prevenção, precisam ser realizados. Uma abordagem minimamente invasiva precisa ser seguida, onde as mudanças em um sistema existente são limitadas. Isso pode estar abrindo apenas um mínimo de portas relevantes. Além disso, uma implantação e configuração descuidadas de agentes de monitoramento foram mencionadas como problemas potenciais que podem causar instabilidade e uma carga de rede crescente.

## 5 Ameaças à Validade

Para validade interna, há um risco de que os participantes não tenham declarado a verdadeira situação ou sua opinião. No entanto, esse risco é bem pequeno, porque estávamos garantindo o anonimato das entrevistas e os participantes pareciam não estar preocupados em falar sobre aspectos negativos de seu produto ou empresa. Outra ameaça à validade interna são conceitos potencialmente mal compreendidos usados nas perguntas. Portanto, fornecemos explicações adicionais para conceitos importantes. Caso contrário, fizemos perguntas para esclarecer termos usados pelos participantes que poderiam ter um significado específico de domínio ou empresa. Para reduzir o viés do pesquisador e, portanto, aumentar a validade da interpretação, cada transcrição foi revisada por pelo menos um pesquisador adicional. Além disso, nossos participantes tiveram a chance (e aproveitaram) de ajustar declarações em sua transcrição que estavam incorretas, indistintas ou continham dados confidenciais.

Para aumentar a validade externa, pedimos aos participantes não exclusivamente baseados na Alemanha, mas também participantes vindos de empresas internacionais com diversidade em termos de domínio e tamanho. Além disso, com nossos participantes, estamos cobrindo diferentes funções, vindas de diferentes camadas da pilha de aplicativos, bem como incluindo provedores de soluções de monitoramento e consultores aconselhando empresas e equipes na integração de soluções de monitoramento. Portanto, foi possível gerar uma visão geral das relações complexas em termos de aspectos técnicos e organizacionais que levam a desafios, bem como requisitos e soluções. Ainda assim, como realizamos uma pesquisa qualitativa, não afirmamos que nossos resultados sejam generalizáveis.

## 6 Conclusão

Nosso objetivo de pesquisa foi explorar desafios, requisitos e boas práticas contemporâneas, bem como soluções em termos de monitoramento e observabilidade de sistemas distribuídos. Portanto, conduzimos entrevistas com 28 profissionais de software de 16 organizações. Identificamos que o monitoramento e a observabilidade de sistemas distribuídos não são mais uma questão puramente técnica, mas se tornam

um tópico mais transversal e estratégico, crítico para o sucesso de uma empresa que oferece serviços. Os paradigmas de desenvolvimento e implantação de microsserviços, DevOps e nuvem estão criando máxima independência e especialização, resultando em soluções isoladas de monitoramento e observabilidade, não permitindo gerenciar um serviço de uma visão centrada no cliente ou no negócio. A maioria das empresas já tem soluções e boas práticas em vigor, mas em muitos casos elas permanecem abordagens isoladas devido a estruturas empresariais isoladas. Com referência às descobertas do estado contemporâneo da prática, vemos a necessidade de mais trabalho em boas práticas e exemplos do mundo real para alinhar objetivos de negócios com métricas técnicas para quebrar silos e permitir desenvolvimento e operação eficientes. Além disso, os pesquisadores podem levar esses resultados em consideração para projetar métodos focados na indústria.

## Referências

1. Aceto, G., Botta, A., de Donato, W., Pescap`e, A.: Monitoramento de nuvens: uma pesquisa. *Comput. Rede* 57(9), 2093–2115 (2013)
2. Alhamazani, K., et al.: Uma visão geral das ferramentas comerciais de monitoramento de nuvem: dimensões de pesquisa, questões de design e estado da arte. *Computing* 97(4), 357–377 (2015)
3. Beyer, B., Jones, C., Petoff, J., Murphy, NR: Engenharia de confiabilidade do site: como o Google executa sistemas de produção. O'Reilly Media Inc., Sebastopol (2016)
4. Colville, RJ: CMDDB ou banco de dados de configuração: saiba a diferença (2006)
5. Fatema, K., Emeakaroha, VC, Healy, PD, Morrison, JP, Lynn, T.: Uma pesquisa sobre ferramentas de monitoramento de nuvem: taxonomia, capacidades e objetivos. *J. Parallel Distrib. Comput.* 74(10), 2918–2933 (2014)
6. Gamez-Diaz, A., Fernandez, P., Ruiz-Cortes, A.: Uma análise das ofertas de APIs RESTful na indústria. Em: Maximilien, M., Vallecillo, A., Wang, J., Oriol, M. (eds.) *ICSOC 2017. LNCS*, vol. 10601, pp. 589–604. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-69035-3\\_43](https://doi.org/10.1007/978-3-319-69035-3_43)
7. Gopal, M.: *Modern Control System Theory*, 2ª ed. Halsted Press, Nova York (1993)
8. Gupta, M., Mandal, A., Dasgupta, G., Serebrenik, A.: Monitoramento de tempo de execução em implantação contínua por meio do modelo de comportamento de execução diferencial. Em: Pahl, C., Vukovic, M., Yin, J., Yu, Q. (eds.) *ICSOC 2018. LNCS*, vol. 11236, pp. 812–827. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-03596-9\\_58](https://doi.org/10.1007/978-3-030-03596-9_58)
9. Heger, C., van Hoorn, A., Mann, M., Okanovic, D.: Application performance management: state of the art and challenges for the future. Em: *Anais da 8ª Conferência Internacional ACM/SPEC sobre Engenharia de Desempenho (ICPE 2017)*. ACM (2017)
10. IEEE: Glossário Padrão IEEE de Terminologia de Engenharia de Software (1990). <https://ieeexplore.ieee.org/document/159342>
11. Johng, H., Kim, D., Hill, T., Chung, L.: Estimando o desempenho de sistemas baseados em nuvem usando benchmarking e simulação de forma complementar. Em: Pahl, C., Vukovic, M., Yin, J., Yu, Q. (eds.) *ICSOC 2018. LNCS*, vol. 11236, pp. 576–591. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-03596-9\\_42](https://doi.org/10.1007/978-3-030-03596-9_42)



12. Kinsella, J.: A lacuna de complexidade da nuvem: tornando o software mais inteligente para lidar com infraestrutura complexa. <https://www.cloudcomputing-news.net/news/2015/jun/17/cloud-complexity-gap-making-software-more-intelligent-address-complex-infrastructure/> 13. Knoche, H., Hasselbring, W.: Drivers e barreiras para adoção de microsserviços – uma pesquisa entre profissionais na Alemanha. *Emp. Model. Inf. Syst. Architect. (EMISAJ)–Int. J. Modelo Conceitual.* 14(1), 1–35 (2019)
14. Lin, J., Chen, P., Zheng, Z.: Microscópio: identifique problemas de desempenho com gráficos causais em ambientes de microsserviços. Em: Pahl, C., Vukovic, M., Yin, J., Yu, Q. (eds.) *ICSOC 2018. LNCS*, vol. 11236, pp. 3–20. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-03596-9\\_1](https://doi.org/10.1007/978-3-030-03596-9_1)
15. Mayring, P.: *Análise de conteúdo qualitativa: fundamento teórico, procedimentos básicos e solução de software* (2014)
16. Natsu, M., Ghosh, RK, Shyamsundar, RK, Ranjan, R.: Monitoramento holístico de desempenho de nuvens híbridas: complexidades e direções futuras. *IEEE Cloud Comput.* 3(1), 72–81 (2016)
17. Niedermaier, S., Koetter, F., Freymann, A., Wagner, S.: Diretriz de entrevista sobre observabilidade e monitoramento de sistemas distribuídos (2019). <https://doi.org/10.5281/zenodo.3346579>
18. Picoreti, R., Pereira do Carmo, A., Mendonça de Queiroz, F., Salles Garcia, A., Frizera Vassallo, R., Simeonidou, D.: Observabilidade multinível na orquestração de nuvens. Em: 2018 IEEE 16th International Conference on DASC/PiCom/DataCom/ CyberSciTech, pp. 776–784, agosto de 2018
19. Runeson, P., Höst, M.: Diretrizes para conduzir e relatar pesquisa de estudo de caso em engenharia de software. *Empirical Softw. Eng.* 14(2), 131 (2008)
20. Sambasivan, RR, Shafer, I., Mace, J., Sigelman, BH, Fonseca, R., Ganger, GR: Rastreamento centrado em fluxo de trabalho de sistemas distribuídos. Em: *Anais do Sétimo Simpósio da ACM sobre Computação em Nuvem*, pp. 401–414. ACM (2016)
21. Sfondrini, N., Motta, G., Longo, A.: Adoção de nuvem pública em empresas multinacionais: uma pesquisa. Em: 2018 IEEE International Conference on Services Computing (SCC), pp. 177–184, julho de 2018
22. Singer, J., Sim, SE, Lethbridge, TC: Coleta de dados de engenharia de software para estudos de campo. Em: Shull, F., Singer, J., Sjøberg, DIK (eds.) *Guia para engenharia avançada de software empírico*, pp. Springer, Londres (2008). [https://doi.org/10.1007/978-1-84800-044-5\\_1](https://doi.org/10.1007/978-1-84800-044-5_1)
23. Sun, C., Li, M., Jia, J., Han, J.: Testes orientados a modelos baseados em restrições de serviços da web para conformidade de comportamento. Em: Pahl, C., Vukovic, M., Yin, J., Yu, Q. (eds.) *ICSOC 2018. LNCS*, vol. 11236, pp. 543–559. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-03596-9\\_40](https://doi.org/10.1007/978-3-030-03596-9_40)
24. Yang, Y., Wang, L., Gu, J., Li, Y.: Capturando transparentemente o caminho de execução do processamento de solicitação de serviço/trabalho. Em: Pahl, C., Vukovic, M., Yin, J., Yu, Q. (eds.) *Português ICSOC 2018. LNCS*, vol. 11236, pp. 879–887. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-03596-9\\_63](https://doi.org/10.1007/978-3-030-03596-9_63)