



Universidade Federal do Piauí
Centro de Ciências da Natureza
Programa de Pós-Graduação em Ciência da Computação

Monitoramento de QoS em Múltiplas Nuvens da Perspectiva de um Arquiteto de Software

Lucas de Sousa Nunes

Número de Ordem PPGCC: M001

Picos-PI, Novembro de 2025

Lucas de Sousa Nunes

Monitoramento de QoS em Múltiplas Nuvens da Perspectiva de um Arquiteto de Software

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da UFPI (área de concentração: Sistemas de Computação), como parte dos requisitos necessários para a obtenção do Título de Mestre em Ciência da Computação.

Universidade Federal do Piauí – UFPI

Centro de Ciências da Natureza

Programa de Pós-Graduação em Ciência da Computação

Orientadora: Profa. Dra. Juliana Oliveira de Carvalho

Picos-PI

Novembro de 2025

FICHA CATALOGRÁFICA
Universidade Federal do Piauí
Sistema de Bibliotecas UFPI - SIBi/UFPI
Biblioteca Setorial do CCN

N972m Nunes, Lucas de Sousa.
Monitoramento de QoS de múltiplas nuvens da perspectiva
de um arquiteto de software / Lucas de Sousa Nunes. -- 2025.
59 f. : il. color.

Dissertação (Mestrado) - Universidade Federal do Piauí.
Centro de Ciências da Natureza. Programa de Pós-Graduação
em Ciências da Computação, Teresina, 2025.

“Orientadora: Profa. Dra. Juliana Oliveira de Carvalho”.

1. Sistemas Operacionais. 2. Computação em Nuvem. 3.
Multi-Cloud. I. Carvalho, Juliana Oliveira de. II. Título.

CDD 005.43


Bibliotecária: Caryne Maria da Silva Gomes - CRB3/1461

Lucas de Sousa Nunes


Monitoramento de QoS em Múltiplas Nuvens da Perspectiva de um Arquiteto de Software

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da UFPI (área de concentração: Sistemas de Computação), como parte dos requisitos necessários para a obtenção do Título de Mestre em Ciência da Computação.


Trabalho aprovado. Picos-PI, Novembro de 2025:

Documento assinado digitalmente
 JULIANA OLIVEIRA DE CARVALHO
Data: 11/11/2025 09:46:57-0300
Verifique em <https://validar.iti.gov.br>


Profa. Dra. Juliana Oliveira de Carvalho
Universidade Federal do Piauí (UFPI)

Documento assinado digitalmente
 FRANCISCO AIRTON PEREIRA DA SILVA
Data: 11/11/2025 10:11:30-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Francisco Airtton Pereira da Silva
Universidade Federal do Piauí (UFPI)

Documento assinado digitalmente
 GLAUBER DIAS GONCALVES
Data: 11/11/2025 18:13:40-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Glauber Dias Gonçalves
Universidade Federal do Piauí (UFPI)

Documento assinado digitalmente
 FERNANDO ANTONIO MOTA TRINTA
Data: 12/11/2025 11:28:36-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Fernando Antônio Mota Trinta
Universidade Federal do Ceará (UFC)

Picos-PI
Novembro de 2025

*Em especial aos meus pais, Antonio Sousa e Rosimar Nunes,
por sempre estarem comigo em todos os momentos.*

Agradecimentos

Em um primeiro momento, agradeço a Deus por ter me dado saúde, força, fé e perseverança para não desistir, mesmo diante de vários obstáculos. O curso de mestrado não foi fácil para mim; diferente da graduação, tive que conciliar trabalho, estudos e projetos pessoais.

Foi uma caminhada difícil, mas que muito me preparou para novos desafios e oportunidades pessoais e profissionais. Hoje me sinto muito feliz com esta realização, em poder dar esse orgulho aos meus pais.

Assim, deixo meu abraço e minha profunda gratidão aos meus pais, Rosimar Nunes e Antônio Alves, e aos meus avós, Alsira Maria, Francisco Alves e Luis Martins — estes dois últimos, que infelizmente não estão mais presentes fisicamente, mas que permanecem vivos em minhas lembranças e conquistas. Agradeço por todo o apoio, pelo amor e por serem a base sólida que me permitiu me tornar a pessoa que sou hoje.

Estendo também meu reconhecimento e carinho a todos os meus irmãos — Andressa Sousa, Letícia Sousa, Vanessa Sousa, Felipe Sousa, Romário Sousa e Maurício Nunes — pelo incentivo constante e pelas palavras de motivação ao longo dessa jornada.

À minha namorada Laiane Laurinda pelo apoio, companheirismo e entendimento nas vezes que precisei ficar ausente.

À minha orientadora Profa. Dra. Juliana Oliveira pela orientação, confiança e paciência durante toda essa jornada de pesquisas.

Ao professor e amigo Rayner Gomes pelo incentivo, pela parceria e pelos bons momentos de descontração no ciclismo.

Também agradeço a todos os demais professores do programa de pós-graduação em Ciência da Computação da UFPI. Todos foram fundamentais para o meu aprendizado, como também foram fontes de inspiração e valores que com certeza me moldaram para melhor.

*“A persistência é o menor caminho do êxito.”
(Charles Chaplin)*

Resumo

A computação em nuvem disponibiliza serviços computacionais sob demanda por meio da Internet, com um modelo de pagamento baseado no uso, o que favorece a otimização dos custos operacionais. Essa abordagem proporciona benefícios como acesso seguro a partir de qualquer local e dispositivo, além de facilitar o compartilhamento de informações. Nesse contexto, os usuários (arquitetos de *software*) podem definir requisitos de QoS para os serviços contratados, os quais devem constar do SLA. Como resultado, a computação em nuvem tem se consolidado como uma solução atrativa para empresas de diversos portes, o que vem impulsionando seu crescimento nos últimos anos. Para maximizar os benefícios da computação em nuvem, muitos usuários têm adotado o modelo *multi-cloud*, no qual uma aplicação é distribuída em vários provedores, geralmente sem um acordo prévio de integração entre eles. No entanto, a maioria dos provedores não disponibiliza diretamente as métricas dos serviços contratados, o que dificulta a verificação do cumprimento dos SLAs e a avaliação da conformidade com os parâmetros de QoS estabelecidos. Diante disso, os usuários precisam recorrer a estratégias alternativas para aferir os valores de QoS dos serviços contratados. Essa tarefa torna-se ainda mais complexa em ambientes *multi-cloud*, em que múltiplos SLAs devem ser considerados. Nesses casos, é necessário que o usuário avalie as métricas de desempenho tanto sob a perspectiva de cada provedor quanto sob a perspectiva da aplicação como um todo. Diante deste contexto, torna-se importante o monitoramento do QoS a partir da ótica do usuário, com base no SLA. O monitoramento permite não só avaliar a qualidade da oferta, mas também identificar se a aplicação demanda mais recursos. Para isso, apresentamos primeiro uma taxonomia que apoia a análise comparativa das soluções existentes para o monitoramento de nuvem. Em seguida, com base nas lacunas identificadas, propomos uma arquitetura para o monitoramento do QoS de serviços contratados em múltiplos provedores de nuvem, para microsserviços distribuídos de uma aplicação. Para validar a proposta, desenvolvemos o Cloud Sentinel, uma ferramenta capaz de monitorar métricas ao longo do tempo e apoiar a tomada de decisões, permitindo aferir o QoS de cada serviço utilizado por cada microsserviço, em diferentes provedores contratados.

Palavras-chaves: multi-cloud. monitoramento. SLA. QoS. arquiteto de software.

Abstract

Cloud computing offers on-demand computing services over the Internet, utilizing a pay-as-you-go model that facilitates the optimization of operational costs. This approach provides benefits such as secure access from any location and device, as well as facilitating information sharing. In this context, users (software architects) can define QoS requirements for the contracted services, which must be included in the SLA. As a result, cloud computing has become an attractive solution for companies of all sizes, driving its growth in recent years. To maximize the benefits of cloud computing, many users have adopted the multi-cloud model, in which an application is distributed across multiple providers, usually without a prior integration agreement between them. However, most providers do not directly provide metrics for the contracted services, which makes it difficult to verify compliance with SLAs and assess conformity with established QoS parameters. Given this, users must resort to alternative strategies to measure the QoS values of contracted services. This task becomes even more complex in multi-cloud environments, where multiple SLAs must be considered. In these cases, the user must evaluate performance metrics from both the provider's perspective and the application's overall perspective. In this context, monitoring QoS from the user's perspective, based on the SLA, becomes essential. Monitoring allows not only the evaluation of the quality of the offer, but also the identification of whether the application demands more resources. To this end, we first present a taxonomy that supports the comparative analysis of existing cloud monitoring solutions. Then, based on the identified gaps, we propose an architecture for monitoring the QoS of contracted services from multiple cloud providers for the distributed microservices of an application. To validate the proposal, we developed Cloud Sentinel, a tool capable of monitoring metrics over time and supporting decision-making, which enables us to measure the QoS of each service used by each microservice across different contracted providers.

Keywords: multi-cloud. monitoring. SLA. QoS. software architect.

Lista de ilustrações

Figura 1 – Metodologia para o desenvolvimento deste projeto.	4
Figura 2 – Relações entre os termos que descrevem múltiplas nuvens, com base em Petcu (2014).	8
Figura 3 – Cenário comparativo entre nuvem única e múltiplas nuvens.	9
Figura 4 – Cenário comparativo entre arquitetura monolítica e arquitetura de microsserviços.	10
Figura 5 – Taxonomia proposta para o monitoramento em múltiplas nuvens. . . .	17
Figura 6 – Fluxo de revisão de soluções de monitoramento de múltiplas nuvens. .	24
Figura 7 – Visão geral dos resultados para o modelo de entrega adotado pelas soluções.	27
Figura 8 – Visão geral dos resultados sobre as perspectivas de monitoramento. . .	28
Figura 9 – Resumo dos resultados sobre o modelo de serviço.	28
Figura 10 – Resultados da investigação sobre a granularidade do monitoramento. .	29
Figura 11 – Resultados da investigação sobre o estado dos requisitos de monitoramento considerados pelas soluções.	30
Figura 12 – Arquitetura proposta para monitoramento de aplicações em <i>multi-cloud</i> . .	36
Figura 13 – Solução de monitoramento.	38
Figura 14 – Arquitetura da aplicação para Monitoramento de Múltiplas Nuvens. . .	41
Figura 15 – Ilustração da configuração experimental para validação da solução de monitoramento em multi-cloud.	46
Figura 16 – Ilustração da funcionalidade de definição de SLA para métricas de monitoramento.	46
Figura 17 – Ilustração da página de dependências coletadas pelo Cloud Sentinel. . .	49
Figura 18 – Classes de serviços de acordo com o Cloud Sentinel.	50
Figura 19 – Relatório de serviços conforme o Cloud Sentinel.	51
Figura 20 – Métricas monitoradas no Cenário 01 do Redis.	51
Figura 21 – Classes de serviços para o cenário 02 de acordo com o Cloud Sentinel. .	52
Figura 22 – Relatório para o cenário 02 de acordo com o Cloud Sentinel.	53
Figura 23 – Métricas monitoradas no Cenário 02 para o <i>PostgreSQL</i>	53
Figura 24 – Comparativo de desempenho de serviços em ambiente distribuído e não distribuído.	55

Lista de tabelas

Tabela 1 – Strings de busca para encontrar trabalhos sobre monitoramento em múltiplas nuvens (surveys).	14
Tabela 2 – Comparativo entre trabalhos que tratam do monitoramento de serviços em um único provedor de nuvem.	15
Tabela 3 – Resumo dos artigos por país, local de publicação e ano.	25
Tabela 4 – Visão geral dos trabalhos relacionados e suas propostas principais. . . .	26
Tabela 5 – Resumo da quantidade de pesquisas e flexibilidade dos requisitos considerados no monitoramento.	31
Tabela 6 – Configurações dos serviços para realização de experimentos.	47
Tabela 7 – Cenários de teste realizados.	50

Lista de abreviaturas e siglas

AWS	Amazon Web Services
API	Application Programming Interface (Interface de Programação de Aplicações)
CPU	Central Processing Unit (Unidade Central de Processamento)
DBaaS	Database as a Service (Banco de Dados como Serviço)
EMMCS	Edge Monitoring Framework for Multi-Cloud Systems
HTTP	HyperText Transfer Protocol (Protocolo de Transferência de Hipertexto)
IaaS	Infrastructure as a Service (Infraestrutura como Serviço)
MMC	Monitoramento em Multi-Cloud
PaaS	Platform as a Service (Plataforma como Serviço)
SaaS	Software as a Service (Software como Serviço)
SLA	Service Level Agreement (Acordo de Nível de Serviço)
SLAaaS	Service Level Agreement as a Service (Acordo de Nível de Serviço como Serviço)
SLO	Service Level Objective (Objetivo de Nível de Serviço)
QoE	Quality of Experience (Qualidade da Experiência)
QoS	Quality of Service (Qualidade de Serviço)
REST	Representational State Transfer (Transferência de Estado Representacional)
RTT	Round Trip Time (Tempo de Ida e Volta)
TI	Tecnologia da Informação
VM	Virtual Machine (Máquina Virtual)

Sumário

1	Introdução	1
1.1	Objetivos	3
1.2	Metodologia	3
1.3	Organização	5
2	Referencial Teórico	7
2.1	Computação em Nuvem	7
2.2	Múltiplas Nuvens	8
2.3	Arquitetura de Microsserviços	10
2.4	Acordo de Nível de Serviço	11
2.5	Monitoramento de Serviços	12
3	Taxonomia para Monitoramento em <i>Multi-cloud</i>	13
3.1	Revisão de Literatura para Taxonomia em Nuvem	13
3.2	Taxonomia Proposta	16
3.2.1	Objetivo para um Monitoramento	16
3.2.2	Perspectiva do Monitoramento	18
3.2.3	Tipo de Nuvem	18
3.2.4	Modelo de Entrega	18
3.2.5	Modelo de Serviço	19
3.2.6	Arquitetura da Aplicação	19
3.2.7	Granularidade do Monitoramento	20
3.2.8	Estado dos Requisitos	20
3.2.9	Quantidade de Requisitos	20
4	Trabalhos Relacionados	23
4.1	Questões de Pesquisa	23
4.2	Revisão de literatura sobre soluções de monitoramento	24
4.3	Análises e Discussões	27
5	Monitoramento Cloud Sentinel	35
5.1	Arquitetura Proposta	35
5.2	Coletor de requisitos	37
5.3	Cloud Sentinel	37
5.3.1	Arquitetura do Cloud Sentinel	38
5.3.2	Integração entre o <i>Cloud Sentinel</i> e a Aplicação	40

5.4	Serviços disponíveis	41
5.5	Compliance Check	42
5.6	Sistema de Recomendação	43
6	Experimentos e Resultados	45
6.1	Ferramenta	45
6.2	Experimentos	48
6.3	Resultados para o Cenário 01	50
6.4	Resultados para o Cenário 02	52
6.5	Resultados para o Cenário 03	54
7	Considerações finais e Trabalhos Futuros	57
	Referências	59

1 Introdução

A utilização de serviços de Tecnologia da Informação (TI) fornecidos pela computação em nuvem ganhou popularidade e teve um crescimento acelerado ao longo dos últimos anos. De acordo com o relatório [Flexera \(2025\)](#), em 2025, um terço (33%) das organizações globais está gastando mais de US\$ 12 milhões anuais com nuvem pública, representando um aumento de 4% em relação ao ano anterior (29%). Um crescimento significativo que destaca a dependência da nuvem para operações críticas. Entre as grandes empresas, essa porcentagem sobe para 40%, refletindo uma tendência ainda mais acentuada de migração de cargas de trabalho para a nuvem.

A popularidade da computação em nuvem se deve principalmente às suas características essenciais, como o conjunto de recursos compartilhados, o amplo acesso à rede, o serviço medido, o autoatendimento sob demanda e a rápida elasticidade ([CARVALHO et al., 2018](#)). Na literatura, é comum encontrar vários conceitos para computação em nuvem. Para [Vaquero et al. \(2008\)](#), a computação em nuvem é vista como um grande conjunto de recursos virtualizados, facilmente acessíveis e utilizáveis (como *hardware*, plataformas de desenvolvimento e/ou serviços). Esses recursos podem ser dinamicamente reconfigurados para se ajustar a uma carga variável, permitindo também uma utilização otimizada dos recursos. Esse conjunto de recursos é tipicamente explorado através de um modelo de pagamento por uso, no qual as garantias são oferecidas pelo provedor de infraestrutura através de acordos de nível de serviços (SLA) personalizados. [Fox et al. \(2009\)](#) apresenta um conceito mais amplo e define a computação em nuvem como todos os aplicativos entregues como serviços pela Internet e todos os recursos de *hardware* e *software* dos *datacenters* que oferecem esses serviços.

A adoção de estratégias de múltiplas nuvens, nas quais uma organização utiliza serviços de mais de um provedor simultaneamente, também tem impulsionado o crescimento da computação em nuvem. Essa escolha é motivada por diversos fatores, que vão desde a otimização de custos até a possibilidade de combinar os melhores recursos oferecidos por cada plataforma. O modelo de múltiplas nuvens permite explorar de forma mais eficiente os benefícios de diferentes provedores, como desempenho otimizado, redundância, escalabilidade e flexibilidade na alocação de cargas de trabalho. Ainda de acordo com o relatório de ([Flexera, 2025](#)), cerca de 70% das empresas já adotam múltiplas nuvens. Entretanto, essa estratégia também apresenta novos desafios, incluindo a portabilidade de aplicações e dados entre diferentes provedores, a interoperabilidade entre plataformas heterogêneas e a necessidade de monitoramento e gerenciamento integrado de múltiplos SLAs, assegurando que os requisitos de desempenho e disponibilidade sejam atendidos em

toda a infraestrutura distribuída.

É notório que os avanços na computação em nuvem têm proporcionado inúmeros benefícios aos usuários. Entre eles, destaca-se o acesso a recursos computacionais apresentados como virtualmente ilimitados e a flexibilidade de pagar apenas pelo consumo real dos serviços. Além disso, a computação em nuvem elimina a necessidade de os usuários se preocuparem com a instalação, manutenção e atualização de infraestrutura ou serviços, o que resulta em uma significativa redução dos custos operacionais (CARVALHO et al., 2018). No entanto, mesmo a nuvem suportando o provisionamento elástico de aplicações e trazendo melhorias para os usuários em relação aos custos, devido ao número de ofertas especializadas, ainda existem vários desafios a serem enfrentados.

O monitoramento de serviços sob a perspectiva de um arquiteto de *software* representa um desafio relevante. Embora os provedores de nuvem ofereçam soluções próprias, essas geralmente se concentram em métricas de nível de infraestrutura, negligenciando aspectos mais amplos relacionados ao desempenho e à usabilidade das aplicações. Além disso, tais soluções frequentemente intensificam o problema de *vendor lock-in*, restringindo a portabilidade e a independência dos usuários em relação a um único provedor. Outro ponto crítico é que os provedores de nuvem pública não oferecem visibilidade completa sobre características de desempenho das aplicações e dos recursos de *hardware* alocados, além de apresentarem limitações na exposição de métricas críticas, como as relacionadas à comunicação entre usuário e provedor (TURK et al., 2016). Nesse contexto, o monitoramento eficiente dos serviços em nuvem torna-se essencial para garantir que o QoS esteja alinhado com os termos definidos no SLA. Essa prática não apenas contribui para a melhoria do desempenho das aplicações, mas também otimiza a utilização e a alocação de recursos, além de assegurar uma cobrança justa pelo uso dos serviços (BUYYA; RANJAN; CALHEIROS, 2010; BELOGLAZOV; BUYYA, 2010).

O monitoramento regular desempenha um papel crucial na otimização adaptativa do uso de recursos e na identificação de problemas relacionados aos serviços (FAHAD; AHMED; KAHAR, 2017). Diversos estudos realizaram análises comparativas sobre soluções de monitoramento disponíveis no mercado, dentre eles, Syed et al. (2017), Rodrigues et al. (2016) e Kesavulu et al. (2018). Entre esses, apenas o trabalho de Kesavulu et al. (2018) explorou o monitoramento com foco específico no usuário. Contudo, nenhum dos estudos analisados considerou aspectos fundamentais como a granularidade do monitoramento (por exemplo, aplicação, serviço ou tarefa) e o estado das métricas monitoradas (estático ou dinâmico). Além disso, não foi investigado se as soluções avaliadas permitem o monitoramento de múltiplos requisitos simultaneamente, ou se limitam a um único requisito por vez, tampouco se as métricas podem ser configuradas diretamente pelo usuário.

Dessa forma, observa-se que ainda não existem análises abrangentes sobre as soluções de monitoramento disponíveis atualmente para atender aos requisitos de um

usuário que deseja monitorar múltiplas nuvens. Embora alguns estudos tenham destacado a necessidade de monitoramento do lado do usuário, o desenvolvimento de soluções eficazes nesta área continua sendo importante (KESAVULU et al., 2018). Para avançar nesse campo, é essencial compreender o que já foi abordado e de que maneira, assim como identificar os problemas não resolvidos no monitoramento de serviços de nuvem sob a perspectiva do usuário. Esse entendimento permitirá aos pesquisadores propor novas soluções, visando tanto melhorar os desafios já identificados quanto resolver questões ainda não exploradas. Além disso, possibilitará que os usuários analisem se suas necessidades estão sendo atendidas ou se novas soluções são necessárias.

1.1 Objetivos

Este trabalho tem como objetivo geral propor uma arquitetura para o monitoramento em múltiplas nuvens que permita monitorar métricas de QoS definidas no SLA, sob a perspectiva de um arquiteto de *software*.

Os objetivos específicos deste trabalho são:

- Contribuir com pesquisadores e profissionais da indústria através de um levantamento do estado da arte sobre o monitoramento em múltiplas nuvens;
- Propor uma taxonomia para auxiliar a análise das soluções para monitoramento de nuvem existentes;
- Desenvolver uma solução para o monitoramento de *multicloud* baseado na arquitetura proposta.

1.2 Metodologia

Nesta seção, apresentamos a metodologia para o monitoramento de múltiplas nuvens sob a perspectiva do usuário. A metodologia usada nesse projeto inclui etapas que vão desde a revisão de literatura, proposta de taxonomia, desenvolvimento de soluções, validações e comparações com soluções existentes. A Figura 1 apresenta uma visão geral da metodologia seguida neste trabalho, fornecendo um panorama estruturado de suas etapas principais. Nos tópicos seguintes, detalhamos cada uma dessas etapas.

Revisão de literatura: na etapa inicial da metodologia, foi conduzido um levantamento de literatura com o objetivo de identificar os trabalhos já realizados e mapear os principais lacunas relacionados ao monitoramento em ambientes de múltiplas nuvens. Essa análise também considerou as demandas e necessidades específicas de arquitetos de software, o contexto de aplicações nativas para nuvem e as particularidades da arquitetura

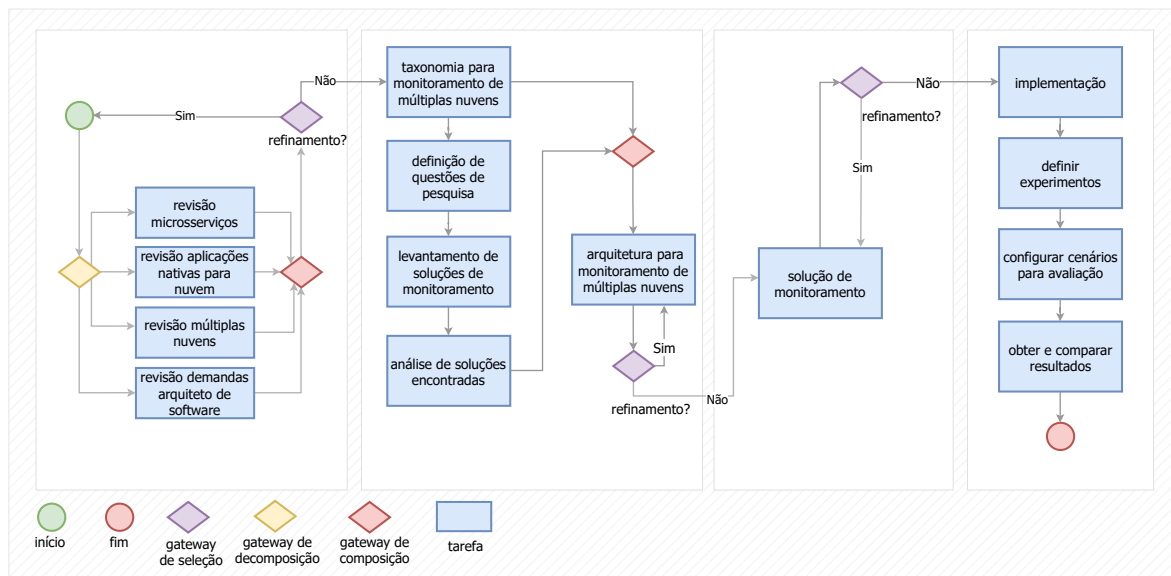


Figura 1 – Metodologia para o desenvolvimento deste projeto.

de microserviços no desenvolvimento de aplicações para serem implantadas em vários provedores;

Taxonomia: após o levantamento do estado da arte sobre o monitoramento de serviços em ambientes de múltiplas nuvens, propomos uma taxonomia com o propósito de orientar e apoiar a análise das diferentes opções de soluções de monitoramento disponíveis no cenário atual. Essa taxonomia também serve como referência para o desenvolvimento de novas abordagens, fornecendo uma estrutura sistemática que facilita a avaliação e a comparação de tecnologias, bem como a identificação de lacunas e oportunidades de melhoria;

Definição de questões de pesquisa: com base na taxonomia proposta, fizemos a definição das questões de pesquisa, as quais foram fundamentadas nos problemas identificados. O objetivo é direcionar a investigação para lacunas específicas do cenário atual, possibilitando a análise crítica das soluções existentes e a comparação com a nossa proposta de solução;

Levantamento de soluções de monitoramento: etapa destinada a levantar as soluções de monitoramento de nuvem mais atuais para serem analisadas neste trabalho;

Análise de soluções de monitoramento: nesta etapa analisamos as soluções de monitoramento existentes na literatura atual tendo como base as questões de pesquisa elaboradas;

Proposta de arquitetura: de acordo com a taxonomia e a análise das soluções da literatura, fizemos a proposta de uma arquitetura para o monitoramento de aplicações distribuídas em múltiplas nuvens da perspectiva do usuário. A arquitetura tem como base aplicações baseadas em microserviços;

Desenvolvimento de solução: para validar a arquitetura proposta, fizemos a formalização de uma solução de monitoramento em múltiplas nuvens capaz de monitorar aplicações distribuídas baseadas em microsserviços;

Experimentos e análise dos resultados: por fim, as 3 últimas etapas da metodologia consistem na implementação, definição e configuração de experimentos destinados a validar a arquitetura proposta. Em seguida, fez-se uma análise crítica dos resultados obtidos e a comparação destes com soluções de monitoramento já existentes, de modo a identificar vantagens, limitações e possíveis melhorias.

1.3 Organização

O restante deste trabalho está organizado como segue: O Capítulo 2 apresenta alguns conceitos sobre as tecnologias que fazem parte do contexto deste trabalho. O Capítulo 3 apresenta a taxonomia proposta para o monitoramento em múltiplas nuvens da visão de um arquiteto de soluções para nuvem. O Capítulo 4 mostra alguns trabalhos relacionados explorando as questões de pesquisa e dispendo das análises por meio de gráficos e tabelas. O Capítulo 5 apresenta a arquitetura proposta para o monitoramento em múltiplas nuvens sobre a visão de um arquiteto de *software*. O Capítulo 6 aborda os resultados alcançados com o desenvolvimento e validações realizadas nesta pesquisa e também discorre sobre os mesmos. Por fim, o Capítulo 7 traz as considerações finais e dá direcionamentos futuros.

2 Referencial Teórico

Este capítulo apresenta um referencial teórico relacionado a conceitos, metodologias e tecnologias importantes para este trabalho. Desta forma, são abordados temas como computação em nuvem, ambientes em múltiplas nuvens, acordos de nível de serviço (SLA) e monitoramento de serviços.

2.1 Computação em Nuvem

A computação em nuvem remete-se à disponibilidade e à capacidade de acessar serviços de computação (incluindo servidores, armazenamento, bancos de dados, rede e *software*) pela *Internet* para oferecer inovações mais rápidas, serviços flexíveis e economias em escala. Na computação em nuvem, paga-se apenas pelos serviços de nuvem usados, o que ajuda a reduzir os custos, a executar a infraestrutura com mais eficiência e a escalar conforme as necessidades dos usuários. Assim, empresas pequenas podem reduzir os empecilhos para adentrar no mercado, permitindo que os custos operacionais sejam proporcionais ao crescimento da receita (CARVALHO, 2019). Diante de todas as vantagens da computação em nuvem, a mesma teve um crescimento enorme nos últimos anos (MEENA, 2021) e, desta forma, pode-se dizer que a computação em nuvem já não é mais uma tendência; ela representa uma grande mudança na forma tradicional de pensamento adotada pelas empresas sobre os recursos de TI.

Conforme apresentado no decorrer do documento, a computação em nuvem proporciona diversas vantagens para seus consumidores, especialmente através dos avanços da virtualização. Com isso, reduz-se a necessidade de as empresas investirem em aspectos como aluguel de espaço físico, infraestrutura de *hardware*, contratação de profissionais qualificados para suporte técnico, atualizações de *software*, entre outros. Além disso, a possibilidade de trabalho remoto também contribui para a redução de gastos com aluguel e energia elétrica.

No entanto, os benefícios da nuvem não se limitam somente à redução de custos, tampouco nas suas características ditas como essenciais. Pois, somado a isso, as organizações estão migrando para a nuvem por influência de aspectos como escalabilidade, alcance, disponibilidade, confiabilidade, capacidade de recuperação a desastres (possibilidade de fazer *backups* e restaurar dados), mobilidade etc. Elas estão usando uma variedade de métricas para medir o valor de negócios resultante da nuvem. As muitas vantagens oferecidas pela computação em nuvem provaram ser especialmente valiosas à medida que as organizações se adaptaram e vêm se adaptando desde o início do ano de 2020 para

atender às necessidades em rápida evolução apresentadas pelo advento de uma maior adoção do trabalho remoto frente aos negócios (LUXNER, 2021).

2.2 Múltiplas Nuvens

O conceito de múltiplas nuvens refere-se ao uso conjunto de serviços provenientes de diferentes provedores de nuvem, públicos ou privados, por uma mesma aplicação. Essa abordagem tem ganhado relevância à medida que as organizações buscam maior flexibilidade, resiliência e otimização de recursos em ambientes computacionais distribuídos.

Em virtude da diversidade de termos utilizados na literatura para descrever arquiteturas que envolvem múltiplos provedores de nuvem, Petcu (2014) propõe uma taxonomia que visa organizar e padronizar esses conceitos. Essa taxonomia, adotada neste trabalho, é ilustrada na Figura 2 e categoriza três modelos principais com suas sucessivas derivações, *multi-cloud*, *cloud federation* e *inter-cloud*. Estes modelos são explorados a seguir:

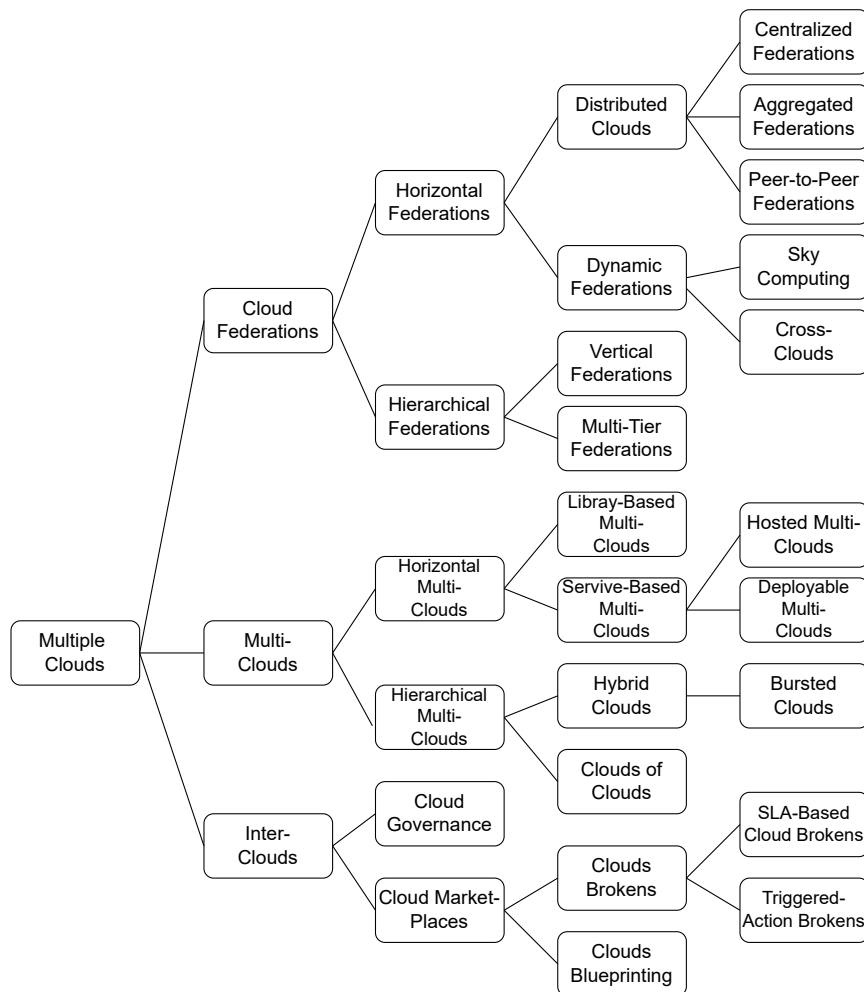


Figura 2 – Relações entre os termos que descrevem múltiplas nuvens, com base em Petcu (2014).

- *Multi-cloud*: modelo em que os serviços são fornecidos por diferentes provedores, sem a existência de acordos formais entre eles, exigindo a intermediação por terceiros para integração e gerenciamento;
- *Cloud-federation*: modelo no qual há colaboração entre provedores de nuvem, estabelecida por meio de acordos formais que permitem o compartilhamento de recursos e serviços;
- *Inter-cloud*: modelo híbrido que combina características dos dois anteriores, mas que demanda maior dinamismo e escalabilidade dos serviços, além da atuação de um *cloud broker*, responsável pelo gerenciamento do uso, desempenho e entrega dos serviços.

A escolha por uma estratégia baseada em nuvem única ou múltiplas nuvens depende das características e objetivos de cada organização. A abordagem de múltiplas nuvens apresenta vantagens significativas, como a redução da dependência de um único fornecedor e o aumento da resiliência e da continuidade dos negócios, ao permitir maior flexibilidade na escolha dos serviços que melhor se ajustem às necessidades da aplicação.

A Figura 3 apresenta um cenário comparativo entre o uso de uma única nuvem e de múltiplas nuvens. Nesse exemplo, a organização demanda serviços de e-mail, computação e memória. Ao optar por um único provedor, está restrita às configurações e soluções por ele oferecidas. Já em um cenário de múltiplas nuvens, o arquiteto tem a liberdade de selecionar, para cada serviço, o provedor que melhor atenda a requisitos específicos, como desempenho, custo ou disponibilidade.

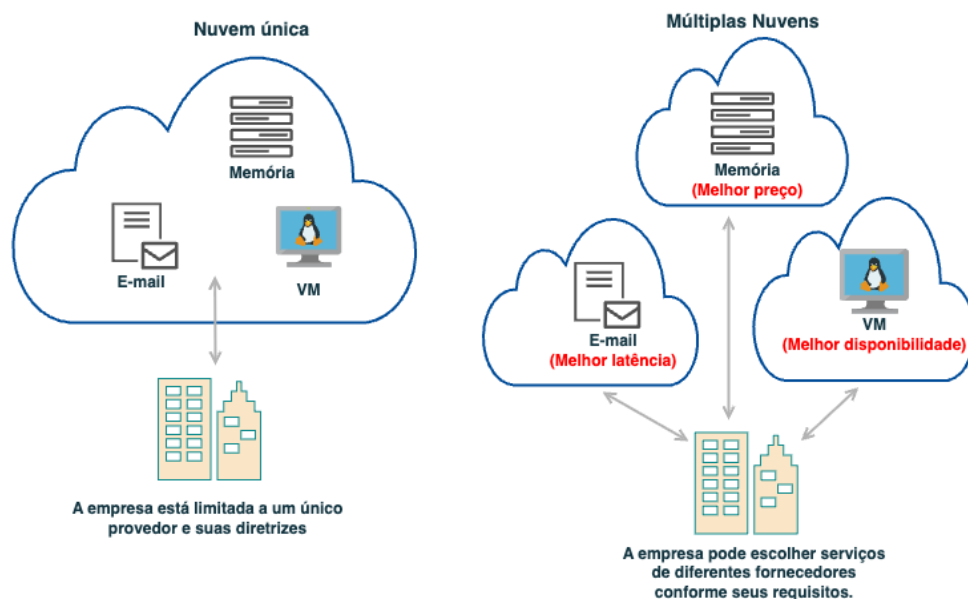


Figura 3 – Cenário comparativo entre nuvem única e múltiplas nuvens.

A adoção da estratégia de várias nuvens, no entanto, impõe desafios consideráveis. Dentre os principais estão interoperabilidade e portabilidade entre diferentes provedores, conforme discutido por [Carvalho et al. \(2018\)](#). Além disso, sua implementação requer maior esforço de planejamento, ferramentas especializadas de gestão e uma equipe técnica qualificada, o que pode representar um obstáculo inicial para muitas organizações.

Por fim, destaca-se que a adoção consciente de uma arquitetura de múltiplas nuvens, fundamentada em taxonomias consolidadas e apoiada por ferramentas adequadas, pode representar um diferencial competitivo relevante ao proporcionar maior controle sobre os recursos computacionais utilizados e maior capacidade de adaptação a diferentes cenários de negócios.

2.3 Arquitetura de Microserviços

Microserviço é um estilo arquitetônico para o desenvolvimento de aplicações, as quais devem ser concebidas e construídas como uma coleção de serviços independentes, fracamente acoplados e com escopo funcional bem definido. Esses serviços podem se comunicar entre si, e cada um é responsável por uma parte específica da lógica de negócio da aplicação. Eles podem ser desenvolvidos, implantados e escalados de forma independente. Isso contrasta com as arquiteturas monolíticas, nas quais as aplicações são construídas como uma única unidade totalmente integrada. A Figura 4 ilustra essa diferença: em uma arquitetura de microserviços, cada serviço possui seu próprio banco de dados e lógica de acesso a dados, enquanto, na arquitetura monolítica, esses elementos estão centralizados no mesmo contexto.

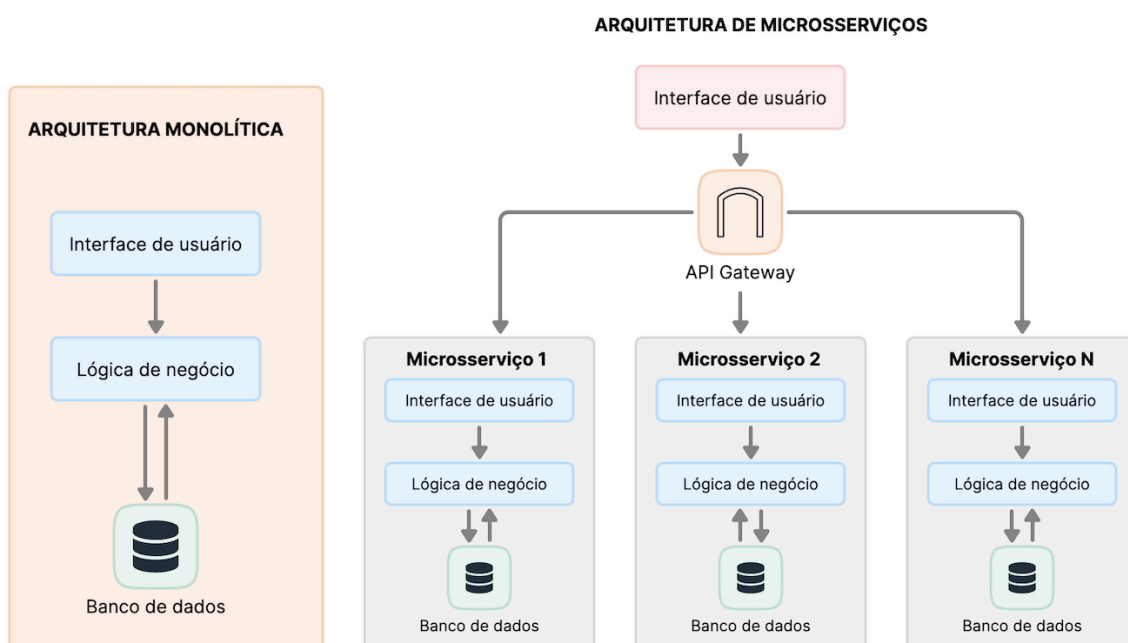


Figura 4 – Cenário comparativo entre arquitetura monolítica e arquitetura de microserviços.

Microserviços e múltiplas nuvens são duas abordagens distintas, mas complementares, no desenvolvimento de *software*. Por sua natureza modular e distribuída, os microserviços permitem que diferentes partes de uma aplicação sejam implantadas de forma independente e em provedores de nuvem distintos. Essa característica favorece a flexibilidade na escolha dos serviços mais adequados para cada componente do sistema, otimizando aspectos como, por exemplo, custo, desempenho, escalabilidade e disponibilidade. Considerando um sistema de *e-commerce* como exemplo, podemos hospedar seu serviço de autenticação em um provedor especializado em segurança, como a *Amazon Web Services* (AWS), e utilizar recursos de computação do *Google Cloud* para o processamento de pedidos e o armazenamento de imagens em um serviço de objeto da *Azure*. Essa distribuição estratégica permite tirar proveito das melhores ofertas de cada provedor. Além disso, a combinação de microserviços e múltiplas nuvens permite que as aplicações sejam flexíveis, escaláveis e resilientes, pois falhas em um serviço ou provedor não comprometem toda a aplicação.

2.4 Acordo de Nível de Serviço

O Acordo de Nível de Serviço (*Service Level Agreement* – SLA) consiste em um acordo contratual entre o fornecedor de serviços e o cliente, com o propósito de assegurar que um nível mínimo de desempenho seja mantido. No contexto da computação em nuvem, o SLA assume papel estratégico, pois define de forma explícita as responsabilidades tanto do provedor quanto do usuário, servindo como base para a gestão da qualidade e confiabilidade dos recursos ofertados.

O SLA formaliza, em termos jurídicos e operacionais, as condições sob as quais os serviços serão prestados. Um SLA tradicional descreve o relacionamento e as atribuições das partes envolvidas, especifica os padrões acordados para a entrega dos serviços — expressos por meio dos Objetivos de Nível de Serviço (*Service Level Objectives* – SLOs) — e estabelece as obrigações e penalidades aplicáveis em casos de descumprimento (HUSSAIN et al., 2017). Além disso, o documento define os parâmetros de QoS aplicáveis aos serviços contratados, permitindo que o cliente monitore o desempenho com base em métricas objetivas, detecte falhas potenciais e identifique violações, assim como possibilita ao provedor validar a conformidade de sua entrega. Ao estabelecer parâmetros claros e mensuráveis, o SLA promove transparência na relação entre provedor e cliente, fortalecendo a confiança mútua. Em caso de não atendimento aos níveis acordados, o SLA também prevê mecanismos compensatórios e sanções, assegurando que a prestação de serviços seja regida por critérios objetivos e verificáveis.

Historicamente, a violação de SLAs já gerou impactos significativos em operações críticas. Por exemplo, em 2016, um ataque massivo de negação de serviço distribuída (DDoS)

direcionado à empresa Dyn, provedora de serviços de DNS, provocou a indisponibilidade de grandes plataformas como *Twitter*, *Netflix* e *Spotify*. Embora o incidente tenha sido motivado por questões de segurança, a consequência imediata foi a quebra dos níveis de disponibilidade acordados, evidenciando a importância de mecanismos robustos de monitoramento e resposta para prevenir, detectar e mitigar situações que possam levar ao descumprimento de SLAs (DN.org, 2025). Esse evento reforça a importância de arquiteturas de monitoramento independentes e tolerantes a falhas, especialmente em cenários *multi-cloud*, nos quais a ausência de redundância adequada pode maximizar o impacto de falhas em um único provedor.

2.5 Monitoramento de Serviços

O monitoramento de serviços de nuvem consiste em monitorar, analisar e gerenciar processos operacionais e o fluxo de trabalho dos serviços de um provedor de nuvem. Com o monitoramento na nuvem, é possível verificar se os serviços estão funcionando corretamente. Ele também permite que as equipes de *DevOps* obtenham percepções críticas sobre o desempenho dos ambientes nativos da nuvem complexos. Por meio do monitoramento, o usuário pode obter dados e informações sobre os serviços dos provedores de nuvem. Tais dados e informações servem como base para a tomada de decisões e para traçar estratégias para, por exemplo, melhorar o desempenho de serviços contratados ou até mesmo contratar serviços de outro provedor de nuvem que melhor atenda às suas necessidades.

Segundo Ciuffoletti (2016), o monitoramento de serviços de nuvem foi fortemente influenciado por um extenso estudo sobre estruturas de monitoramento distribuído. Seguindo um modelo comumente adotado, o provedor é responsável pela implementação da infraestrutura de monitoramento, e o usuário inscreve-se para receber os dados. No entanto, uma interface de monitoramento como serviço deve permitir que o usuário tenha controle sobre a infraestrutura de monitoramento e não só obter os dados disponíveis. Ainda, de acordo com Ciuffoletti (2016), essa reivindicação é endossada por vários trabalhos que não se concentram diretamente nos recursos fornecidos sob demanda.

Nesse contexto, tanto os provedores de nuvem quanto os usuários que utilizam serviços da nuvem devem monitorar constantemente o estado de seus produtos para garantir que os serviços estejam atendendo às dimensões de qualidade: disponibilidade, confiabilidade, desempenho, etc. (POURMAJIDI et al., 2018). Assim, o monitoramento é um componente essencial de uma plataforma em nuvem e desempenha um papel fundamental tanto para provedores quanto para consumidores de nuvem.

3 Taxonomia para Monitoramento em *Multi-cloud*

Uma taxonomia é uma estrutura conceitual destinada à classificação e à organização sistemática de elementos que compartilham características comuns em um domínio de conhecimento específico. Essa estrutura possibilita a análise comparativa, a identificação de padrões e lacunas, bem como o estabelecimento de uma terminologia unificada que favorece o avanço científico e tecnológico da área. No contexto da computação em nuvem, uma taxonomia permite organizar e compreender, de forma estruturada, as diversas soluções, técnicas e abordagens empregadas em determinado tema, como o monitoramento de serviços. Para este trabalho, além de auxiliar pesquisadores na análise de soluções de monitoramento existentes e na proposta de novas abordagens, também beneficia usuários de serviços em nuvem, permitindo que identifiquem soluções de monitoramento que atendam às suas necessidades específicas. No entanto, com base em uma revisão de literatura, não foi possível encontrar uma taxonomia que aborde o monitoramento em várias nuvens do ponto de vista do usuário. Assim, neste capítulo, primeiro, apresentamos a revisão da literatura realizada e, depois, propomos uma taxonomia destinada a auxiliar no desenvolvimento, na análise e na seleção de soluções para o monitoramento de serviços em múltiplas nuvens, com foco nas necessidades e perspectivas de um arquiteto de *software*.

3.1 Revisão de Literatura para Taxonomia em Nuvem

Esta seção apresenta uma revisão da literatura sobre trabalhos que abordam soluções de monitoramento. A Tabela 2 fornece um comparativo dos trabalhos considerando alguns critérios: se a abordagem é feita sobre ferramentas ou soluções; a perspectiva, se é do lado do usuário ou provedor; se trata sobre métricas no monitoramento; se considera a granularidade; o estado dos requisitos; se o trabalho trata do monitoramento de aplicações distribuídas; se propõe taxonomia ou classificação; por fim, se aborda desafios e possíveis direcionamentos para o desenvolvimento de novas soluções de monitoramento.

Inicialmente, é importante observar que nenhum dos estudos encontrados até o momento aborda o monitoramento de múltiplas nuvens, que constitui o foco central deste documento. Até onde nossa pesquisa na literatura nos permite afirmar, este é o primeiro trabalho a explorar o contexto das soluções de monitoramento para ambientes de múltiplas nuvens. Assim, foram realizadas várias pesquisas para encontrar tais trabalhos, no entanto, não foi possível obter sucesso. As pesquisas foram realizadas nas bases: *ACM Digital Library*, *Google Academic*, *IEEE Xplore*, *ScienceDirect*, *Scopus* e *Web of Science*. Algumas

das *strings* de busca que foram elaboradas podem ser observadas na Tabela 1.

Tabela 1 – Strings de busca para encontrar trabalhos sobre monitoramento em múltiplas nuvens (surveys).

# id	String de busca
01	(survey or review or future directions) and (monitorement or monitoring or moni-torement resource management) and (service or application) and (cloud or multiple clouds)
02	future directions and solutions and monitoring of service and (cloud or multiple clouds)
03	(survey or future directions or review) and monitoring of service and (cloud or multiple clouds)
04	(issues or survey or future directions or solutions or review) and cloud monitoring and multiple clouds
05	(solutions or challenges or future directions or review) and cloud monitoring
06	(survey or future directions or review or open researches) and (monitoring of service or service management or cloud monitoring solutions) and (cloud or multiple clouds)
07	challenges and cloud monitoring and cloud

Os dois primeiros critérios **Ferramentas ou Soluções e Perspectiva** analisados focaram em observar se os trabalhos fazem suas abordagens baseadas em ferramentas ou soluções de monitoramento e se estes são da perspectiva do usuário ou do provedor. [Syed et al. \(2017\)](#), [Rodrigues et al. \(2016\)](#), [Kesavulu et al. \(2018\)](#) fizeram suas abordagens centradas em soluções, as quais são o foco deste trabalho. O artigo de [Kesavulu et al. \(2018\)](#) apresenta uma visão geral do monitoramento no nível do usuário e objetiva fornecer princípios para soluções de monitoramento com base em padrões de nuvem. [Syed et al. \(2017\)](#) faz uma análise comparativa e abrangente sobre soluções de monitoramento de nuvem e propõe uma taxonomia. [Rodrigues et al. \(2016\)](#) mostra uma visão geral das soluções de monitoramento de nuvem do cenário atual. No entanto, estes trabalhos abordam o monitoramento de nuvem de uma forma mais geral e, diferente deles, esse projeto tem a premissa de adentrar mais no monitoramento de nuvem no nível do usuário considerando o QoS definido no SLA. Apenas [Kesavulu et al. \(2018\)](#) e [Taherizadeh et al. \(2018\)](#) focaram especificamente no usuário. Os demais, apesar de suas abordagens serem benéficas para ambas as partes, só contemplam a ótica do usuário de forma parcial, ou seja, não fazem uma pesquisa abrangente direcionada ao mesmo. Portanto, mesmo eles considerando o lado do usuário, de forma específica ou não, não se preocupam com aspectos como, por exemplo, os requisitos que o usuário deseja monitorar.

Os critérios **Métricas, Granularidade, Estado do requisito e Aplicações Distribuídas** tiveram o propósito de identificar se os artigos analisam o monitoramento de métricas, como por exemplo o tempo da comunicação entre o usuário e o provedor (*round trip time - RTT*); a granularidade (se monitora a aplicação como um todo, um

Tabela 2 – Comparativo entre trabalhos que tratam do monitoramento de serviços em um único provedor de nuvem.

trabalho	I	II	III	IV	V	VI	VII	VIII
<i>Syed et al. (2017)</i>	ambos	ambos	sim	não	não	não	taxonomia	sim
<i>Rodrigues et al. (2016)</i>	soluções	ambos	não	não	não	não	não aborda	não
<i>Kesavulu et al. (2018)</i>	soluções	usuário	não	não	não	não	não aborda	não
<i>Pourmajidi et al. (2018)</i>	não aborda	ambos	não	não	não	não	não aborda	sim
<i>Birje e Bulla (2019)</i>	não aborda	ambos	não	não	não	não	não aborda	sim
<i>Fahad, Ahmed e Kahar (2017)</i>	ferramentas	ambos	não	não	não	não	não aborda	não
<i>Hauser e Wesner (2018)</i>	não aborda	ambos	sim	não	não	não	não aborda	não
<i>Labidi et al. (2017)</i>	não aborda	ambos	não	não	não	não	não aborda	não
<i>Taherizadeh et al. (2018)</i>	ferramentas	usuário	sim	sim	não	não	taxonomia	sim
<i>Birje e Bulla (2020)</i>	ferramentas	ambos	não	não	não	não	taxonomia	não
esta revisão	soluções	usuário	sim	sim	sim	sim	taxonomia	sim

I - ferramentas ou soluções; II - perspectiva; III - métricas; IV - granularidade; V - estado do requisito; VI - aplicações distribuídas; VII - classificação/taxonomia; VIII - desafios.

serviço, um grupo de serviços etc); o estado dos requisitos (estáticos ou dinâmicos, por exemplo o consumo de CPU que é dinâmico); e se os trabalhos abordam o monitoramento de aplicações distribuídas. Desses critérios, apenas os dois primeiros foram observados. O primeiro critério foi analisado por [Syed et al. \(2017\)](#), [Hauser e Wesner \(2018\)](#) e [Taherizadeh et al. \(2018\)](#). No entanto, os dois primeiros só trataram de métricas relacionadas a preço e a desempenho, ou seja, métricas a nível de infraestrutura, como: consumo de CPU, memória e disco. Já [Taherizadeh et al. \(2018\)](#) abordou tanto métricas como a granularidade. O trabalho foi mais abrangente que os anteriores e contemplou tanto métricas relacionadas a infraestrutura, quanto a aplicação.

Este trabalho, assim como [Taherizadeh et al. \(2018\)](#) também considera a granularidade e métricas, tanto a nível de aplicação, como as definidas no SLA. A abordagem deste trabalho difere da de [Taherizadeh et al. \(2018\)](#) no aspecto de tratar de soluções de monitoramento e não de ferramentas, como também pelo modelo de entrega: *cloud* e *multiple clouds* ao invés de *edge computing*. Este trabalho considera também aplicações distribuídas, seja na *cloud* ou em *multiple clouds*. Assim, é válido destacar a necessidade de considerar todos esses critérios em uma solução de monitoramento de nuvem, pois estes garantem mais flexibilidade para atender as necessidades de monitoramento de nuvem por parte do usuário.

Foi investigado também se os trabalhos propõem alguma **classificação ou taxonomia** para o monitoramento de nuvem. Tal fator foi abordado por [Syed et al. \(2017\)](#), [Taherizadeh et al. \(2018\)](#), [Birje e Bulla \(2020\)](#). No entanto, apenas o trabalho de [Syed et al. \(2017\)](#) é voltado a soluções de monitoramento de nuvem. Este trabalho estende a taxonomia de [Syed et al. \(2017\)](#) e propõe uma nova taxonomia. Dessa forma, acredita-se

que a proposta da mesma poderá facilitar a identificação dos pontos em aberto e desafios futuros e/ou necessidades do usuário.

Outro fator observado, foram **desafios** para o monitoramento de nuvem. O trabalho de [Pourmajidi et al. \(2018\)](#) se concentrou principalmente neste aspecto de mostrar desafios para soluções de monitoramento e destacar a importância de mais pesquisas. [Birje e Bulla \(2019\)](#) também trabalhou sobre esta ideia acerca de Sistemas de Monitoramento de Nuvem e fez um comparativo do estado da arte neste domínio. Alguns dos desafios identificados fazem relação ao monitoramento nativo da nuvem, gerenciamento de monitoramento de problemas de armazenamento de dados, escalabilidade e gerenciamento de SLA ([SYED et al., 2017](#)), desempenho, manutenção, interoperabilidade ([BIRJE; BULLA, 2019](#)) e estabelecimento de estratégias de alta disponibilidade ([POURMAJIDI et al., 2018](#)). Este trabalho analisará todos esses desafios já tratados, bem como identificará desafios em aberto.

3.2 Taxonomia Proposta

A taxonomia proposta para o monitoramento de múltiplos provedores de nuvem tem como base a taxonomia proposta por [Carvalho et al. \(2018\)](#) para gerenciamento de recursos em múltiplas nuvens e a proposta por [Syed et al. \(2017\)](#) para monitoramento de nuvem.

A Figura 5 apresenta uma visão geral dos critérios que a compõe a taxonomia proposta para monitoramento de múltiplos provedores de nuvem, sendo eles: [3.2.1](#) Objetivo; [3.2.2](#) Perspectiva; [3.2.3](#) Modelo de Implantação; [3.2.4](#) Modelo de Entrega; [3.2.5](#) Modelo de Serviço; [3.2.6](#) Arquitetura da Aplicação; [3.2.7](#) Granularidade; [3.2.8](#) Estado dos requisitos; [3.2.9](#) Quantidade de requisitos.

3.2.1 Objetivo para um Monitoramento

O propósito do monitoramento permite entender a necessidade e o escopo como um todo. Os critérios de interesse mais comuns para monitorar serviços estão centrados em monitorar: a disponibilidade dos serviços, o desempenho de tempo, o custo e o SLA. O monitoramento pode ser individual ou múltiplos critérios e observando as prioridades do usuário.

A computação em nuvem adota um modelo de pagamento de acordo com o uso. Informações precisas de contabilidade e faturamento são muito importantes tanto para o cliente quanto para o provedor de serviços ([SIM, 2016](#)). Assim, o faturamento representa um papel vital para apoiar esse conceito, tendo em vista a importância de ter faturas precisas, integras e verificáveis. Para a execução de tal tarefa, as formas de cobrança exigem a captura de informações exatas de consumo e alocação de recursos virtuais, bem como a

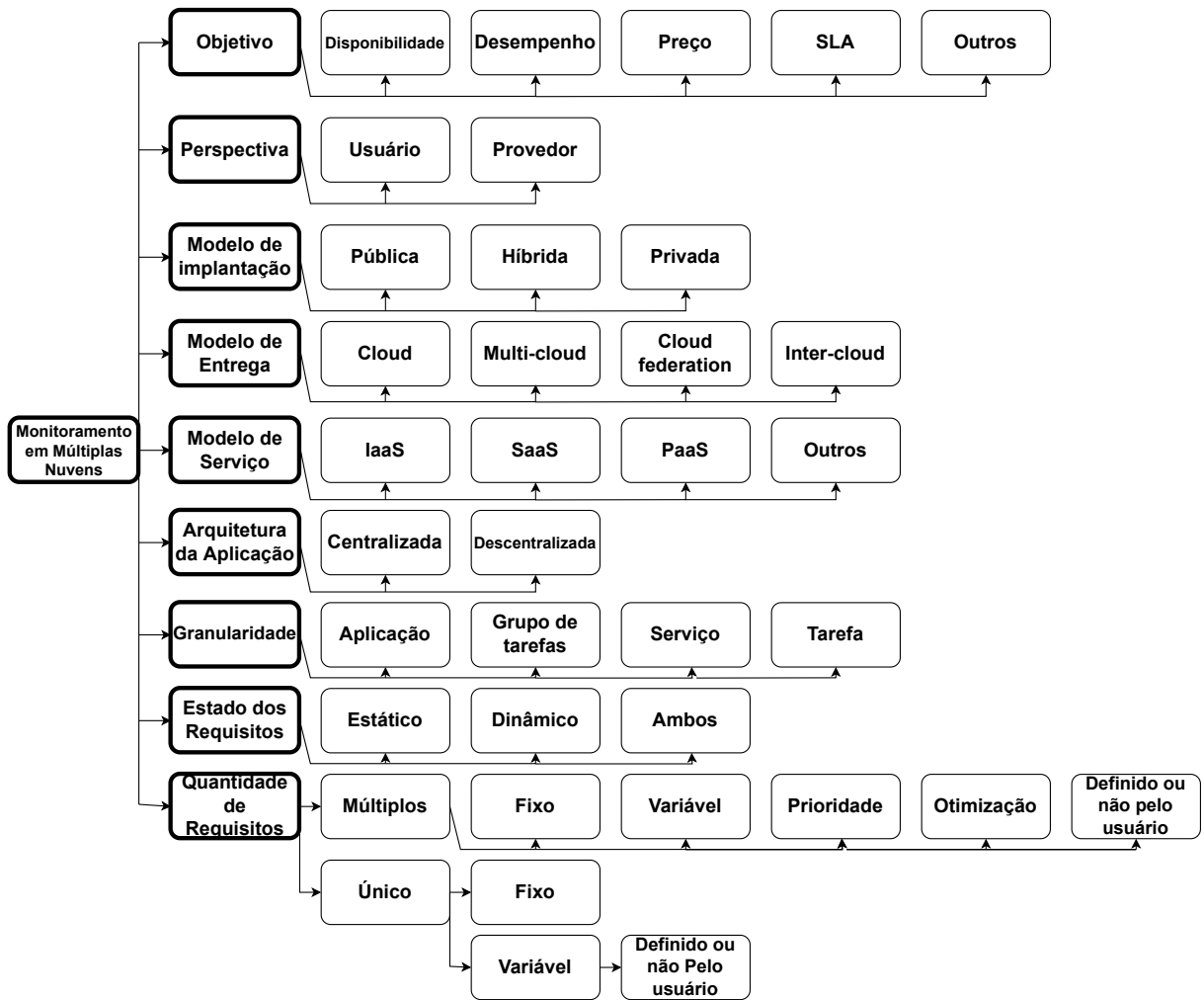


Figura 5 – Taxonomia proposta para o monitoramento em múltiplas nuvens.

natureza da atividade de uso de recursos de TI em tempo de execução (BIRJE; BULLA, 2019). As soluções de monitoramento em nuvem exploram os dados métricos coletados para gerar faturas precisas e verificáveis (SYED et al., 2017). O desempenho de tempo também é um critério fundamental para o monitoramento. Ele influencia diretamente na concepção, aquisição e utilização de sistemas computacionais. Um sistema de avaliação de desempenho tem como objetivo principal auxiliar a tomada de decisão. Nesse contexto, atentar para o bom funcionamento da infraestrutura física evita violações do SLA e assegura a estabilidade e o desempenho adequado da mesma. Logo, as soluções de monitoramento são importantes, pois ajudam a detectar falhas e melhorar o desempenho de tal infraestrutura com o mínimo de intervenção humana (LEVIN et al., 2017; SYED et al., 2017). Por fim, a auditoria do SLA tem igual importância tanto para o provedor quanto para o usuário. O monitoramento de SLA contribui para garantir o QoS das aplicações em nuvem (ANITHAKUMARI; CHANDRASEKARAN, 2015).

3.2.2 Perspectiva do Monitoramento

A perspectiva está diretamente associada com o propósito do monitoramento. O monitoramento se dá em duas principais vertentes: na ótica do usuário e na do provedor. Os provedores de nuvem e os usuários possuem necessidades de monitoramento distintas. O provedor de serviço normalmente precisa verificar anomalias a nível de *hardware* e *software* físicos, bem como otimizar a utilização geral do *data center*, colocando os recursos virtuais da maneira mais otimizada possível nos físicos (HAUSER; WESNER, 2018). Já os clientes de nuvem monitoram sua infraestrutura virtual com métricas adicionais específicas a nível de aplicativos, com o intuito de identificar deficiências de recursos ou gastos excessivos de recursos. Como já mencionado, o usuário preocupa-se principalmente com os custos, performance, disponibilidade e com as definições contidas no SLA.

3.2.3 Tipo de Nuvem

A computação em nuvem pode ser dividida em três principais modelos de implantação, sendo eles: nuvem pública, nuvem privada e híbrida. O monitoramento deve necessariamente considerar este modelo de implantação, pois estas possuem políticas de uso e acesso distintos. Em uma nuvem privada o usuário por ser o proprietário, tem mais domínio sobre a infraestrutura como um todo. Assim, o mesmo tem mais facilidade de desenvolver técnicas sobre o fluxo entre a aplicação e os serviços que a mesma depende. No entanto, no monitoramento em nuvens públicas e híbridas, o arquiteto não tem controle sobre a infraestrutura subjacente, tornando o monitoramento uma tarefa difícil. Dessa forma, o arquiteto se torna dependente de ferramentas/técnicas do próprio provedor de nuvem. Logo, o usuário não tem liberdade de customizar o que deseja monitorar, bem como não tem visibilidade do desempenho da aplicação.

3.2.4 Modelo de Entrega

As soluções de monitoramento de nuvem podem monitorar uma única nuvem ou várias nuvens. Neste cenário, existem quatro principais modelos de entrega, os quais: *cloud* que é um modelo de nuvem única e *multi-cloud*, *cloud-federation* e *inter-cloud* que são modelos de múltiplas nuvens. O interesse do usuário de distribuir seus serviços em várias nuvens tem aumentado significativamente. Os motivos se devem principalmente a escalabilidade e melhores ofertas. No entanto, cada um desses modelos de entrega possui particularidades em relação ao monitoramento, principalmente no que se refere ao acesso a dados e informações, bem como distribuição e alocação de recursos/serviços. Os sistemas de monitoramento de nuvem atuais identificados na literatura apresentam várias limitações, pois só monitoram a plataforma específica, bem como não fornecem uma visão mais completa para o usuário a nível de aplicação. Logo, é uma necessidade que as partes interessadas do serviço em nuvem busquem um sistema de monitoramento que

seja independente de provedor de nuvem e que seja interoperável ([TRIHINAS; PALLIS; DIKAIKOS, 2015](#)).

3.2.5 Modelo de Serviço

Os modelos de serviço da nuvem dizem respeito a forma como os serviços são oferecidos pelo provedor de nuvem, bem como são acessados pelo usuário. Nessa taxonomia foram considerados os três principais modelos de serviço encontrados na literatura:

- Infraestrutura como Serviço (IaaS) - o usuário possui um controle maior sobre a infraestrutura contratada e se responsabiliza por gerenciar os recursos computacionais, como processamento, memória e armazenamento de um provedor IaaS;
- Plataforma como Serviço (PaaS) - o usuário só se preocupa com desenvolvimento da aplicação. Nesse nível a configuração e manutenção da infraestrutura ficam a cargo do fornecedor de PaaS;
- Software como Serviço (SaaS) - os usuários simplesmente utilizam um navegador da web para acessar o *software* que outros desenvolvem e oferecem como serviço na web.

3.2.6 Arquitetura da Aplicação

A arquitetura de implantação da aplicação permite compreender como os serviços estão sendo consumidos e como traçar estratégias para obter dados dos mesmos. A arquitetura pode ser do tipo centralizada ou descentralizada. O projeto de arquitetura centralizada define que a aplicação está implantada em um só provedor de nuvem. Já na arquitetura descentralizada a aplicação pode estar distribuída no mesmo provedor de nuvem ou em vários provedores.

Na arquitetura centralizada um ou mais nós clientes são conectados diretamente a um servidor central, ou seja, tal nó atua como um ponto central e tem controle sobre todo o fluxo de informações. Assim, no monitoramento centralizado tem-se uma probabilidade maior de ocorrerem colapsos, caso haja indisponibilidade desse servidor central ([SYED et al., 2017](#)). Nesse tipo de arquitetura, o monitoramento se torna mais fácil devido a existência de tal centralizador. Na arquitetura descentralizada, o aplicativo pode ser distribuído em vários contêineres/VMs e em vários provedores. Logo, para aplicações intensivas em contêineres, é um desafio ter um ponto centralizador para o monitoramento, pois cada contêiner pode ser hospedado em locais diferentes que não oferecem suporte a um ponto/*endpoint* de monitoramento comum ([NOOR et al., 2019](#)).

3.2.7 Granularidade do Monitoramento

A granularidade define a escalabilidade da solução de monitoramento em relação aos requisitos do usuário. Dessa forma, é possível identificar o quanto é possível refinar os requisitos para o monitoramento. De acordo com (CARVALHO et al., 2018) existem quatro categorias em que os requisitos podem ser minimizados:

- Aplicação - cenário no qual o usuário deseja monitorar e obter métricas sobre a aplicação como um todo. Neste caso, todos os serviços da aplicação possuem os mesmos requisitos;
- Serviço - quando o usuário deseja monitorar um serviço ou serviços em específico, por exemplo, o banco de dados, a VM entre outros. Nesse cenário os serviços possuem requisitos distintos;
- Grupo de tarefas e Tarefa - quando o usuário deseja monitorar rotinas/tarefas da aplicação, por exemplo, monitorar as rotinas de um microsserviço ou microsserviços. Aqui os requisitos também podem ser distintos.

3.2.8 Estado dos Requisitos

Os requisitos do usuário podem ser classificados em três estados: estático, dinâmico e ambos, conforme descrito em (CARVALHO et al., 2018).

- Requisitos estáticos permanecem constantes ao longo do processo de monitoramento, sem sofrer alterações nos valores inicialmente definidos.
- Requisitos dinâmicos, por outro lado, apresentam valores que podem variar durante a atividade de monitoramento. Um exemplo típico são os requisitos especificados em SLAs que permitem escalabilidade, adaptando-se automaticamente às demandas da aplicação.
- Por fim, os requisitos podem ser configurados de forma a abranger tanto o estado estático quanto o dinâmico, permitindo flexibilidade no monitoramento.

Essa flexibilidade possibilita ao arquiteto de software monitorar requisitos de diferentes naturezas, adaptando-se a cenários em que os valores se mantêm fixos ou mudam ao longo do tempo, de acordo com as necessidades do sistema e dos usuários.

3.2.9 Quantidade de Requisitos

Uma solução de monitoramento deve também levar em consideração a quantidade de requisitos configurados para a aplicação. O arquiteto de *software* pode definir um único

requisito ou múltiplos requisitos para serem monitorados simultaneamente (CARVALHO et al., 2018). Quando optar por monitorar múltiplos requisitos, é necessário estabelecer prioridades, definir se os requisitos são fixos ou variáveis, e determinar se podem ou não ser configurados pelo usuário.

A prioridade determina o peso ou a importância relativa de cada requisito, ajudando a otimizar o monitoramento de acordo com os objetivos desejados. Além disso, é possível especificar quais requisitos devem ser maximizados (ex.: desempenho) ou minimizados (ex.: latência). O estado (fixo ou variável) identifica se o valor de cada requisito permanece constante ou pode oscilar ao longo do monitoramento, permitindo maior flexibilidade para requisitos dinâmicos. Com a personalização algumas soluções permitem ao arquiteto definir quais requisitos serão monitorados e seus respectivos valores, enquanto outras oferecem configurações predefinidas, limitando a personalização. No caso de monitoramento de um único requisito, o arquiteto ainda precisa determinar se ele é fixo ou variável, bem como sua prioridade dentro do contexto do sistema. Portanto, para garantir uma abordagem flexível, é essencial que a solução de monitoramento possibilite tanto a configuração quanto a adaptação dos requisitos às necessidades específicas da aplicação e do ambiente.

4 Trabalhos Relacionados

Neste capítulo, apresentamos o processo de identificação e descrição de estudos na literatura sobre monitoramento em *multi-cloud* sob a perspectiva de um arquiteto de *software*. Esse processo envolveu a definição de métricas e questões de pesquisa (Seção 4.1), com base na taxonomia proposta (Capítulo 3). Também conduzimos um processo de seleção e exclusão para identificar trabalhos relevantes, utilizando *strings* de busca ilustradas na imagem apresentada na Seção 4.2. Por fim, realizamos uma análise comparativa dos trabalhos, tomando como referência as questões de pesquisa.

4.1 Questões de Pesquisa

Esta seção detalha as questões de pesquisa que definimos para guiar a análise de soluções de monitoramento de nuvem.

- **Questão 01: A solução monitora uma nuvem ou múltiplas nuvens?** esta questão visa obter os modelos de entrega suportados pelas soluções de monitoramento;
- **Questão 02: O monitoramento é proposto sob a perspectiva do usuário ou do provedor de nuvem?** procuramos entender qual o foco da solução proposta, se é para beneficiar o usuário ou o provedor de nuvem;
- **Questão 03: A solução monitora quais modelos de serviço?** Esta questão busca identificar a quais modelos de serviço a solução de monitoramento pode ser aplicada: IaaS, SaaS, PaaS ou outros;
- **Questão 04: Qual é o nível de granularidade do monitoramento?** Esta questão objetiva identificar o nível de granularidade abordado pelo monitoramento, ou seja, se a solução monitora a aplicação como um todo, um conjunto de tarefas, uma tarefa específica, um serviço, entre outros;
- **Questão 05: O estado dos requisitos é estático ou dinâmico?** Esta questão observa se os requisitos monitorados permanecem em um estado estático ou se seus valores mudam ao longo do processo de monitoramento, ou seja, se a solução monitora requisitos que podem mudar de acordo com um cenário específico;
- **Questão 06: Qual é o número de requisitos a serem monitorados?** esta questão visa identificar se a solução monitora apenas um requisito por vez ou múltiplos requisitos simultaneamente. No caso de múltiplos, se estes são definidos pelo usuário ou não, se há uma ordem de prioridade ou otimização definida, e se esses requisitos são fixos ou variáveis;

- **Questão 07: Qual é a arquitetura da aplicação?** esta questão investiga se a solução proposta monitora aplicações com arquitetura centralizada ou descentralizada.

4.2 Revisão de literatura sobre soluções de monitoramento

Esta seção identifica trabalhos na literatura que abordam soluções de monitoramento em múltiplas nuvens. Nosso foco está voltado para propostas de caráter acadêmico, razão pela qual ferramentas comerciais de monitoramento foram desconsideradas. Para alcançar esse objetivo, apresentamos inicialmente o processo adotado para a seleção dos trabalhos incluídos na análise. Foram considerados estudos publicados entre 2021 e 2025, pesquisados nas bases de dados *Scopus* e *Web of Science*, as quais também abrangem repositórios como *ACM*, *ScienceDirect*, *IEEE Xplore* e *SpringerLink*. Adicionalmente, apenas trabalhos que tratam de soluções, arquiteturas, *frameworks* ou categorias semelhantes foram incluídos, restringindo-se a trabalhos escritos em inglês.

A Figura 6 mostra um resumo da metodologia adotada para a análise dos trabalhos. O processo inicia-se com a definição das questões de pesquisa, seguida pela identificação dos artigos e pelas etapas de seleção, que incluem a triagem por título e, posteriormente, a avaliação baseada no resumo e na conclusão.



Figura 6 – Fluxo de revisão de soluções de monitoramento de múltiplas nuvens.

Podemos observar ainda na Figura 6 que, ao todo, foram identificados 5.119 artigos, dos quais 66 foram selecionados na seleção por título. Em seguida, foram removidas 11 duplicações e aplicada a seleção final a partir da leitura do resumo e da conclusão. Como resultado, 25 artigos foram mantidos para a análise final.

A Tabela 3 apresenta um resumo dos estudos selecionados. Os trabalhos são organizados por país, base de publicação e ano. A maioria foi publicada entre 2021 e 2024, evidenciando que a pesquisa sobre monitoramento em múltiplas nuvens ganhou impulso significativo nos últimos anos. A Índia se destaca como o país mais representado, seguida por China, Paquistão e Nigéria, o que revela um forte interesse sobre o tema em economias emergentes. A distribuição também demonstra relevância global, com contribuições da Europa, América do Norte e América do Sul. A *IEEE* aparece como a principal base de publicação, sugerindo elevado rigor técnico nos trabalhos selecionados. Outras editoras, como *Springer*, *ScienceDirect* e *InderScience*, também estão bem representadas, reforçando o caráter multidisciplinar da pesquisa. De forma geral, a tabela ilustra um panorama de pesquisa em expansão, geograficamente diverso e tecnicamente robusto na área de monitoramento de várias nuvens.

Tabela 3 – Resumo dos artigos por país, local de publicação e ano.

Trabalho	País	Local de publicação	Ano de publicação
(KHAN et al., 2021)	Paquistão	Springer	2021
(ZHAN et al., 2021)	China	Tech Science Press	2021
(ABHISHEK; CHOBARI; NARAYAN, 2021)	Índia	IEEE	2021
(VIEGAS et al., 2021)	Brasil	Springer	2021
(ZEDAN; ATTIYA; EL-FISHAWY, 2021)	Egito	IEEE	2021
(BOSE; SUJATHA, 2021)	Índia	JATIT	2021
(PANDEY; NARAYAN; SHIVARAJ, 2021)	Índia	IEEE	2021
(CEDILLO et al., 2021)	Equador	IEEE	2021
(KHAN et al., 2022)	Paquistão	ScienceDirect	2022
(TAN et al., 2022)	China	Taylor & Francis Online	2022
(MOEYERSONS et al., 2022)	Bélgica	Wiley	2022
(MURALIDHARAN et al., 2022)	Índia	IET Communications	2022
(ABDULLAH; SURPUTHEEN, 2022)	Índia	IJCSNS	2022
(NZANZU et al., 2022)	Nigéria	IEEE	2022
(KIM et al., 2022)	EUA	IEEE	2022
(AFZAL et al., 2022)	Paquistão	Big Data Magazine	2022
(NADEEM et al., 2023)	Paquistão	MDPI Magazine	2023
(ALONSO; HUARTE; ARRIETA, 2023)	Espanha	InderScience	2023
(GUO et al., 2023)	China	IEEE	2023
(RAJAKUMAR et al., 2023)	Índia	IEEE	2023
(MARQUES et al., 2024)	Portugal	ScienceDirect	2024
(SHENDE; CHANDAK, 2024)	Índia	Grenze	2024
(SAIFENG, 2024)	China	InderScience	2024
(ADETIBA et al., 2024)	Nigéria	IEEE	2024
(WU et al., 2025)	China	ScienceDirect	2025

A Tabela 4 fornece uma visão geral de cada um dos trabalhos selecionados. O objetivo é resumir a ideia central e as principais contribuições de cada trabalho, destacando

Tabela 4 – Visão geral dos trabalhos relacionados e suas propostas principais.

Trabalho	Objetivo
(KHAN et al., 2021)	Monitorar SLAs em tempo real usando blockchain
(ZHAN et al., 2021)	Introspecção de contêineres por meio de contêineres externos para monitoramento leve e isolado
(ABHISHEK; CHOBARI; NARAYAN, 2021)	Detectar violações de SLA em ambientes multi-cloud usando blockchain (Hyperledger Fabric)
(VIEGAS et al., 2021)	Manutenibilidade de serviços com monitoramento e auditoria dinâmicos de violações de SLA
(ZEDAN; ATTIYA; EL-FISHAWY, 2021)	Combinar monitoramento ativo com balanceamento de carga para manter desempenho em nuvem
(BOSE; SUJATHA, 2021)	Framework de monitoramento sensível a QoS com políticas adaptativas baseadas em métricas e limiares pré-definidos
(PANDEY; NARAYAN; SHIVARAJ, 2021)	Integrar monitoramento de SLA com compensação automática de violações usando blockchain
(CEDILLO et al., 2021)	Avaliar um método de monitoramento em tempo de execução baseado em modelos formais
(KHAN et al., 2022)	Monitoramento adaptativo em tempo de execução que ajusta políticas com base nos dados de desempenho atuais
(TAN et al., 2022)	Modelo de SLA baseado em contrato para manufatura em nuvem, com rastreamento automatizado via blockchain
(MOEYERSONS et al., 2022)	Monitorar desempenho de aplicações de forma remota e não intrusiva em ambientes multi-vendor
(MURALIDHARAN et al., 2022)	Sugerir modelo de SLA em três fases com gerenciamento de confiança via broker de nuvem
(ABDULLAH; SUR-PUTHEEN, 2022)	Desenvolver framework baseado em agentes cooperativos para alocação dinâmica de recursos e monitoramento de SLA
(NZANZU et al., 2022)	Criar arquitetura de monitoramento federado escalável para múltiplos domínios administrativos
(NZANZU et al., 2022)	Garantir SLA durante picos de uso com agendador sensível a recursos que monitora comportamento de aplicações
(AFZAL et al., 2022)	Fornecer SLAaaS como serviço de auditoria independente para aumentar confiança e neutralidade
(NADEEM et al., 2023)	Integrar monitoramento em tempo de execução com provisionamento de recursos para prevenir violações de SLA
(ALONSO; HUARTE; ARRIETA, 2023)	Monitorar serviços em nuvem federada considerando desafios de interoperabilidade e escalabilidade
(GUO et al., 2023)	Criar modelo de confiança que avalia provedores com base em atributos de SLA e indicadores comportamentais.
(RAJAKUMAR et al., 2023)	Monitoramento no nível do cliente para melhorar segurança e transparência dos dados
(MARQUES et al., 2024)	Antecipar demandas de aplicações e ajustar alocação de recursos para manter desempenho
(SHENDE; CHANDAK, 2024)	Otimizar custos em nuvem combinando alocação automática de recursos e monitoramento em tempo real
(SAIFENG, 2024)	Gerenciar QoS com negociação inteligente e monitoramento adaptativo, ajustando SLAs dinamicamente
(ADETIBA et al., 2024)	Monitorar recursos e eventos em nuvens federadas com notificações em tempo real
(WU et al., 2025)	Arquitetura distribuída para monitoramento coordenado em infraestruturas de nuvem conjunta
Este trabalho	Monitoramento de QoS em múltiplas nuvens sob a ótica do usuário

sua abordagem para monitoramento em ambientes de nuvem e como elas abordam os desafios identificados.

4.3 Análises e Discussões

Para guiar nossa discussão, esta seção fornece um resumo e uma análise dos estudos listados na Tabela 3. Examinamos se os trabalhos selecionados abordam efetivamente os aspectos centrais do monitoramento de múltiplas nuvens, conforme definido pela taxonomia apresentada no Capítulo 3.

Primeiramente, analisamos os resultados obtidos com a questão de pesquisa 01 (**A solução monitora múltiplas nuvens ou uma única nuvem?**). Esta questão visa avaliar o escopo das abordagens de monitoramento propostas na literatura, particularmente se elas são projetadas para ambientes de nuvem única ou se são capazes de operar em múltiplos provedores de nuvem. Compreender essa distinção é essencial, visto que estratégias de múltiplas nuvens são cada vez mais comuns na prática devido aos seus potenciais benefícios em termos de tolerância a falhas, otimização de custos e flexibilidade. A análise da Figura 7 revelou que 72% dos estudos revisados se concentram exclusivamente no monitoramento de uma única nuvem, enquanto apenas 28% consideram o monitoramento de múltiplas nuvens. Esses resultados indicam um foco predominante em cenários de nuvem única e sugerem a necessidade de mais pesquisas sobre soluções de monitoramento que possam abordar efetivamente a complexidade adicional e os desafios de interoperabilidade impostos por ambientes múltiplas nuvens.

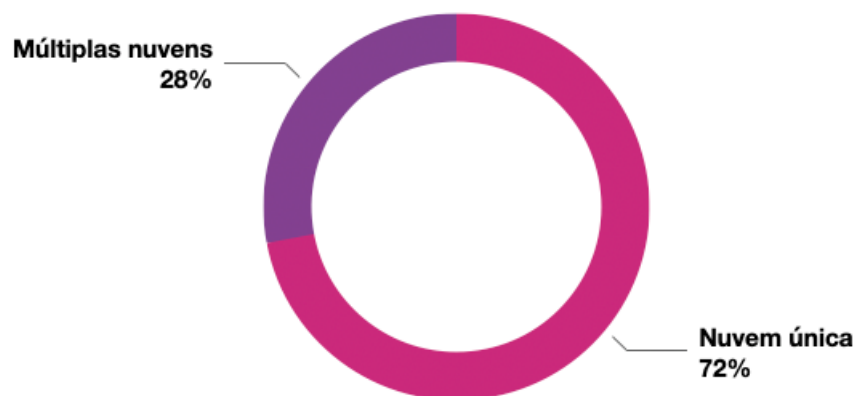


Figura 7 – Visão geral dos resultados para o modelo de entrega adotado pelas soluções.

A questão 02 trata da identificação da perspectiva de monitoramento adotada pelas soluções, seja ela projetada sob a ótica do arquiteto de *software*, do provedor de nuvem ou de ambos. A Figura 8 apresenta a distribuição dos estudos analisados, mostrando que 8 (32%) trabalhos adotam a perspectiva do lado do provedor, 7 (28%) adotam a perspectiva do lado do usuário e 9 (36%) consideram ambas as perspectivas. Apenas em um dos estudos não foi possível determinar a perspectiva adotada com base nas informações disponíveis.

Embora se observe uma distribuição relativamente equilibrada, permanece uma tendência notável ao monitoramento centrado no provedor. Isso sugere que muitas soluções podem ser limitadas em oferecer a flexibilidade necessária aos usuários para definir e controlar o que está sendo monitorado. Soluções que suportem ambas as perspectivas são essenciais para permitir um monitoramento mais abrangente e personalizável, empoderando os usuários e mantendo o alinhamento com as operações em nível de provedor.

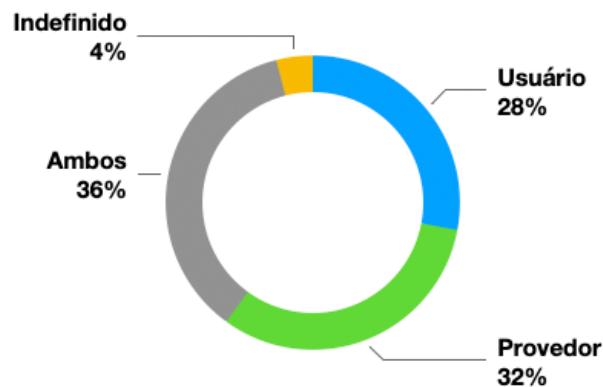


Figura 8 – Visão geral dos resultados sobre as perspectivas de monitoramento.

A questão de pesquisa 03 focou na identificação dos modelos de serviços aos quais as soluções possuem suporte para o monitoramento. Os resultados são ilustrados na Figura 9, que revela uma forte ênfase em Infraestrutura como Serviço (IaaS). Especificamente, 18 estudos focam exclusivamente em IaaS, enquanto apenas um número limitado aborda combinações com PaaS ou SaaS, e apenas um estudo abrange todos os três modelos. Essa distribuição indica que a maioria das abordagens existentes ainda está centrada no monitoramento em nível de infraestrutura.

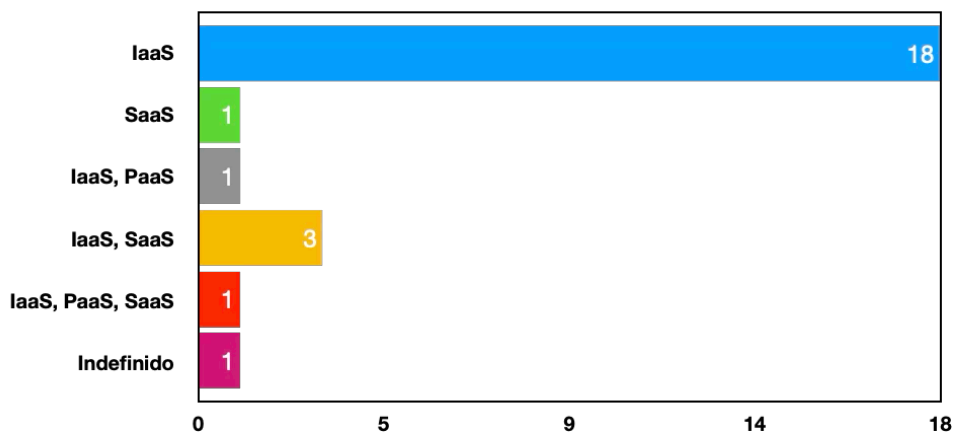


Figura 9 – Resumo dos resultados sobre o modelo de serviço.

Além disso, a Figura 9 ainda destaca uma lacuna em soluções abrangentes que possam atender aos requisitos distintos de modelos de serviço de alto nível. Também é importante observar que modelos de serviço emergentes, como Banco de Dados como Serviço (DBaaS), não estão representados nos estudos revisados, sugerindo que pesquisas

futuras devem expandir seu escopo para incluir esses paradigmas cada vez mais relevantes em ambientes de computação em nuvem.

A Figura 10 mostra a distribuição dos níveis de granularidade de monitoramento identificados nos estudos analisados. A maioria dos trabalhos (13) adota uma granularidade em nível de serviço, indicando um foco predominante no monitoramento de serviços em nuvem, como máquinas virtuais, bancos de dados ou contêineres. Isso reflete uma abordagem de monitoramento de granularidade fina, oferecendo maior visibilidade da infraestrutura e permitindo um controle mais preciso sobre componentes individuais. Seis estudos adotam uma granularidade ainda mais fina no nível de tarefa, permitindo o monitoramento de unidades de execução específicas dentro dos serviços, o que pode ser particularmente útil para ajuste de desempenho e detecção de falhas. Em contraste, apenas cinco estudos operam no nível do aplicativo, refletindo uma granularidade mais geral que pode ser mais fácil de implementar, mas oferece menos detalhes. Notavelmente, um estudo não especificou claramente a granularidade considerada. Essas descobertas sugerem uma clara preferência pelo monitoramento de granularidade fina em ambientes de nuvem, o que se alinha à crescente necessidade de observabilidade e controle detalhados em sistemas distribuídos. Soluções que suportam múltiplos níveis de granularidade são especialmente valiosas, pois fornecem flexibilidade para adaptar estratégias de monitoramento de acordo com diferentes necessidades operacionais e arquitetônicas.

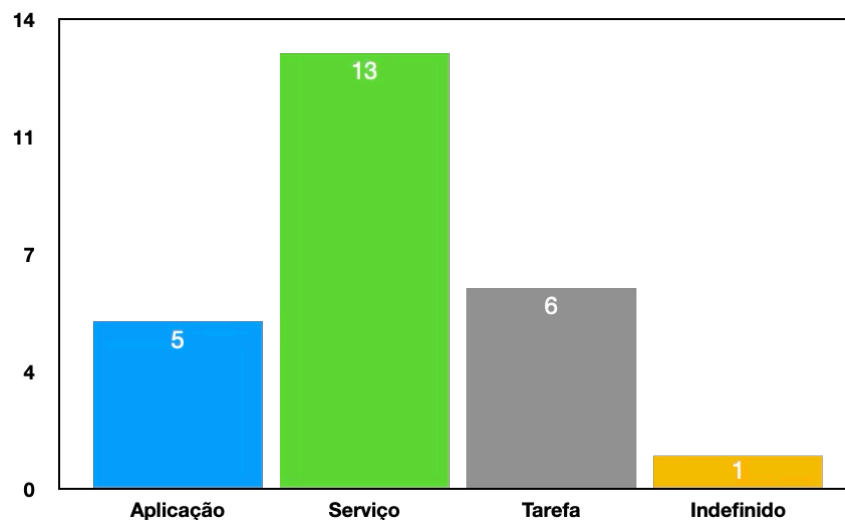


Figura 10 – Resultados da investigação sobre a granularidade do monitoramento.

A Figura 11 resume os resultados da questão 05, que verifica o estado dos requisitos monitorados. A análise dos critérios observados nas soluções avaliadas revela uma predominância do monitoramento dinâmico, com 14 das 25 soluções focando em parâmetros voláteis, como uso de CPU e consumo de memória. Requisitos estáticos, que normalmente representam aspectos mais estáveis e com alterações menos frequentes, como disponibilidade, são monitorados por nove soluções. Curiosamente, apenas uma solução

aborda requisitos estáticos e dinâmicos simultaneamente, indicando uma potencial lacuna em abordagens abrangentes de monitoramento. Além disso, uma solução se enquadra em uma categoria indefinida, indicando a falta de uma classificação precisa ou uma estratégia de monitoramento mista. Esses resultados destacam uma tendência geral na qual a maioria das ferramentas de monitoramento prioriza métricas dinâmicas relacionadas ao desempenho, possivelmente devido ao seu impacto significativo no comportamento do sistema e na experiência do usuário. No entanto, a ênfase relativamente menor em requisitos estáticos pode indicar oportunidades para estruturas de monitoramento mais holísticas que integrem aspectos estáveis e voláteis do sistema para aprimorar a confiabilidade e o *insight*.

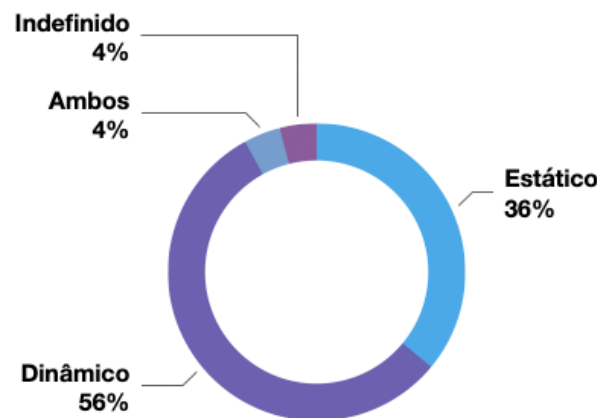


Figura 11 – Resultados da investigação sobre o estado dos requisitos de monitoramento considerados pelas soluções.

A Tabela 5 apresenta uma visão geral dos requisitos, considerando tanto a quantidade de itens monitorados quanto o grau de flexibilidade concedido ao usuário na definição dos parâmetros de monitoramento. De acordo com a análise desta tabela, embora a maioria das abordagens revisadas suporte o monitoramento de múltiplos requisitos, um número significativo delas negligencia a priorização e a otimização desses requisitos. Essa é uma limitação relevante, especialmente considerando a crescente complexidade dos sistemas baseados em nuvem, onde diferentes serviços podem ter objetivos de nível de serviço (SLOs) conflitantes ou interdependentes. O monitoramento de múltiplos requisitos é essencial para garantir uma compreensão abrangente do desempenho do sistema e da experiência do usuário. No entanto, sem mecanismos para atribuir prioridades ou otimizar as compensações entre esses requisitos, os esforços de monitoramento podem se tornar ineficientes ou não atingir os objetivos.

Além disso, habilitar requisitos definidos pelo usuário permite maior flexibilidade e adaptabilidade ao processo de monitoramento. Isso permite que os sistemas alinhem as metas de monitoramento com necessidades comerciais específicas ou contextos de aplicação. Apesar disso, vários estudos ainda dependem de requisitos fixos ou definidos pelo sistema, o que pode não refletir a natureza dinâmica das implantações no mundo real. Entre os trabalhos analisados, apenas (ABDULLAH; SURPUTHEEN, 2022), (KIM et al.,

Tabela 5 – Resumo da quantidade de pesquisas e flexibilidade dos requisitos considerados no monitoramento.

Trabalho	Requisitos		Def. pelo usuário
	Único/Múltiplo	Fixo/Variável	
(KHAN et al., 2021)	único	variável	não
(ZHAN et al., 2021)	múltiplo	variável	não
(ABHISHEK; CHOBARI; NARAYAN, 2021)	único	fixo	sim
(VIEGAS et al., 2021)	múltiplo	variável	não
(ZEDAN; ATTIYA; EL-FISHAWY, 2021)	múltiplo	variável	não
(BOSE; SUJATHA, 2021)	múltiplo	variável	não
(PANDEY; NARAYAN; SHIVARAJ, 2021)	único	fixo	sim
(CEDILLO et al., 2021)	múltiplo	variável	sim
(KHAN et al., 2022)	múltiplo	variável	sim
(TAN et al., 2022)	múltiplo	variável	sim
(MOEYERSONS et al., 2022)	múltiplo	variável	sim
(MURALIDHARAN et al., 2022)	único	variável	sim
(ABDULLAH; SURPUTHEEN, 2022)	múltiplo	variável	sim
(NZANZU et al., 2022)	múltiplo	variável	sim
(KIM et al., 2022)	múltiplo	variável	não
(AFZAL et al., 2022)	múltiplo	variável	sim
(NADEEM et al., 2023)	único	variável	não
(ALONSO; HUARTE; ARRIETA, 2023)	múltiplo	variável	não
(GUO et al., 2023)	único	fixo	sim
(RAJAKUMAR et al., 2023)	único	fixo	não
(MARQUES et al., 2024)	múltiplo	variável	sim
(SHENDE; CHANDAK, 2024)	múltiplo	variável	sim
(SAIFENG, 2024)	múltiplo	variável	sim
(ADETIBA et al., 2024)	múltiplo	variável	não
(WU et al., 2025)	múltiplo	variável	não

2022) e (SHENDE; CHANDAK, 2024) abordam a priorização ou otimização de requisitos de monitoramento. Esses estudos representam avanços significativos no reconhecimento da necessidade de gerenciar múltiplos objetivos de monitoramento. Isso destaca uma direção para pesquisas futuras: desenvolver soluções de monitoramento que não sejam apenas capazes de lidar com requisitos múltiplos e dinâmicos, mas também suportem sua priorização e otimização com base nas preferências do usuário ou nos objetivos do sistema.

De acordo com as análises realizadas podemos fazer as seguintes pontuações:

- Questão 01: Vale destacar que apenas cerca de 27% dos estudos focaram no monitoramento de múltiplas nuvens sob a perspectiva do usuário;

- Questão 02: Poucos trabalhos são direcionados especificamente a arquitetos de *software*. A maioria considera esse critério apenas de forma superficial;
- Questão 03: A maior parte das pesquisas concentra-se no monitoramento de infraestrutura, sendo que um dos estudos aborda ambientes de múltiplas nuvens (IaaS, PaaS e SaaS);
- Questão 04: Cerca de 6 dos trabalhos selecionados consideram o monitoramento em granularidade fina. No entanto, ainda é necessário permitir que os usuários sejam mais específicos e monitorem componentes menores, como rotinas dentro de sua aplicação;
- Questão 05: Em relação a esta questão, a maioria das soluções já está bem preparada para lidar com a natureza dinâmica dos requisitos monitorados;
- Questão 06: Alguns dos trabalhos selecionados permitem que o usuário personalize os requisitos de monitoramento. No entanto, não há evidências de gestão desses requisitos. Ou seja, não é possível definir prioridades, otimizar e ordenar.

Assim, com base em nossa taxonomia e na investigação das soluções de monitoramento de nuvem existentes, observamos que, embora muitas soluções forneçam as funcionalidades necessárias, elas ainda não conseguem abordar completamente as complexidades do monitoramento de múltiplas nuvens. A maioria das abordagens analisadas é desenvolvida principalmente sob a perspectiva de provedores de nuvem, priorizando a visibilidade em nível de infraestrutura em detrimento das necessidades dos usuários. Isso indica uma necessidade de soluções que adotem uma perspectiva mais centrada no usuário.

Com a heterogeneidade dos serviços de nuvem e a crescente adoção de modelos como Plataforma como Serviço (PaaS), Banco de Dados como Serviço (DBaaS) e computação sem servidor, as demandas de monitoramento tornaram-se mais diversas e específicas para cada aplicação. No entanto, as soluções atuais frequentemente carecem da granularidade e da adaptabilidade necessárias para gerenciar tarefas em nível de aplicação de forma eficaz em diferentes ambientes de nuvem. Além disso, os usuários normalmente têm autonomia limitada para definir ou ajustar os requisitos de monitoramento, incluindo a capacidade de priorizar métricas específicas ou ajustar limites com base no comportamento de suas aplicações. Essa inflexibilidade é particularmente problemática em cenários que envolvem arquiteturas baseadas em microsserviços, que exigem monitoramento dinâmico e detalhado. Portanto, embora as ferramentas existentes forneçam uma base, elas ainda não oferecem suporte abrangente às necessidades de aplicativos complexos, distribuídos e múltiplas nuvens.

Por fim, em comparação com os trabalhos relacionados apresentados na Tabela 4, esta pesquisa destaca-se principalmente por propor uma solução para o monitoramento de

QoS em múltiplas nuvens sob a perspectiva do usuário. Enquanto grande parte dos estudos concentra-se na validação dos valores de SLA ou na detecção de violações, muitas vezes por meio de abordagens baseadas em *blockchain* e aplicadas a ambientes de nuvem isolados, o presente trabalho avança além disso, integrando métricas provenientes de diferentes provedores e fornecendo uma visão unificada, orientada à análise contínua e à melhoria da experiência do usuário em relação a observação dos recursos contratados.

5 Monitoramento Cloud Sentinel

Neste capítulo, apresentamos uma arquitetura voltada ao monitoramento de aplicações distribuídas em ambientes de múltiplas nuvens, a qual incorpora a solução de monitoramento denominada *Cloud Sentinel*. A proposta contempla cenários compostos por aplicações baseadas no estilo arquitetural de microsserviços, implantadas em diferentes provedores de nuvem, sem que haja colaboração ou integração direta entre esses provedores. Essa característica configura um modelo de monitoramento *multi-cloud*, no qual a coleta e a análise de métricas ocorrem de forma independente em cada ambiente. As próximas seções descrevem os componentes e o funcionamento da arquitetura proposta.

5.1 Arquitetura Proposta

Nesta seção, apresentamos a arquitetura proposta para o monitoramento de ambientes *multi-cloud*, cujo propósito é apoiar arquitetos de *software* na tomada de decisões estratégicas, como a alocação de recursos, a reconfiguração de componentes ou a substituição de serviços, com base em métricas que vão desde o desempenho (por exemplo, cpu, memória, latência) até a disponibilidade e o custo operacional.

A arquitetura foi desenvolvida com base nos critérios definidos na taxonomia proposta (Seção 3.2) e, embora existam diversas ferramentas para o monitoramento de aplicações e infraestrutura, como, por exemplo, *New Relic*¹, *Datadog*², *Zabbix*³ e *Sentry*⁴, essas soluções geralmente exigem que o usuário realize uma análise manual e contínua das métricas coletadas para determinar se um serviço está atendendo aos requisitos definidos no SLA. Em outras palavras, tais ferramentas fornecem dados brutos e alertas de falhas, mas não oferecem, de forma direta e automatizada, uma visão consolidada de conformidade com os acordos de nível de serviço. Nesse contexto, a nossa arquitetura se diferencia por adotar uma abordagem centrada na avaliação do atendimento ao SLA, apresentando, de forma simplificada, se o serviço, em um determinado intervalo de tempo, está ou não em conformidade com os parâmetros estabelecidos pelo usuário. Essa característica reduz a necessidade de acompanhamento manual e permite uma tomada de decisão mais rápida e fundamentada pelo arquiteto de *software*.

A Figura 12 apresenta os principais componentes. O fluxo da arquitetura tem início com o componente voltado às configurações de monitoramento (coletor de requisitos) (etapa

¹ <<https://newrelic.com/>>

² <<https://www.datadoghq.com/>>

³ <<https://www.zabbix.com/>>

⁴ <<https://sentry.io/welcome/>>

1), no qual o arquiteto de *software* pode integrar seus microsserviços ao ecossistema de monitoramento e definir os requisitos, alinhados ao SLA e às características específicas da aplicação. Esses requisitos são então encaminhados para a **solução de monitoramento**, que chamamos de *cloud sentinel*. O *cloud sentinel* monitora os microsserviços da aplicação com base nas configurações definidas pelo usuário. O monitoramento ocorre em paralelo em cada microsserviço (etapa 2) da aplicação e é realizado por meio de um agente instalado em cada um. Esse agente baseia-se no uso de ferramentas de coleta de métricas de desempenho para obter informações sobre a saúde dos serviços.

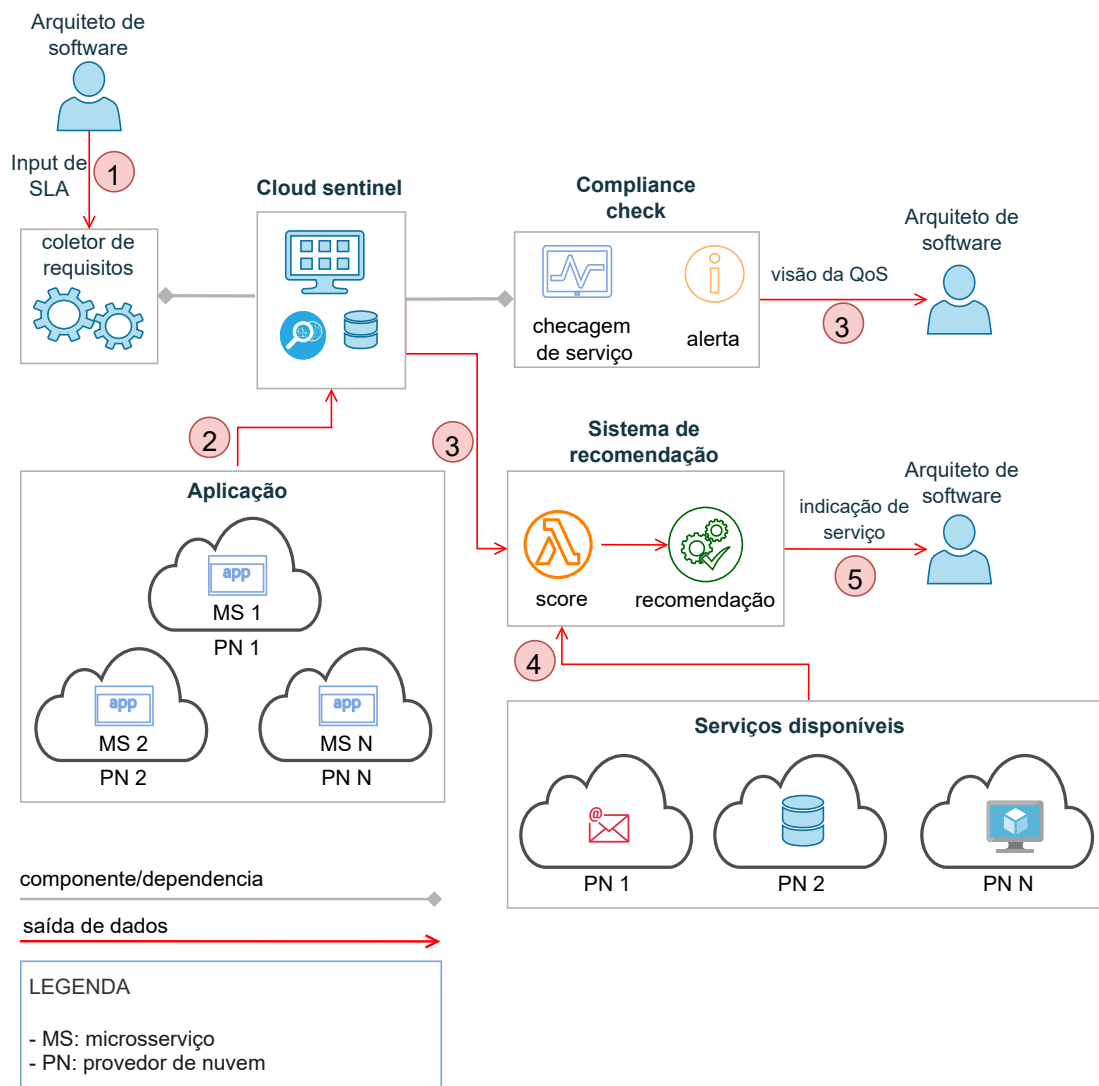


Figura 12 – Arquitetura proposta para monitoramento de aplicações em *multi-cloud*.

A arquitetura também inclui o componente de **serviços disponíveis** (etapa 4) que armazena dados sobre provedores de nuvem e serviços alternativos que podem ser utilizados em casos de migração de serviços. Além disso, a arquitetura conta com o **sistema de verificação de compliance** (etapa 3) (*compliance check*), um componente do *Cloud*

Sentinel voltado a verificar se os serviços contratados estão em conformidade com os requisitos definidos pelo usuário, bem como a apresentar relatórios detalhados sobre isso. Por fim, o **sistema de recomendação** (etapa 5) utiliza os dados do monitoramento e realiza um cruzamento com os serviços disponíveis, oferecendo recomendações sobre os melhores serviços a serem incorporados à aplicação.

5.2 Coletor de requisitos

O ponto de partida da arquitetura proposta consiste na capacidade do arquiteto de *software* de configurar, de forma centralizada, as métricas de monitoramento, alinhando-as aos SLAs de cada provedor e aos respectivos microsserviços que compõem a aplicação. Essa necessidade foi evidenciada na taxonomia apresentada na Seção 3.2.

A introdução de um componente dedicado à gestão de requisitos de monitoramento tem como objetivo oferecer maior autonomia ao arquiteto, permitindo que ele atualize definições, estabeleça prioridades e adapte os valores de avaliação dos serviços à evolução da aplicação. Tudo isso sem exigir alterações diretas nos microsserviços.

Neste contexto, o componente de configuração de métricas foi concebido para atuar de forma independente, garantindo também que sua operação não altere o contexto da solução de monitoramento. Desse modo, ele permite a centralização da lógica de avaliação, interpretando os dados coletados conforme os requisitos definidos pelo arquiteto. Para isso, conta com:

- uma *dashboard* administrativa, que permite ao arquiteto visualizar, ajustar e gerenciar de forma intuitiva os parâmetros e valores de monitoramento tendo como base o SLA;
- um repositório de acesso rápido, responsável por armazenar essas definições para aplicá-las durante o processo de análise.

5.3 Cloud Sentinel

Nesta seção, detalhamos o componente *Cloud Sentinel*, uma solução projetada para o monitoramento de serviços em ambientes *multi-cloud*. Na subseção 5.3.1, apresentamos sua arquitetura, descrevendo as funções e responsabilidades de cada módulo que a compõe. Em seguida, na subseção 5.3.2, discutimos a integração do *Cloud Sentinel* com a aplicação, apresentando uma visão macro de seu funcionamento no contexto do monitoramento.

5.3.1 Arquitetura do Cloud Sentinel

A Figura 13 ilustra o fluxo arquitetural do *cloud sentinel*, o qual pode ser resumido em 8 etapas. A primeira etapa conta com a integração do componente voltado à gestão de requisitos, na qual o usuário define parâmetros de monitoramento, como custo, tempo de resposta, disponibilidade, latência, consumo de CPU e de memória, sem se limitar a esses parâmetros. A arquitetura oferece suporte à coleta de valores de requisitos por meio de bibliotecas e ferramentas de coleta de métricas de desempenho, o que confere ao projetista autonomia para integrar soluções adicionais. Essa flexibilidade permite a incorporação de métricas complementares ou específicas, de modo a assegurar maior aderência ao contexto de cada aplicação.

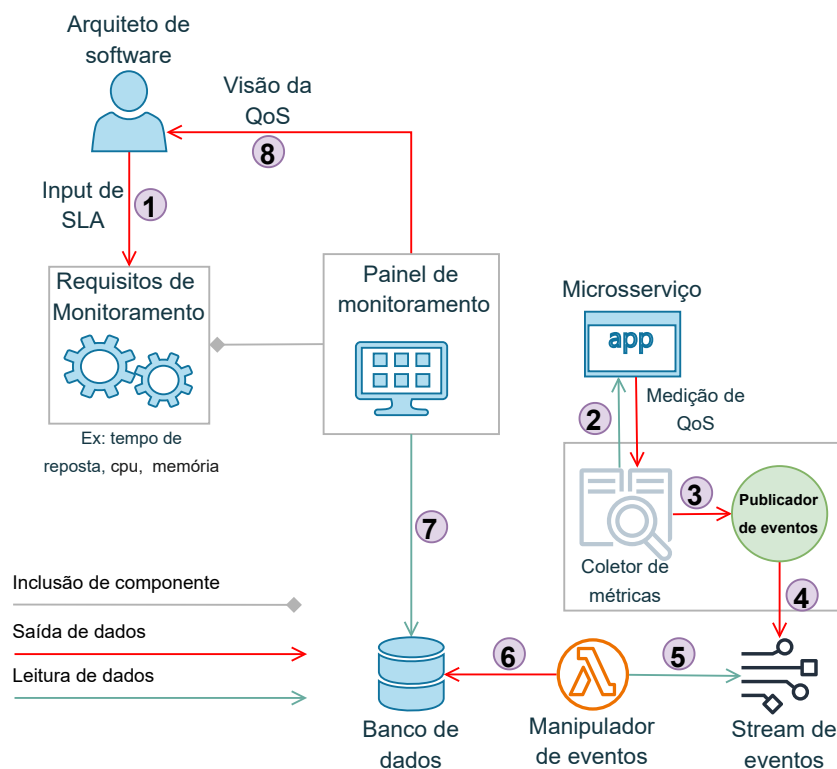


Figura 13 – Solução de monitoramento.

Além disso, a arquitetura permite especificar o intervalo de agregação dessas informações, determinando, por exemplo, se a consolidação dos dados ocorre a cada X minutos ou a cada Y horas. Essa flexibilidade permite ao usuário obter uma visão mais precisa e contextualizada do desempenho dos serviços ao longo do tempo, apoiando a tomada de decisão de forma mais fundamentada. Cabe ressaltar que o intervalo de agregação é configurável, uma vez que o objetivo da solução não é fornecer resultados de monitoramento em tempo real, mas sim análises consolidadas, com base em intervalos de tempo, para auxiliar na avaliação da conformidade dos serviços monitorados.

Os passos 2 a 4 correspondem ao agente de monitoramento, cuja implementação baseia-se em ferramentas para a coleta das métricas previamente configuradas. Esse agente é instalado diretamente no servidor responsável pela execução de cada microsserviço, permitindo-lhe interceptar requisições, identificar dependências e coletar dados de desempenho do microsserviço. Com base nas informações obtidas, na etapa 3 é iniciado um processo de publicação de eventos, fundamentado no paradigma de arquitetura orientada a eventos. Os eventos são publicados no momento em que ocorrem no microsserviço e incluem o *timestamp*, além dos dados de monitoramento, para preservar a cronologia nas análises e nos relatórios. Essa escolha deve-se ao fato de tal abordagem operar de forma assíncrona, evitando a sobrecarga no microsserviço e garantindo a integridade no tratamento dos dados. O publicador de eventos alimenta uma estrutura denominada *stream* de eventos (4), na qual as inserções podem ocorrer normalmente em paralelo, pois o próprio mecanismo da *stream* gerencia a concorrência nativamente.

Na etapa 5 do fluxo arquitetural, consta o manipulador de eventos, responsável por consumir os registros provenientes da *stream*(4) e processá-los para armazenamento. O objetivo é garantir que cada evento capturado pelo sistema de monitoramento seja persistido na base de dados (etapa 6), permitindo que seja analisado posteriormente no painel administrativo da solução (etapa 7).

Esse manipulador foi concebido como uma função *lambda* baseada em *serverless*, o que proporciona escalabilidade automática e otimização de custos, pois o processamento ocorre apenas sob demanda, em resposta à chegada de novos eventos. Essa abordagem reduz a sobrecarga operacional, especialmente em ambientes de múltiplas nuvens, onde o volume de eventos pode variar significativamente. Além disso, a função *lambda* e a *stream* de eventos foram projetadas para residirem no mesmo provedor de nuvem, de modo a minimizar problemas de latência na comunicação e evitar custos adicionais associados à transferência de dados entre provedores distintos. A escolha da estrutura de armazenamento é flexível, permitindo ao projetista optar por bancos de dados relacionais ou não relacionais, conforme os requisitos de consistência e escalabilidade da aplicação monitorada.

Com os dados devidamente coletados, armazenados e organizados, a arquitetura disponibiliza ao arquiteto uma interface de relatórios capaz de oferecer tanto uma visão geral quanto uma análise aprofundada dos serviços monitorados. Isso pode ser observado na etapa 8 da figura, na qual se vê a saída de relatórios que proporciona visibilidade sobre os serviços monitorados. A ideia é que o painel apresente relatórios simplificados, que atribuam estados aos serviços com base em métricas agregadas, e relatórios detalhados, nos quais as métricas históricas são apresentadas ao longo do tempo, possibilitando a identificação de tendências, padrões de degradação ou melhorias.

5.3.2 Integração entre o *Cloud Sentinel* e a Aplicação

O agente de monitoramento que faz parte da solução proposta adota uma abordagem na qual é incorporado diretamente à VM que hospeda o código de cada microsserviço. Essa integração permite a coleta de informações detalhadas sobre os serviços de que a aplicação depende. Seu funcionamento é pautado na adaptação dinâmica ao contexto operacional, ajustando o processo de monitoramento conforme a carga de trabalho — considerando tanto o volume quanto a frequência das requisições processadas.

A instalação e a inicialização do agente de monitoramento ocorrem simultaneamente à ativação do servidor do microsserviço, de forma análoga ao funcionamento de *middlewares*. Pois assim, garante-se que a coleta de métricas seja contínua, integrada ao ciclo de execução do sistema e capaz de fornecer dados precisos para análise e tomada de decisão. O algoritmo abaixo destaca a forma de integração do *Cloud Sentinel* a qual ocorre com a instalação do pacote, seguida da importação do módulo e logo após execução do módulo dentro do arquivo **main** da aplicação.

Algorithm 1 Instalação e Inicialização do Cloud Sentinel

```
1: procedure INSTALARCLOUDSENTINEL
2:   Baixar e instalar o pacote cloud-sentinel
3:   Importar módulo de integrações
4: end procedure

5: procedure INICIALIZARAPLICAÇÃO
6:   Executar patch_all_integrations() para ativar o monitoramento
7:   Iniciar ciclo de vida da aplicação
8: end procedure

9: procedure ENCERRARAPLICAÇÃO
10:  Finalizar coleta de métricas
11: end procedure
```

Nossa arquitetura propõe uma abordagem de monitoramento para aplicações baseadas em microsserviços distribuídos por múltiplos provedores de nuvem. Assim, consideramos um cenário em que a aplicação é composta por N microsserviços, mas, para minimizar problemas de latência, cada microsserviço, juntamente com suas dependências, como bancos de dados, por exemplo, deve ser alocado no mesmo provedor de nuvem. Acreditamos que essa medida assegura que, mesmo com a existência de N microsserviços, o monitoramento e a comunicação entre os serviços permaneçam otimizados, pois manter cada microsserviço e seus recursos associados no mesmo provedor reduz significativamente os atrasos na comunicação, melhorando o desempenho geral da aplicação distribuída.

A Figura 14 apresenta uma visão geral da integração entre o agente de monitoramento e a aplicação. No exemplo ilustrado, a aplicação é composta por dois microsserviços:

o primeiro é responsável pelo gerenciamento de notificações, e o segundo integra funcionalidades baseadas em serviços de Inteligência Artificial. No fluxo apresentado, o usuário define requisitos de monitoramento (passo 1), como tempo de resposta, uso de CPU e de memória, e conta com um painel administrativo para visualizar os resultados. Conforme evidenciado na figura, o agente de monitoramento atua de forma integrada à aplicação, sendo acoplado a cada microsserviço um agente dedicado (passo 2), o que possibilita a coleta e a análise individualizadas das métricas de desempenho.

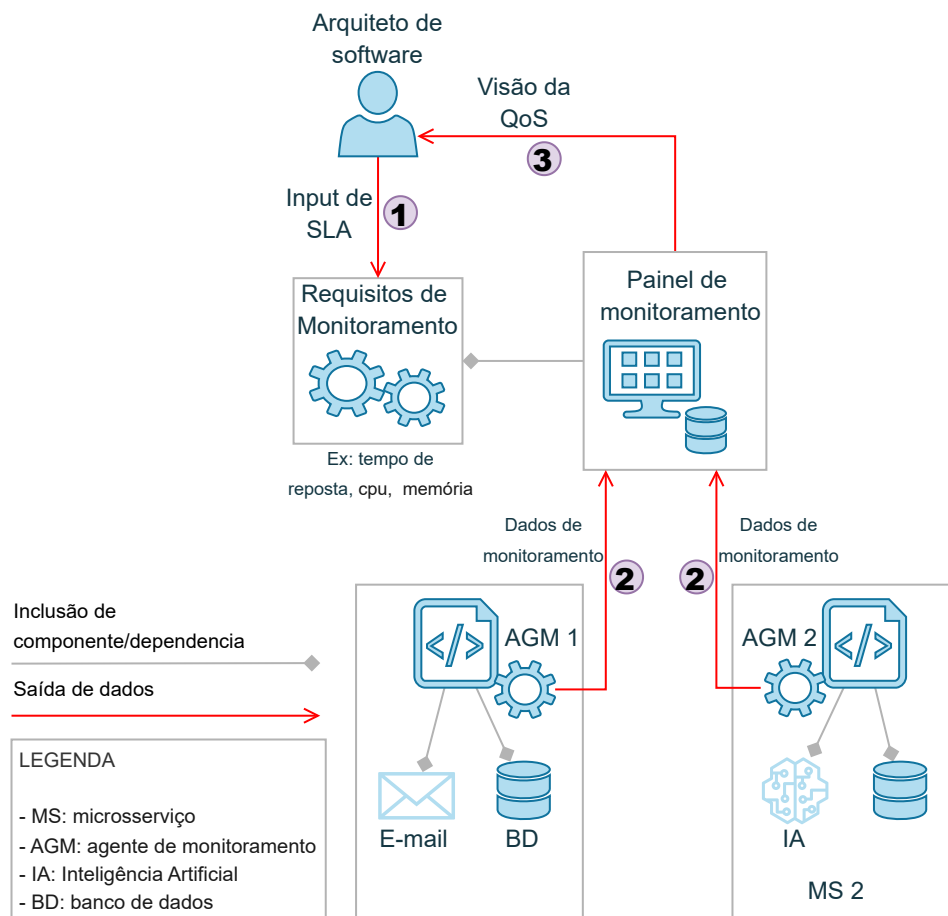


Figura 14 – Arquitetura da aplicação para Monitoramento de Múltiplas Nuvens.

5.4 Serviços disponíveis

A arquitetura proposta organiza os serviços dos provedores de nuvem em duas categorias: serviços em uso e disponíveis. Os serviços em uso correspondem aos recursos atualmente contratados pelo usuário e integrados aos microsserviços da aplicação, apresentando, portanto, dependências diretas para o seu funcionamento, por exemplo, de serviços de banco de dados, de mensageria ou de armazenamento. A seleção desses provedores pode

ter ocorrido com base em critérios técnicos e operacionais, como custo, tempo de resposta, disponibilidade ou ainda por preferências específicas do usuário, independentemente de métricas objetivas. Logo, a gestão ativa dos contratos e o monitoramento contínuo do desempenho desses serviços são elementos essenciais para garantir a qualidade e a eficiência.

Já os serviços disponíveis, por sua vez, englobam recursos oferecidos por outros provedores de nuvem que, embora não estejam atualmente em uso, podem ser considerados para futuras migrações ou expansões, a fim de alocar mais recursos à aplicação. Essa categoria envolve a análise comparativa de múltiplos atributos, como custo, disponibilidade, tempo de resposta e escalabilidade. Para isso, a arquitetura disponibiliza uma interface que permite ao usuário registrar as opções disponíveis e suas características relevantes, viabilizando a avaliação e comparação por meio do sistema de recomendação. Assim, o serviço de recomendação calcula *scores* para cada serviço e os compara para recomendar o que melhor atenda às necessidades da aplicação do usuário.

5.5 Compliance Check

O *Compliance Check* constitui um dos componentes centrais do *Cloud Sentinel*, tendo como principal atribuição apoiar o arquiteto de *software* no processo de tomada de decisão por meio da geração de relatórios claros e simplificados sobre o estado de conformidade dos serviços monitorados. Tal tarefa vai de encontro à principal vantagem desta solução: o monitoramento de múltiplas nuvens, que traz simplicidade e objetiva a avaliação do cumprimento do SLA. Assim, esse componente atua como uma camada analítica responsável por verificar, de forma contínua, se os serviços utilizados pela aplicação estão em conformidade com os requisitos previamente estabelecidos na fase de configuração do monitoramento.

A função do *Compliance Check* é antecipar o risco de descumprimento dos requisitos do SLA. Para isso, ele realiza uma análise sistemática e temporal das métricas coletadas pelo agente de monitoramento, buscando identificar tendências ou padrões que indiquem degradação de desempenho ou indisponibilidade parcial de algum serviço. Essa abordagem preditiva permite não apenas a detecção de anomalias, mas também a adoção de medidas preventivas antes que ocorram interrupções significativas ou violações de SLA.

A versão inicial do *Compliance Check* apresenta os resultados das análises em tabelas, com agregações em intervalos de tempo configuráveis pelo usuário. Essas visualizações têm como objetivo simplificar a interpretação dos dados, evidenciando o estado atual de cada serviço em relação aos parâmetros estabelecidos no SLA. Para tanto, foram definidas três categorias principais de avaliação: (i) serviço atendendo ao SLA; (ii) serviço atendendo parcialmente ao SLA; e (iii) serviço não atendendo a nenhum dos parâmetros definidos. Essa categorização fornece uma visão consolidada do comportamento dos

serviços, facilitando a priorização de ações corretivas.

Além disso, o *Compliance Check* mantém um histórico detalhado das métricas de desempenho e de disponibilidade de cada serviço monitorado, organizado por momento de coleta (ordem de *timestamp*). Esse registro histórico possibilita análises mais aprofundadas, permitindo correlações entre eventos de degradação e alterações no ambiente, bem como a avaliação da estabilidade dos provedores ao longo do tempo. O componente também oferece a visualização individualizada de métricas para cada dependência de microsserviço, possibilitando uma investigação direcionada de possíveis gargalos ou falhas.

Em conjunto, essas funcionalidades tornam o *Compliance Check* um elemento estratégico na arquitetura do *Cloud Sentinel*, promovendo uma visão integrada, preditiva e orientada a requisitos sobre a conformidade operacional de ambientes em múltiplas nuvens.

Para uma versão futura do *Compliance Check*, idealizamos a incorporação de modelos de aprendizado de máquina voltados à detecção antecipada de degradação de desempenho e de desvios operacionais. Por meio dessa abordagem, o componente poderá identificar indícios de anomalias antes que evoluam para violações contratuais ou interrupções críticas, viabilizando intervenções proativas e minimizando os impactos na aplicação.

A utilização de técnicas de aprendizado de máquina representa uma estratégia promissora para a análise de dados de monitoramento em ambientes distribuídos e dinâmicos. Modelos baseados em *machine learning* têm a capacidade de processar grandes volumes de informações heterogêneas, reconhecer padrões complexos e identificar tendências sutis que dificilmente seriam perceptíveis por métodos estatísticos tradicionais. Essa capacidade analítica avançada permite ao *Compliance Check* antecipar falhas potenciais, avaliar o risco de descumprimento de SLAs e gerar alertas preventivos, promovendo maior resiliência operacional.

Além disso, a aplicação de algoritmos de aprendizado de máquina automatiza etapas críticas do processo de monitoramento, reduzindo a dependência de intervenções manuais e ampliando a eficiência da análise contínua. Com isso, o componente não apenas aprimora a detecção de anomalias, mas também fornece *insights* estratégicos que auxiliam na tomada de decisão e no planejamento de ações corretivas, consolidando-se como uma ferramenta inteligente de apoio à governança de serviços em múltiplas nuvens.

5.6 Sistema de Recomendação

O sistema de recomendação proposto desempenha uma tarefa importante ao apoiar arquitetos de *software* na seleção de serviços que atendam aos requisitos funcionais e não funcionais das aplicações. A concepção desse componente baseia-se na integração

entre os dados provenientes do monitoramento contínuo e as informações sobre os serviços disponíveis em diferentes provedores de nuvem. Com base nessa correlação, o sistema tem como propósito identificar e sugerir alternativas que melhor atendam às necessidades especificadas, favorecendo decisões mais bem embasadas e alinhadas aos objetivos de desempenho, custo e disponibilidade da aplicação.

Embora o sistema de recomendação tenha sido idealizado no contexto da arquitetura proposta, sua implementação ainda não foi implementada nesta versão do *Cloud Sentinel*. Trata-se, portanto, de um componente em fase conceitual, cuja execução futura deverá contemplar mecanismos de análise comparativa e técnicas de priorização multicritério, a fim de oferecer recomendações mais precisas para o ambiente de monitoramento em múltiplas nuvens.

Além disso, a integração desse sistema com o componente *Compliance Check* permitirá que as recomendações sejam orientadas por evidências concretas de conformidade dos serviços monitorados. Dessa forma, o sistema de recomendação poderá identificar, de forma automática, serviços alternativos em casos de descumprimento do SLA, sugerindo migrações ou ajustes que assegurem o atendimento contínuo aos requisitos da aplicação. Essa relação entre os dois componentes reforça a capacidade do *Cloud Sentinel* de atuar de forma proativa, não apenas detectando violações de SLA, mas também propondo soluções para manter a qualidade e a confiabilidade do ambiente em múltiplas nuvens.

O sistema de recomendação opera em três etapas principais para auxiliar na seleção do serviço mais adequado às necessidades do usuário, são elas:

- **Processamento dos dados de monitoramento:** na etapa inicial, os dados são lidos e processados, resultando em uma pontuação (*score*) para cada serviço monitorado. Da mesma forma, pontuações são calculadas para os serviços disponíveis, com base nos critérios definidos pelo usuário, que podem incluir desempenho, custo e conformidade com o SLA, entre outros.
- **Comparação dos scores:** na segunda etapa, os *scores* dos serviços monitorados são comparados aos dos serviços disponíveis. Essa análise permite identificar alternativas que potencialmente oferecem melhor desempenho ou custo-benefício em relação ao serviço em uso;
- **Geração de relatório de recomendação:** por fim, na terceira etapa, o sistema gera um relatório detalhado com a recomendação do serviço mais adequado. O relatório indica se existe uma alternativa que atenda melhor às necessidades do usuário em comparação com o serviço atual, evidenciando o benefício de uma migração.

6 Experimentos e Resultados

Nesta seção, apresentamos os resultados obtidos a partir do processo de validação da arquitetura proposta no Capítulo 5. Para a validação, desenvolvemos o *Cloud Sentinel*, detalhado na Seção 5.3. Nas seções seguintes, descrevemos as configurações adotadas para a validação e discutimos os resultados obtidos com base nos cenários de teste.

6.1 Ferramenta

Para validar a arquitetura proposta, desenvolvemos uma ferramenta, apresentada na Figura 15.

O processo de validação foi realizado com base no desenvolvimento de uma aplicação composta por dois microsserviços. Onde esses microsserviços representam o contexto de catálogo e os produtos de uma loja fictícia. A aplicação foi implementada na linguagem *Python*, utilizando o micro *framework* *FastAPI*¹ e o *Serverless Framework*² para o desenvolvimento do componente, com arquitetura orientada a eventos. A implantação dos microsserviços foi realizada em dois provedores de nuvem pública: *Amazon Web Services* (AWS)³, caracterizado como um provedor de infraestrutura como serviço (IaaS), e *Heroku*⁴, que opera no modelo de plataforma como serviço (PaaS). A AWS oferece ao arquiteto maior controle e flexibilidade sobre a infraestrutura, exigindo que ele defina detalhadamente como a aplicação será implantada, desde a configuração de máquinas virtuais, a escolha e a parametrização de bancos de dados até políticas de segurança e de escalabilidade. O Heroku adota uma proposta distinta, baseada na abstração e na simplicidade. Nesse modelo, a complexidade da infraestrutura é totalmente gerenciada pela própria plataforma, de modo que o arquiteto precisa apenas seguir comandos e instruções pré-definidos para colocar a aplicação no ar. Assim, a principal diferença está no grau de autonomia e responsabilidade: na AWS, o arquiteto tem mais poder, porém, mais deveres; no Heroku, a plataforma assume a gestão para privilegiar agilidade e praticidade no *deploy*.

O fluxo implementado segue o proposto na solução descrita na Seção 5.3, contemplando tanto os mecanismos de monitoramento quanto os elementos de integração entre os microsserviços. De forma resumida, a solução foi composta por três elementos principais: (i) painel de monitoramento, (ii) microsserviços de aplicação e (iii) infraestrutura de eventos, cujos detalhes são descritos a seguir.

¹ <<https://fastapi.tiangolo.com/>>

² <<https://www.serverless.com/>>

³ <<https://aws.amazon.com/>>

⁴ <<https://www.heroku.com/>>

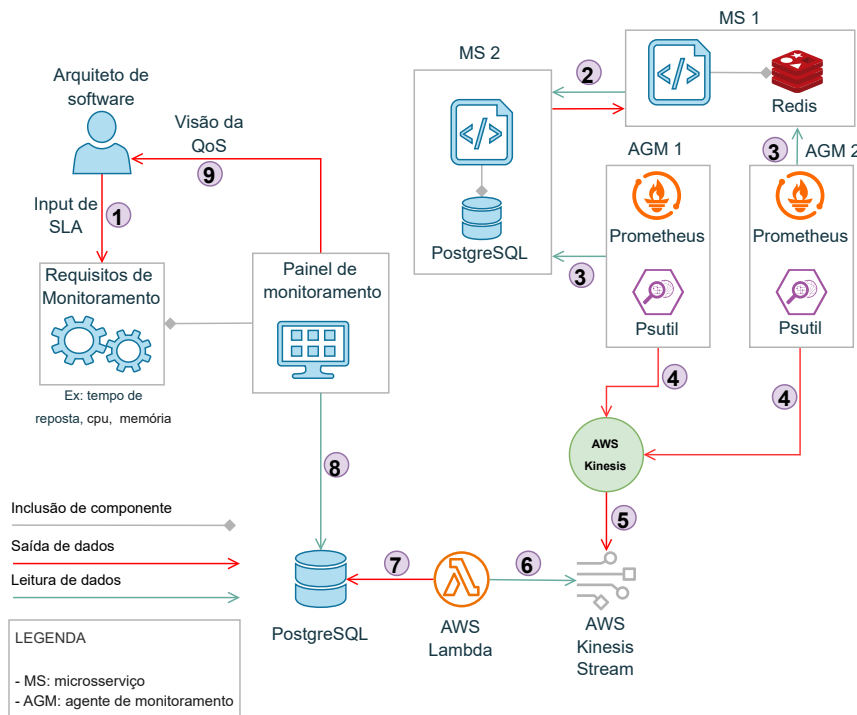


Figura 15 – Ilustração da configuração experimental para validação da solução de monitoramento em multi-cloud.

O painel de monitoramento foi desenvolvido em *Python* com *FastAPI* e integrado a um banco de dados relacional *PostgreSQL*. Esse painel permite ao usuário configurar os limiares das métricas a serem monitoradas; no caso desta validação, consideramos **tempo de resposta, consumo de memória, uso de CPU e disponibilidade**. Essa funcionalidade é evidenciada na Figura 16. Além disso, o painel fornece acesso a relatórios que consolidam informações sobre o desempenho e a confiabilidade das dependências da aplicação. Para fins de validação, todo o painel foi hospedado no *Heroku*, explorando sua facilidade de implantação e de manutenção.

show 10 entries

Métrica: Search for metric

ID	MÉTRICA	MIN_THRESHOLD	MAX_THRESHOLD	IS_ACTIVE	CREATED_AT
1	Disponibilidade	98.0	100.0	true	2025-08-05 17:25:34
2	Tempo de Resposta	0.0	30.0	true	2025-08-05 17:25:46
4	Consumo de CPU	0.0	75.0	true	2025-08-05 17:26:14
3	Consumo de Memória	0.0	75.0	true	2025-08-05 17:26:00

Figura 16 – Ilustração da funcionalidade de definição de SLA para métricas de monitoramento.

O primeiro microsserviço (MC 01) foi responsável pelo gerenciamento de produtos, simulando uma API de armazenamento com persistência em um banco de dados *PostgreSQL*. Toda a sua infraestrutura foi hospedada na AWS, explorando seus recursos de IaaS para um provisionamento e controle mais granulares dos serviços. O segundo microsserviço (MC 02) foi projetado para atuar como um catálogo de produtos de uma loja virtual. Esse microsserviço, hospedado no *Heroku*, utilizou um banco de dados não relacional, o Redis, para armazenar cache de produtos, reduzindo o número de chamadas diretas ao Microsserviço 01 e otimizando o tempo de resposta. A comunicação entre os dois microsserviços foi realizada por meio de requisições HTTP.

Ambos os microsserviços foram instrumentados com o agente de monitoramento proposto nesta pesquisa, desenvolvido por meio da integração do *Prometheus* à biblioteca *Psutil* para a coleta de métricas. Esse agente coletou métricas conforme a carga e as encaminhou via *AWS Kinesis* para a *Stream* de eventos. É válido destacar que cada microsserviço possui um agente.

A infraestrutura de eventos foi apoiada pelos serviços da AWS. O *Kinesis* foi empregado para a publicação dos eventos de monitoramento, enquanto o *Kinesis Stream* atuou como camada de armazenamento temporário desses dados. Para processar e transformar os eventos, bem como inseri-los na base de dados do painel, foi desenvolvido um componente baseado em *AWS Lambda*, implementado com o *Serverless Framework*. Essa abordagem permitiu a gestão e a manipulação dos eventos, assegurando a escalabilidade e a confiabilidade no fluxo de monitoramento. Em conjunto, tais componentes possibilitaram a avaliação prática da arquitetura proposta, demonstrando sua viabilidade em um ambiente de nuvem distribuído e heterogêneo.

Por fim, as configurações apresentadas na Tabela 6 correspondem às dependências utilizadas durante a execução dos experimentos com o *Cloud Sentinel*. Esses serviços foram implantados em provedores distintos — Heroku e AWS — utilizando planos básicos pagos, de modo a representar ambientes de produção com recursos limitados, porém realistas. As máquinas virtuais, bancos de dados e serviços de cache foram configurados conforme os parâmetros de CPU, memória e armazenamento descritos, garantindo a consistência na comparação de desempenho e de comportamento da solução em diferentes infraestruturas de nuvem.

Tabela 6 – Configurações dos serviços para realização de experimentos.

Serviço	Provedor	CPU	Memória	Armazenamento
VM	Heroku	1 vCPU	512 MB RAM	1 GB
VM	AWS	2 vCPU	1 GB RAM	8 GB
PostgreSQL	Heroku	1 vCPU	1 GB RAM	1 GB
PostgreSQL	AWS	2 vCPU	1 GB RAM	20 GB
Redis	Heroku	1 vCPU	25 MB RAM	25 MB

6.2 Experimentos

Nesta seção, apresentamos, em mais detalhes, os experimentos conduzidos para avaliar o comportamento da solução em relação ao objetivo de verificação de conformidade dos serviços de uma aplicação distribuída e não distribuída em múltiplas nuvens. Para isso, foram idealizados três cenários distintos, de modo a contemplar diferentes situações de operação e permitir uma análise abrangente do desempenho da aplicação.

Os cenários foram definidos e consideram três classes representativas do estado atual dos serviços, em comparação ao SLA cadastrado pelo usuário. Segue a definição de cada classe, que representa os cenários 1, 2 e 3, respectivamente.

- **SLA atendido:** refere-se ao estado em que o serviço opera de forma estável, sem interrupções ou perda de desempenho, atendendo aos requisitos estabelecidos.
- **SLA parcialmente atendido:** corresponde a um estado intermediário, no qual o serviço permanece operacional, mas não satisfaz todos os parâmetros definidos pelo usuário;
- **SLA não atendido:** representa a condição em que o serviço não cumpre os valores mínimos estabelecidos nos requisitos do usuário, caracterizando que não está em conformidade com a necessidade do mesmo;

Conforme destacado na seção anterior sobre a configuração dos experimentos, foram consideradas quatro métricas principais para validação. Os respectivos valores de SLA definidos no *Cloud Sentinel* estão descritos a seguir:

- **Disponibilidade:** mínimo de 98%;
- **Consumo de CPU:** máximo 75%;
- **Consumo de Memória:** máximo 75%;
- **Tempo de resposta:** máximo 30 segundos.

Antes de apresentar os resultados, é relevante ressaltar que o sistema é capaz de identificar automaticamente as dependências de cada serviço monitorado. A Figura 17 ilustra esse processo, evidenciando, por exemplo, que o microsserviço de Catálogo possui como dependências um banco de dados *Redis*, uma máquina virtual e o microsserviço de Produtos, em razão da comunicação entre eles. Por sua vez, o microsserviço de Produtos depende do banco de dados *PostgreSQL* e de uma máquina virtual. Além desse mapeamento, o sistema também registra informações detalhadas, como o endereço e a porta de cada dependência identificada, fornecendo uma visão abrangente da arquitetura em execução.

NAME	APPLICATION NAME	ADDRESS	PORT
3.94.204.88	API Catalog	3.94.204.88	8000
postgres	API Products	database-1.cwhsyas2fbf.us-east-1.rds.amazonaws.com:5432	5432
redis	API Catalog	ec2-52-205-107-90.compute-1.amazonaws.com	30780
vm	API Catalog	8df40e76-53d3-44c4-89e5-4cb03e134baa	0
vm	API Products	ip-172-31-42-78	0

Figura 17 – Ilustração da página de dependências coletadas pelo Cloud Sentinel.

Para a avaliação do *Cloud Sentinel*, foram definidos três cenários experimentais distintos, cada um projetado para representar diferentes condições operacionais e níveis de conformidade dos serviços monitorados. Todos os experimentos tiveram duração de uma hora, com consultas realizadas a cada seis segundos, o que equivale a aproximadamente dez requisições por minuto. Esses parâmetros foram mantidos constantes em todos os cenários, permitindo uma comparação equilibrada dos resultados.

No Cenário 1, o experimento foi executado em múltiplos provedores de nuvem, com todos os serviços operando dentro dos limites estabelecidos pelo SLA. Nesse contexto, o objetivo foi avaliar o comportamento da solução em condições normais de operação e verificar a capacidade do sistema de coletar e consolidar métricas de diferentes ambientes sem registrar falsos positivos de não conformidade. Todas as requisições partiram do microserviço de Catálogo, que atuou como ponto de origem para as comunicações com suas dependências, incluindo o banco de dados *Redis*, o microserviço de Produtos e a máquina virtual associada.

O Cenário 2 também foi realizado em múltiplos provedores, mantendo a mesma configuração temporal e a mesma frequência de consultas. No entanto, foram introduzidas anomalias deliberadas em alguns serviços, especificamente no *Redis* e no *PostgreSQL*, com o intuito de simular situações de degradação de desempenho e períodos de indisponibilidade. Esse cenário teve como propósito avaliar a capacidade do *Cloud Sentinel* de detectar falhas reais e de classificar corretamente o estado dos serviços em relação ao SLA definido pelo usuário.

Por fim, o Cenário 3 foi conduzido com um único provedor de nuvem, com duração de uma hora e requisições a cada seis segundos. Nesse caso, todos os serviços permaneceram em operação normal, sem falhas ou anomalias. O objetivo foi comparar o comportamento da solução em um ambiente mais restrito, no qual não há comunicação entre provedores distintos, permitindo observar eventuais diferenças no tempo de resposta, na disponibilidade e na eficiência do monitoramento.

A Tabela 7 apresenta um resumo dos três cenários, destacando suas principais características e objetivos experimentais.

Tabela 7 – Cenários de teste realizados.

Cenário	Duração	Intervalo de requisições	Provedores	Descrição do teste
01	1 hora	6 segundos	AWS e Heroku	Serviços operando normalmente
02	1 hora	6 segundos	AWS e Heroku	Serviços com indisponibilidade
03	1 hora	6 segundos	Heroku	Serviços operando normalmente

Nas subseções a seguir, apresentamos e discutimos os resultados obtidos para cada um dos cenários acima utilizando a ferramenta desenvolvida e descrita na Seção 6.1.

6.3 Resultados para o Cenário 01

Nesta seção, apresentamos os resultados do cenário 01, no qual o objetivo foi avaliar o comportamento dos serviços em condições normais de operação, verificando sua conformidade com os parâmetros de SLA previamente estabelecidos. Os resultados demonstram que o *Cloud Sentinel* foi eficiente em classificar corretamente o estado dos serviços, assegurando que todos permaneceram estáveis ao longo do experimento e atenderam integralmente aos requisitos definidos, conforme ilustrado na Figura 18.

MICROSSERVICE	DEPENDENCE	STATUS
API Catalog	3.94.204.88	SLA atendido
API Products	postgres	SLA atendido
API Catalog	redis	SLA atendido
API Products	vm	SLA atendido
API Catalog	vm	SLA atendido

Figura 18 – Classes de serviços de acordo com o Cloud Sentinel.

Além do relatório simplificado, que apresenta o estado atual dos serviços, o *Cloud Sentinel* também disponibiliza um relatório geral que reúne todo o histórico de execução (Figura 19). Esse relatório reúne informações detalhadas, como os *timestamps* de cada coleta, a vazão registrada ao longo do tempo e as demais métricas monitoradas. Esse nível de detalhamento possibilita uma análise mais aprofundada do comportamento dos serviços, favorecendo a identificação de tendências e padrões de desempenho. É importante ressaltar que, embora os dados tenham sido agrupados para todo o período de execução apresentado, o sistema permite visualizar cada registro individualmente, bem como ajustar a granularidade temporal conforme a necessidade da análise.

Como complemento, o sistema disponibiliza uma visão gráfica das métricas individuais de cada serviço, recurso que facilita tanto a interpretação dos resultados quanto a análise de tendências ao longo do tempo. Tomando o Redis como exemplo, a Figura 20 apresenta essa visualização detalhada, conforme destacado na seleção em verde. Observa-se

MICROSSERVICE	DEPENDENCE	AVAILABILITY	RESPONSE TIME	THROUGHPUT	CPU	MEM	TIMESTAMP
API Catalog	vm	100%	0.100	2	8.8%	42.0%	2025-08-23 00:00
API Products	vm	100%	0.100	2	0.0%	53.1%	2025-08-23 00:00
API Catalog	3.94.204.88	100%	0.520	422	8.4%	45.1%	2025-08-22 18:00
API Catalog	redis	100%	0.003	844	9.5%	45.1%	2025-08-22 18:00
API Catalog	vm	100%	0.101	422	12.1%	45.1%	2025-08-22 18:00
API Products	postgres	100%	0.005	422	3.8%	50.0%	2025-08-22 18:00
API Products	vm	100%	0.101	425	0.2%	50.0%	2025-08-22 18:00

Figura 19 – Relatório de serviços conforme o Cloud Sentinel.

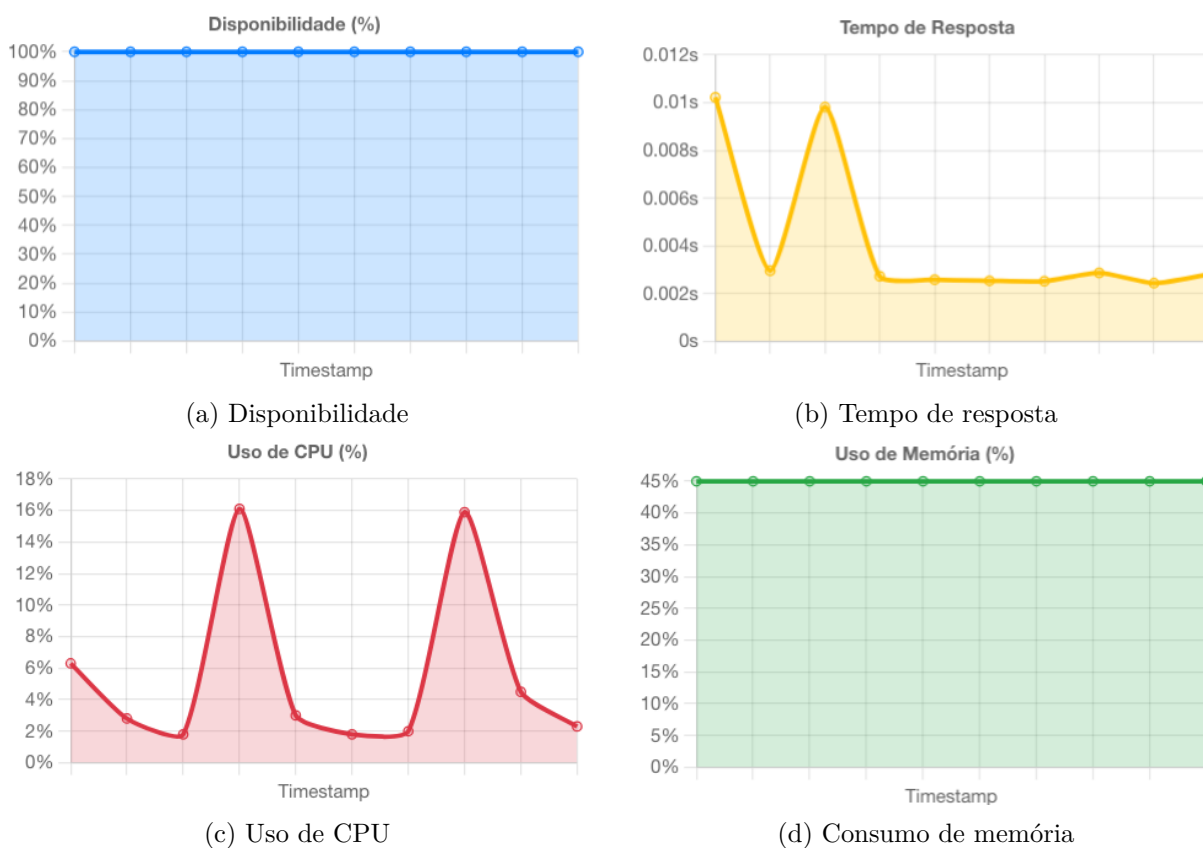


Figura 20 – Métricas monitoradas no Cenário 01 do Redis.

um comportamento estável ao longo de todo o período de testes. A Figura 20a evidencia que a disponibilidade se manteve em 100% em todas as requisições, enquanto a Figura 20d mostra um uso equilibrado da memória. A Figura 20c indica baixa variabilidade no consumo de CPU; embora tenham ocorrido dois picos, estes representaram apenas 16% de utilização e foram considerados irrelevantes. De forma semelhante, o tempo de resposta apresentou pequenas variações, sem impactar a operação. Assim como poder ser observado na Figura 20b Esses resultados reforçam a robustez do serviço em um cenário de operação contínua.

6.4 Resultados para o Cenário 02

Nesta seção, apresentamos os resultados do cenário 02. Neste cenário, objetivamos avaliar a capacidade do *Cloud Sentinel* de identificar o atendimento parcial e o descumprimento do SLA por dependência, ou seja, situações em que algumas dependências do microsserviço não cumprem integralmente os requisitos estabelecidos pelo arquiteto. No microsserviço de Catálogo, o Redis apresentou métricas ligeiramente acima do limite, conforme destacado em amarelo na Figura 22, o que resultou em atendimento parcial ao SLA, enquanto a máquina virtual associada manteve a operação dentro dos parâmetros exigidos. No microsserviço de produtos, o *PostgreSQL* não atendeu a nenhum dos critérios de SLA, conforme destacado em vermelho na Figura 22, o que caracteriza um não atendimento. Porém, a máquina virtual permaneceu operacional. A Figura 21, referente ao relatório simplificado, evidencia que o sistema foi eficiente em identificar e diferenciar essas situações. A análise mostra que, apesar de algumas dependências não terem cumprido o SLA, o atendimento do microsserviço como um todo pode ser classificado como parcial, uma vez que nem todas as dependências estavam fora de conformidade. Com base nesses resultados, o arquiteto pode optar por aumentar os recursos destinados a esses serviços ou, se necessário, considerar sua substituição.

MICROSSERVICE	DEPENDENCE	STATUS
API Catalog	3.94.204.88	SLA atendido
API Products	postgres	SLA não atendido
API Catalog	redis	SLA parcialmente atendido
API Catalog	vm	SLA atendido
API Products	vm	SLA atendido

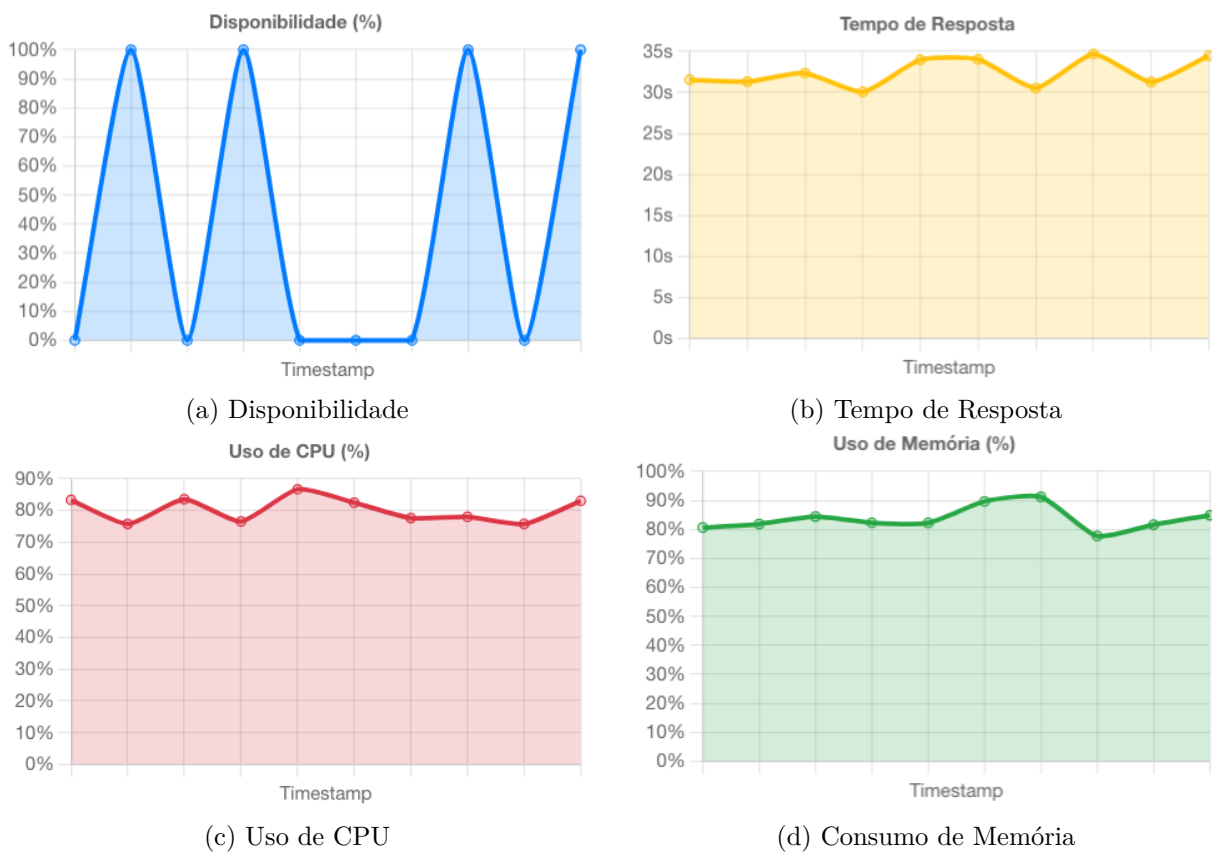
Figura 21 – Classes de serviços para o cenário 02 de acordo com o Cloud Sentinel.

A Figura 22 apresenta uma visão agregada das métricas, na qual se observa que o Redis deixou de atender ao SLA apenas na métrica de consumo de memória, além de apresentar tendência de crescimento no tempo de resposta ao longo do teste. Já o *PostgreSQL* não atendeu ao SLA em nenhuma das métricas avaliadas, evidenciando inclusive possíveis exceções e interrupções no serviço, uma vez que sua disponibilidade registrada foi de apenas 56%

Por fim, a Figura 23 apresenta a evolução temporal das métricas do *PostgreSQL*, utilizada como exemplo para demonsA Figura 23b mostra que o tempo de resposta

MICROSSERVICE	DEPENDENCE	AVAILABILITY	RESPONSE TIME	THROUGHPUT	CPU	MEM	TIMESTAMP
API Products	vm	100%	0.100	4	0.0%	52.3%	2025-08-23 21:00
API Products	vm	100%	0.101	11	0.0%	51.4%	2025-08-23 18:00
API Catalog	3.94.204.88	100%	0.525	349	15.2%	36.9%	2025-08-23 15:00
API Catalog	redis	100%	25.051	698	54.8%	77.4%	2025-08-23 15:00
API Catalog	vm	100%	0.101	349	16.4%	36.9%	2025-08-23 15:00
API Products	postgres	56%	32.556	349	81.4%	84.8%	2025-08-23 15:00
API Products	vm	100%	0.101	349	0.4%	50.6%	2025-08-23 15:00

Figura 22 – Relatório para o cenário 02 de acordo com o Cloud Sentinel.

Figura 23 – Métricas monitoradas no Cenário 02 para o *PostgreSQL*.

permaneceu constante, com valores iguais ou superiores a 30 segundos, o que não atende aos requisitos do SLA no serviço ou de exceções internas, evidenciadas pelos registros de indisponibilidade. A Figura 23b mostra que o tempo de resposta permaneceu constante, com valores iguais ou superiores a 30 segundos, não atendendo aos requisitos do SLA. As Figuras 23c e 23d revelam que, ao longo de todo o experimento, o consumo de CPU e de memória manteve-se, na maior parte do tempo, acima do limite máximo de 75% definido no SLA. De modo geral, a constância e os valores observados nas visualizações gráficas indicam ao arquiteto que, com a configuração atual, o *PostgreSQL* não atende à carga de

trabalho exigida pela aplicação ou pelo microsserviço.

6.5 Resultados para o Cenário 03

Nesta seção, são apresentados e discutidos os resultados obtidos no Cenário 03. Nesse experimento, todos os microsserviços foram configurados e implantados na plataforma Heroku, com o objetivo de avaliar o comportamento do sistema em um ambiente não distribuído e compará-lo ao cenário distribuído, previamente executado na AWS e na Heroku. Essa comparação permitiu analisar o impacto da infraestrutura subjacente nas métricas de desempenho monitoradas pelo *Cloud Sentinel*, incluindo tempo de resposta, disponibilidade, consumo de CPU e uso de memória.

A Figura 24 apresenta uma visão comparativa dos dois ambientes de implantação. Observa-se que, de modo geral, o cenário distribuído obteve resultados superiores em praticamente todos os componentes monitorados — Redis e PostgreSQL, bem como na máquina virtual —, evidenciando tempos de resposta mais baixos e menor uso de recursos (cpu e memória).

Comparando os resultados apresentados nas Figuras 24a e 24b, nota-se uma redução do tempo médio de resposta e uma utilização mais equilibrada dos recursos de CPU e de memória no ambiente distribuído. Esse ganho de desempenho pode estar relacionado às diferenças de configuração entre as infraestruturas utilizadas. Conforme demonstrado na Tabela 6, os serviços provisionados na AWS apresentam especificações superiores às oferecidas pelo Heroku em seus planos básicos, tanto em processamento quanto em armazenamento. Além disso, a AWS oferece mecanismos de alocação dinâmica de recursos, que contribuem para um comportamento mais eficiente em cenários de carga variável.

Esses resultados reforçam a importância de ferramentas de monitoramento, como o *Cloud Sentinel*, no processo de avaliação comparativa e de tomada de decisão. O sistema demonstrou ser capaz de coletar e consolidar métricas, possibilitando uma análise aprofundada do desempenho de cada componente e de suas interdependências. Dessa forma, o *Cloud Sentinel* se evidencia como um instrumento estratégico para a gestão adaptativa de recursos em ambientes de computação em nuvem, permitindo que o arquiteto de sistemas identifique gargalos, otimize custos e defina políticas de alocação com maior precisão e baseadas em evidências.

MICROSSERVICE	DEPENDENCE	AVAILABILITY	RESPONSE TIME	THROUGHPUT	CPU	MEM	TIMESTAMP
API Catalog	vm	100%	0.100	2	8.8%	42.0%	2025-08-23 00:00
API Products	vm	100%	0.100	2	0.0%	53.1%	2025-08-23 00:00
API Catalog	3.94.204.88	100%	0.520	422	8.4%	45.1%	2025-08-22 18:00
API Catalog	redis	100%	0.003	844	9.5%	45.1%	2025-08-22 18:00
API Catalog	vm	100%	0.101	422	12.1%	45.1%	2025-08-22 18:00
API Products	postgres	100%	0.005	422	3.8%	50.0%	2025-08-22 18:00
API Products	vm	100%	0.101	425	0.2%	50.0%	2025-08-22 18:00

(a) Relatório de serviços em ambiente distribuído (AWS e Heroku).

MICROSSERVICE	DEPENDENCE	AVAILABILITY	RESPONSE TIME	THROUGHPUT	CPU	MEM	TIMESTAMP
API Catalog	vm	100%	0.101	2	62.6%	46.0%	2025-11-04 20:00
APP Products	vm	100%	0.101	2	18.2%	41.2%	2025-11-04 10:00
API Catalog	app-products-c3ecce81575b.herokuapp.com	100%	0.544	265	29.3%	59.8%	2025-11-03 20:00
API Catalog	redis	100%	0.004	530	30.9%	59.8%	2025-11-03 20:00
API Catalog	vm	100%	0.101	265	35.2%	59.8%	2025-11-03 20:00
APP Products	postgres	100%	0.016	265	14.9%	61.6%	2025-11-03 20:00
APP Products	vm	100%	0.100	265	13.6%	61.6%	2025-11-03 20:00

(b) Relatório de serviços em ambiente não distribuído (Heroku).

Figura 24 – Comparativo de desempenho de serviços em ambiente distribuído e não distribuído.

7 Considerações finais e Trabalhos Futuros

O objetivo principal deste trabalho foi propor uma solução de monitoramento para ambientes de múltiplas nuvens, considerando métricas de QoS definidas em acordos de SLA, sob a perspectiva de um arquiteto de *software*. Inicialmente, apresentamos o contexto e o *background* relacionados ao monitoramento em múltiplas nuvens. Em seguida, introduzimos a nossa taxonomia (Capítulo 3) para monitoramento *multi-cloud*, seguida da descrição de nossa arquitetura (Capítulo 5) específica para esse tipo de monitoramento. Por fim, desenvolvemos uma ferramenta e conduzimos a validação da arquitetura proposta, demonstrando sua aplicabilidade.

A Taxonomia proposta serviu como base para análise das soluções de monitoramento selecionadas. As investigações em trabalhos relacionados destacaram a necessidade do desenvolvimento de mais pesquisas para o desenvolvimento de soluções de monitoramento de nuvem que atentem para os seguintes fatores: monitorem múltiplas nuvens; sejam mais abrangentes em relação ao arquiteto de *software* e a aplicação; tenham mais independência em relação ao modelo de serviço; permita mais flexibilidade para o usuário monitorar componentes menores; por fim, considerem o gerenciamento dos requerimentos a serem monitorados.

Os experimentos realizados demonstraram que a solução desenvolvida para o monitoramento em ambientes *multi-cloud* mostrou-se capaz de cumprir os objetivos propostos. A ferramenta oferece flexibilidade para que o usuário personalize os parâmetros de SLA de acordo com suas necessidades, além de disponibilizar relatórios simplificados que indicam de forma objetiva se os serviços estão em conformidade com o SLA. Adicionalmente, foi possível verificar sua aplicabilidade tanto em ambientes IaaS quanto em PaaS, evidenciando a versatilidade da solução no monitoramento de diferentes modelos de serviço da computação em nuvem. Outro diferencial é o registro histórico das métricas coletadas, complementado por visualizações por meio de gráficos que auxiliam o arquiteto de *software* na identificação de picos/anomalias e tendências de desempenho. Dessa forma, a solução proposta não apenas apoia a verificação da conformidade dos serviços, mas também se mostra relevante como instrumento de apoio à tomada de decisão, permitindo ao arquiteto avaliar, de forma proativa, a necessidade de escalar recursos ou substituir serviços para garantir a continuidade e a qualidade da aplicação.

Como trabalhos futuros, pretendemos aprofundar as pesquisas no contexto do monitoramento de ambientes *multi-cloud*. Assim, planejamos o aprimoramento dos relató-

rios da solução, tornando-os mais robustos e capazes de indicar tendências por meio da aplicação de algoritmos de aprendizado de máquina. Além disso, visamos à implementação e validação de um sistema de recomendação de serviços, ampliando o apoio à tomada de decisão do arquiteto de *software*. Também pretendemos integrar outras ferramentas de coleta de métricas de desempenho para fins de comparação ao *Prometheus* e *Psutil*. Por fim, almejamos integrar a solução desenvolvida ao *PacificClouds* (CARVALHO, 2019), uma arquitetura voltada ao gerenciamento da implantação e execução de aplicações baseadas em microsserviços distribuídos em ambientes *multi-cloud*, fortalecendo ainda mais a aplicabilidade da proposta.

Referências

- ABDULLAH, M.; SURPUTHEEN, M. M. Cadram-cooperative agents dynamic resource allocation and monitoring in cloud computing. *International Journal of Computer Science & Network Security*, International Journal of Computer Science & Network Security, v. 22, n. 3, p. 95–100, 2022. Citado 4 vezes nas páginas 25, 26, 30 e 31.
- ABHISHEK, P.; CHOBARI, A.; NARAYAN, D. Sla violation detection in multi-cloud environment using hyperledger fabric blockchain. In: IEEE. *2021 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*. [S.l.], 2021. p. 107–112. Citado 3 vezes nas páginas 25, 26 e 31.
- ADETIBA, E. et al. Fedmonitor: An implementation architecture for monitoring of resources and events' notifications on federated cloud computing infrastructures. In: IEEE. *2024 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*. [S.l.], 2024. p. 1–6. Citado 3 vezes nas páginas 25, 26 e 31.
- AFZAL, B. et al. *Service level agreement monitoring as a service: an independent monitoring service for service level agreements in clouds (2022)*. 2022. Citado 3 vezes nas páginas 25, 26 e 31.
- ALONSO, J.; HUARTE, M.; ARRIETA, L. O.-E. Acsmi: a solution to address the challenges of cloud services federation and monitoring towards the cloud continuum. *International Journal of Computational Science and Engineering*, Inderscience Publishers (IEL), v. 26, n. 5, p. 537–554, 2023. Citado 3 vezes nas páginas 25, 26 e 31.
- ANITHAKUMARI, S.; CHANDRASEKARAN, K. Monitoring and management of service level agreements in cloud computing. In: IEEE. *2015 International Conference on Cloud and Autonomic Computing*. [S.l.], 2015. p. 204–207. Citado na página 17.
- BELOGLAZOV, A.; BUYYA, R. Energy efficient resource management in virtualized cloud data centers. In: IEEE. *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*. [S.l.], 2010. p. 826–831. Citado na página 2.
- BIRJE, M. N.; BULLA, C. Cloud monitoring system: Basics phases and challenges. *Int. J. Recent Technol. Eng.*, v. 8, n. 3, p. 4732–4746, 2019. Citado 3 vezes nas páginas 15, 16 e 17.
- BIRJE, M. N.; BULLA, C. Commercial and open source cloud monitoring tools: A review. *Advances in Decision Sciences, Image Processing, Security and Computer Vision*, Springer, p. 480–490, 2020. Citado na página 15.
- BOSE, S. B.; SUJATHA, S. An efficient qos management and monitoring framework for cloud computing. *Journal of Theoretical and Applied Information Technology*, v. 99, n. 18, p. 4399–4411, 2021. Citado 3 vezes nas páginas 25, 26 e 31.
- BUYYA, R.; RANJAN, R.; CALHEIROS, R. N. Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. In: SPRINGER. *International Conference on Algorithms and Architectures for Parallel Processing*. [S.l.], 2010. p. 13–31. Citado na página 2.

- CARVALHO, J. O. d. Sistema de apoio a decisão para implantar uma aplicação baseada em microsserviços em um ambiente multi-cloud. 2019. Citado 2 vezes nas páginas 7 e 58.
- CARVALHO, J. O. de et al. Evolutionary solutions for resources management in multiple clouds: State-of-the-art and future directions. *Future Generation Computer Systems*, Elsevier, v. 88, p. 284–296, 2018. Citado 6 vezes nas páginas 1, 2, 10, 16, 20 e 21.
- CEDILLO, P. et al. Empirical evaluation of a method for monitoring cloud services based on models at runtime. *IEEE Access*, IEEE, v. 9, p. 55898–55919, 2021. Citado 3 vezes nas páginas 25, 26 e 31.
- CIUFFOLETTI, A. Application level interface for a cloud monitoring service. *Computer Standards & Interfaces*, Elsevier, v. 46, p. 15–22, 2016. Citado na página 12.
- DN.org. *SLA Violations: Investigating Real-World Incidents and How Providers Compensate Clients*. 2025. Acesso em: 10 ago. 2025. Disponível em: <<https://dn.org/sla-violations-investigating-real-world-incidents-and-how-providers-compensate-clients/>>. Citado na página 12.
- FAHAD, A. M.; AHMED, A. A.; KAHAR, M. N. M. The importance of monitoring cloud computing: An intensive review. In: IEEE. *TENCON 2017-2017 IEEE Region 10 Conference*. [S.l.], 2017. p. 2858–2863. Citado 2 vezes nas páginas 2 e 15.
- Flexera. *2025 State of the Cloud Report*. [S.l.], 2025. Acesso em: 22 ago. 2025. Disponível em: <<https://www.flexera.com/blog/finops/cloud-computing-trends-flexera-2025-state-of-the-cloud-report>>. Citado na página 1.
- FOX, A. et al. Above the clouds: A berkeley view of cloud computing. *Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS*, v. 28, n. 13, p. 2009, 2009. Citado na página 1.
- GUO, L. et al. A trust model based on characteristic factors and slas for cloud environments. *IEEE Transactions on Network and Service Management*, IEEE, v. 20, n. 4, p. 4383–4396, 2023. Citado 3 vezes nas páginas 25, 26 e 31.
- HAUSER, C. B.; WESNER, S. Reviewing cloud monitoring: Towards cloud resource profiling. In: IEEE. *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*. [S.l.], 2018. p. 678–685. Citado 2 vezes nas páginas 15 e 18.
- HUSSAIN, W. et al. Formulating and managing viable slas in cloud computing from a small to medium service provider’s viewpoint: A state-of-the-art review. *Information Systems*, Elsevier, v. 71, p. 240–259, 2017. Citado na página 11.
- KESAVULU, M. et al. An overview of user-level usage monitoring in cloud environment. In: *UKAIS*. [S.l.: s.n.], 2018. p. 21. Citado 4 vezes nas páginas 2, 3, 14 e 15.
- KHAN, K. M. et al. Blockchain-enabled real-time sla monitoring for cloud-hosted services. *Cluster Computing*, Springer, v. 25, n. 1, p. 537–559, 2021. Citado 3 vezes nas páginas 25, 26 e 31.
- KHAN, S. U. et al. Adaptive runtime monitoring of service level agreement violations in cloud computing. *Computers, Materials and Continua*, v. 71, n. 3, p. 4199–4220, 2022. Citado 3 vezes nas páginas 25, 26 e 31.

- KIM, I. K. et al. Guaranteeing performance slas of cloud applications under resource storms. *IEEE Transactions on Cloud Computing*, IEEE, v. 10, n. 2, p. 1329–1343, 2022. Citado 2 vezes nas páginas 25 e 31.
- LABIDI, T. et al. Cloud sla modeling and monitoring. In: IEEE. *2017 IEEE International Conference on Services Computing (SCC)*. [S.l.], 2017. p. 338–345. Citado na página 15.
- LEVIN, A. et al. Network monitoring in federated cloud environment. In: IEEE. *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*. [S.l.], 2017. p. 1–6. Citado na página 17.
- LUXNER, T. *Cloud Computing Trends: 2021 State of the Cloud Report*. 2021. Disponível em: <<https://www.flexera.com/blog/cloud/cloud-computing-trends-2021-state-of-the-cloud-report/>>. Citado na página 8.
- MARQUES, G. et al. Proactive resource management for cloud of services environments. *Future Generation Computer Systems*, Elsevier, v. 150, p. 90–102, 2024. Citado 3 vezes nas páginas 25, 26 e 31.
- MEENA, S. *Cloud Monitoring Market, Growth Analysis, Business Statistics, with Forecast to 2028*. 2021. Disponível em: <<https://www.linkedin.com/pulse/cloud-monitoring-market-growth-analysis-business-statistics-meena>>. Citado na página 7.
- MOEYERSONS, J. et al. Towards cloud-based unobtrusive monitoring in remote multi-vendor environments. *Software: Practice and Experience*, Wiley Online Library, v. 52, n. 2, p. 427–442, 2022. Citado 3 vezes nas páginas 25, 26 e 31.
- MURALIDHARAN, C. et al. Three-phase service level agreements and trust management model for monitoring and managing the services by trusted cloud broker. *IET Communications*, John Wiley & Sons, Inc., v. 16, n. 19, p. 2309–2320, 2022. Citado 3 vezes nas páginas 25, 26 e 31.
- NADEEM, S. et al. Runtime management of service level agreements through proactive resource provisioning for a cloud environment. *Electronics*, MDPI, v. 12, n. 2, p. 296, 2023. Citado 3 vezes nas páginas 25, 26 e 31.
- NOOR, A. et al. Cyber-physical application monitoring across multiple clouds. *Computers & Electrical Engineering*, Elsevier, v. 77, p. 314–324, 2019. Citado na página 19.
- NZANZU, V. P. et al. Fedargos-v1: a monitoring architecture for federated cloud computing infrastructures. *IEEE Access*, IEEE, v. 10, p. 133557–133573, 2022. Citado 3 vezes nas páginas 25, 26 e 31.
- PANDEY, A. K.; NARAYAN, D.; SHIVARAJ, K. Sla violation detection and compensation in cloud environment using blockchain. In: IEEE. *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. [S.l.], 2021. p. 1–6. Citado 3 vezes nas páginas 25, 26 e 31.
- PETCU, D. Consuming resources and services from multiple clouds. *Journal of Grid Computing*, Springer, v. 12, n. 2, p. 321–345, 2014. Citado 2 vezes nas páginas 13 e 8.

- POURMAJIDI, W. et al. On challenges of cloud monitoring. *arXiv preprint arXiv:1806.05914*, 2018. Citado 3 vezes nas páginas 12, 15 e 16.
- RAJAKUMAR, P. et al. Client-level monitoring approach to secure data in a cloud environment. In: IEEE. *2023 6th International Conference on Contemporary Computing and Informatics (IC3I)*. [S.l.], 2023. v. 6, p. 1165–1170. Citado 3 vezes nas páginas 25, 26 e 31.
- RODRIGUES, G. D. C. et al. Monitoring of cloud computing environments: concepts, solutions, trends, and future directions. In: *Proceedings of the 31st Annual ACM Symposium on Applied Computing*. [S.l.: s.n.], 2016. p. 378–383. Citado 3 vezes nas páginas 2, 14 e 15.
- SAIFENG, Z. Aqinm: an adaptive qos management framework based on intelligent negotiation and monitoring in cloud. *International Journal of Information Technology and Management*, Inderscience Publishers (IEL), v. 23, n. 1, p. 33–47, 2024. Citado 3 vezes nas páginas 25, 26 e 31.
- SHENDE, M. S.; CHANDAK, M. B. Cost optimization of cloud services through automated analytics and resource allocation. *Grenze International Journal of Engineering & Technology (GIJET)*, v. 10, 2024. Citado 3 vezes nas páginas 25, 26 e 31.
- SIM, K. M. Agent-based approaches for intelligent intercloud resource allocation. *IEEE Transactions on Cloud Computing*, IEEE, v. 7, n. 2, p. 442–455, 2016. Citado na página 16.
- SYED, H. J. et al. Cloud monitoring: A review, taxonomy, and open research issues. *Journal of Network and Computer Applications*, Elsevier, v. 98, p. 11–26, 2017. Citado 6 vezes nas páginas 2, 14, 15, 16, 17 e 19.
- TAHERIZADEH, S. et al. Monitoring self-adaptive applications within edge computing frameworks: A state-of-the-art review. *Journal of Systems and Software*, Elsevier, v. 136, p. 19–38, 2018. Citado 2 vezes nas páginas 14 e 15.
- TAN, W. et al. A novel service level agreement model using blockchain and smart contract for cloud manufacturing in industry 4.0. *Enterprise Information Systems*, Taylor & Francis, v. 16, n. 12, p. 1939426, 2022. Citado 3 vezes nas páginas 25, 26 e 31.
- TRIHINAS, D.; PALLIS, G.; DIKAIAKOS, M. D. Monitoring elastically adaptive multi-cloud services. *IEEE Transactions on Cloud Computing*, IEEE, v. 6, n. 3, p. 800–814, 2015. Citado na página 19.
- TURK, A. et al. Seeing into a public cloud: Monitoring the massachusetts open cloud. In: *{USENIX} Workshop on Cool Topics on Sustainable Data Centers (CoolDC 16)*. [S.l.: s.n.], 2016. Citado na página 2.
- VAQUERO, L. M. et al. *A break in the clouds: towards a cloud definition*. [S.l.]: ACM New York, NY, USA, 2008. Citado na página 1.
- VIEGAS, E. et al. Enhancing service maintainability by monitoring and auditing sla in cloud computing. *Cluster Computing*, Springer, v. 24, p. 1659–1674, 2021. Citado 3 vezes nas páginas 25, 26 e 31.

- WU, Y. et al. A distributed monitoring architecture for jointcloud computing. *Future Generation Computer Systems*, Elsevier, v. 168, p. 107773, 2025. Citado 3 vezes nas páginas 25, 26 e 31.
- ZEDAN, M.; ATTIYA, G.; EL-FISHAWY, N. Load balancing based active monitoring load balancer in cloud computing. In: IEEE. *2021 International Conference on Electronic Engineering (ICEEM)*. [S.l.], 2021. p. 1–6. Citado 3 vezes nas páginas 25, 26 e 31.
- ZHAN, D. et al. Container introspection: using external management containers to monitor containers in cloud computing. *Computers, Materials & Continua*, Tech Science Press, v. 69, n. 3, p. 3783–3794, 2021. Citado 3 vezes nas páginas 25, 26 e 31.