

Métricas Aplicadas a Microsserviços

Flávia C. dos Santos¹, Co-Orientador: Michel S. Soares², Orientador: Fabio G. Rocha³

¹Programa de Pós-Graduação em Ciência da Computação - PROCC/UFS
São Cristovão – SE – Brasil

flavicarolines@gmail.com, michel@dcomp.ufs.br, gomesrocha@gmail.com

Nível do Curso: Mestrado

Início: agosto/2022 Fim planejado: agosto/2024 Data da aprovação: 08/jul/2023

Abstract. Introduction: *Microservices provide independent services, reducing development and maintenance time. Objective:* *To collect metrics from academia and industry, identify management needs, and associate these metrics with the ISO/IEC 25010 standard. In addition, create dashboards of graphic elements based on metrics aimed at academia and industry that use microservices. Methodology:* *Design Science Research is adopted to address unsolved problems related to microservices. Conclusion:* *This research will contribute to understanding the needs of academia and industry, comparing the two areas and building metrics that serve as a guideline for both.*

Vídeo Youtube: <https://www.youtube.com/watch?v=fVnYgSGgqQw>

Resumo. Introdução: *Os microsserviços provêm serviços independentes, redução de tempo de desenvolvimento e manutenção. Objetivo:* *Coletar métricas da academia e da indústria, identificando necessidades de gerenciamento, associando essas medidas à norma ISO/IEC 25010. Além disso, construir painéis de elementos gráficos baseado nessas métricas direcionadas à academia e indústria que utilizam microsserviços. Metodologia:* *Foi adotado o Design Science Research para abordar problemas não resolvidos relacionados a microsserviços. Conclusão:* *Esta pesquisa contribuirá para compreensão das necessidades da academia e da indústria, comparando as duas áreas e construindo métricas que sirvam como direcionamento para ambos por meio da construção de um catálogo e de um dashboard direcionado a empresas que utilizam microsserviços em seu desenvolvimento.*

1. Contexto

No desenvolvimento de software a arquitetura monolítica vem sendo adotada há décadas, sendo considerada até hoje como um modelo adequado para desenvolvimento de softwares considerados pequenos ou pouco complexos [Kalske et al. 2017]. Porém, a partir dos anos 2000, com o surgimento do manifesto ágil [Beck et al. 2001], a Engenharia de Software precisou adaptar-se ao ritmo acelerado de demandas e uso de métodos ágeis. Posteriormente, na década de 2010, cresceu a demanda por sistemas distribuídos com conexão de usuários em diferentes partes do mundo, requerendo a capacidade de evoluir ao longo do tempo [Boehm 2006]. Assim, Kalske et. al, [Kalske et al. 2017] apontam que tais demandas deram origem ao que hoje temos como microsserviços. Inicialmente, surgiu a Arquitetura Orientada a Serviços (SOA) [Boehm 2006], sendo a primeira geração de uma arquitetura que tem os serviços como base [Dragoni et al. 2017]. A evolução

do SOA com base nas demandas de agilidade e descentralização, fez surgir a segunda geração, denominada de microsserviços, na qual há melhorias relacionadas ao desempenho, facilidade de implementação e remoção de níveis desnecessários de complexidade para focar na programação. Assim, o microsserviço difere do monolítico, focando em módulos menores, sustentáveis e independentes. Porém, assim como qualquer outro modelo, são necessárias métricas que promovam sua qualidade, bom funcionamento, eficiência e eficácia.

Nesse sentido, esta pesquisa é motivada pela adoção de microsserviços a partir da década de 2010 por empresas como *SoundCloud*, *NetFlix*, *AWS (Amazon Web Services)*, *Uber* entre outras que migraram para essa arquitetura [Balalaie et al. 2015]. Entre os fatores que levaram a essa migração estão a redução de investimento de infraestrutura, capacidade de escalabilidade horizontal e vertical, manutenibilidade, reusabilidade e adaptabilidade [Rocha et al. 2023]. Apesar dos benefícios, surgem pontos de complexidade que são introduzidos nos sistemas e que demandam empenho para serem solucionadas [Balalaie et al. 2015], além de exigir grandes esforços por parte da equipe, impactando no produto final [Bu 2011]. Assim, para uma análise mais assertiva, este trabalho tem como objetivo coletar métricas da indústria e da academia, e por meio desta pesquisa construir um catálogo de métricas, associando essas medidas a norma ISO/IEC 25010. Também será desenvolvido um *dashboard* com o objetivo de auxiliar gestores e desenvolvedores a visualizar seus indicadores.

2. Fundamentação Teórica

2.1. Monolítico

A arquitetura monolítica adota um estilo em que o software é implementado como único e indivisível componente [Kalske et al. 2017]. No entanto, a arquitetura monolítica também traz consigo desafios significativos, sendo que o software é, normalmente, executado em um único processo, gerando problemas de desempenho à medida que cresce em tamanho e complexidade [Rocha et al. 2023]. As atualizações e implantações também podem ser mais arriscadas, pois uma alteração em uma parte do software pode afetar toda a estrutura [Kalske et al. 2017]. Os pontos levantados como problemas da arquitetura monolítica geraram pesquisa na década de 1980, as quais visaram reduzir o impacto relacionado a reaproveitamento e a dificuldade de atualização de sistemas. Assim, surgiu o Knowledge-Based Software Assistant (KBSA) [Boehm 2006]. O KBSA trazia consigo a integração de ambientes que permitia maiores níveis de reaproveitamento, sendo estas algumas das características que deram base ao que hoje denominado microsserviço [Boehm 2006].

2.2. Arquitetura Orientada a Serviços - SOA

A SOA é uma arquitetura que adota um modelo de comunicação por meio de módulos distribuídos e independentes [Toueir et al. 2011], que se comunicam por meio de mensagens [Munialo et al. 2020]. É nesse modelo onde estão as regras de negócio e validações [Hojaji and Shirazi 2010]. Essa arquitetura possui apenas uma base de dados para todos os serviços [Sheikh and Ambhaikar 2021], o que pode impactar negativamente no desempenho do produto [Lee 2010], dando origem posteriormente aos microsserviços [Rocha et al. 2023].

2.3. Microsserviços

A Arquitetura de microsserviços surgiu a partir da SOA em comunidades de desenvolvimento ágil e recentemente vem recebendo atenção da indústria e da academia [Li et al. 2022]. Os microsserviços possuem um estilo de design de software que organiza os serviços em módulos funcionais, autônomos, com base de dados diferentes [Chiba et al. 2019]. Por esse motivo oferece manutenção ágil e, em uma análise mais profunda, redução de tempo de desenvolvimento [dos Santos Moura et al. 2022], desenvolvedores e outros custos financeiros [Li et al. 2022]. No entanto, a sua qualidade depende de características como coesão, escalabilidade, modularidade, tempo de resposta que é um atributo essencial das métricas de desempenho [Dong et al. 2020]. Dessa forma, é possível proporcionar evolução a partir destas métricas que estão nos requisitos [Pulparambil et al. 2018] de qualidade da norma ISO/IEC 25010.

2.4. ISO/IEC 25010

Em 2011 surgiu a ISO/IEC 25010, que tem como características: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade, portabilidade, segurança e compatibilidade. Todas estas podem ser medidas, proporcionando clareza no acompanhamento da qualidade do software [Nik Daud and Wan Kadir 2015]. Tais características podem estar ligadas a uma ou mais métricas para avaliar a qualidade de um sistema e produtividade de um time.

2.5. Métricas

Métricas de software fornecem métodos quantitativos para pontuar a qualidade de um software [Asik and Selcuk 2017], permitindo monitorar seu desempenho, gerenciamento de recursos [Nik Daud and Wan Kadir 2015] e, conseqüentemente, resultar em melhor desempenho organizacional [Asik and Selcuk 2017], além de permitir a identificação de deficiências quando monitoradas adequadamente [Vale et al. 2022]. Software de baixa qualidade pode impactar negativamente os usuários que o utilizam [Nik Daud and Wan Kadir 2015] e, para empresas, resultar em danos significativos impactando sua reputação [Lee 2010]. Portanto, analisar as práticas do setor permite prevenir perdas significativas e gerar motivação para os desenvolvedores [Asik and Selcuk 2017], resultando em processos bem-sucedidos. No entanto, é necessário empregar diversas métricas para que se possa rastrear diferentes ângulos possíveis das anomalias dos serviços [Vale et al. 2022].

3. Trabalhos Relacionados

Li et al. [Li et al. 2022] conduziram experimentos buscando analisar métricas de desempenho, coesão, acoplamento e redundância de código. Foi possível re-arquitetar as aplicações, examinando por meio de grafos. Como resultado foi possível extrair de sistemas monolíticos, entidades, relacionamentos e identificar candidatos a se tornarem microsserviços.

Hassan et al. [Hassan et al. 2022] buscou, em seus experimentos, tomar decisão sobre a granularidade adequada com base na necessidade de escalabilidade e realizando comparações para confirmar que a decisão melhorou a escalabilidade do serviço. Ainda, o experimento permitiu identificar formas de definir um escopo iterativo para arquitetura

de microsserviços. Como resultado, os autores obtiveram um catálogo de métricas para microsserviços com foco em escalabilidade.

O *Scrum Watch*, segundo Florencia et. al. [Vega et al. 2022], teve como objetivo resolver problemas de gestão, lidando com grande volume de dados para melhores tomadas de decisões em equipes que utilizam metodologias ágeis. Os autores visaram permitir que gerentes de projetos visualizassem o desempenho da equipe, o desempenho da sprint e também a qualidade do código-fonte. Como resultado, obtiveram um *dashboard*, por meio do processo de ETL, e as métricas foram definidas com base em uma pesquisa com 10 gerentes de projetos.

Como diferencial, este trabalho possui a construção de um catálogo de medidas, focado em necessidades de empresas e também da academia, diferente da pesquisa de [Hassan et al. 2022] focado em escalabilidade. Para a construção do *dashboard*, Este trabalho dá continuidade ao [Vega et al. 2022], focando sua análise nos microsserviços.

4. Metodologia

Será adotado o *Design Science Research*, pois permite unir pesquisa e inovação no campo de sistemas de informação [Hevner et al. 2004, Horita E.A. et al. 2020]. Isso tornará possível consolidar o conhecimento das pesquisas realizadas sobre métricas aplicadas a microsserviços, bem como desenvolver artefatos [Horita E.A. et al. 2020] que permitam melhorar a avaliação da qualidade de softwares existentes. Artefatos são elementos empregados para descrever o conhecimento obtido em pesquisas e, assim, apoiar o desenvolvimento de novos produtos organizacionais e/ou computacionais [Hevner et al. 2004].

Para desenvolver a pesquisa, foi realizada uma revisão intencional e uma revisão sistemática com análise bibliométrica. Com bases nos resultados da revisão, está sendo realizada uma pesquisa de opinião a partir da análise do ambiente de aplicação em empresas que utilizam microsserviços. Posteriormente, com base nos resultados que serão coletados, será construído um catálogo de métricas para auxiliar gestores e equipes na gerência e acompanhamento da qualidade dos microsserviços. Com base no catálogo, será desenvolvido o artefato de *dashboard*, ou seja, a solução, que será um produto com elementos gráficos e relatórios de métricas. Por fim, será realizado um estudo de caso, onde o produto será testado e validado na indústria, inicialmente nas empresas entrevistadas. A Figura 1 apresenta de forma resumida a metodologia adotada.

5. Contribuições esperadas

Será finalizada a pesquisa de opinião que foi identificada na Figura 1 como último passo da base de conhecimento. Esse passo permitirá comparar o estado da prática das métricas de microsserviços com a pesquisa realizada até o momento. Com base nos resultados das revisões e da pesquisa de opinião, como contribuição científica esperada, será construído um catálogo de métricas considerando as características da ISO/IEC 25010. Posteriormente, como contribuição tecnológica, planeja-se construir um *dashboard* com as métricas identificadas nesta pesquisa, visando auxiliar empresas a acompanhar a qualidade dos microsserviços. Tal artefato tem como objetivo apoiar a tomada de decisão. Tanto o catálogo quanto o *dashboard* serão avaliados por meio de um estudo de caso junto à indústria, em uma empresa parceira que atua na área de segurança.



Figura 1. Ciclos do *Design Science Research*

Referências

- Asik, T. and Selcuk, Y. (2017). Policy Enforcement Upon Software Based on Microservice Architecture. pages 283–287. Institute of Electrical and Electronics Engineers Inc.
- Balalaie, A., Heydarnoori, A., and Jamshidi, P. (2015). Migrating to Cloud-Native Architectures using Microservices: An Experience Report. In *European Conference on Service-Oriented and Cloud Computing*, pages 201–215. Springer.
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D. (2001). Manifesto for agile software development.
- Boehm, B. (2006). A View of 20th and 21st Century Software Engineering. In *Proceedings of the 28th International Conference on Software Engineering*, pages 12–29.
- Bu, H. (2011). Metrics for Service Granularity in Service Oriented Architecture. In *Proceedings of 2011 International Conference on Computer Science and Network Technology*, volume 1, pages 491–494. IEEE.
- Chiba, T., Nakazawa, R., Horii, H., Suneja, S., and Seelam, S. (2019). Confadvisor: A performance-centric configuration tuning framework for containers on kubernetes. In *2019 IEEE International Conference on Cloud Engineering (IC2E)*, pages 168–178.
- Dong, B., Ying, S., Li, L., Luo, H., and Yang, Z. (2020). Impact analysis about response time considering deployment change of saas software. *International Journal of Software Engineering and Knowledge Engineering*, 30(07):977–1004.
- dos Santos Moura, A., de Oliveira Lima, M. A., Rocha, F. G., and Soares, M. S. (2022). Recommendation of microservices patterns through automatic information retrieval using problems specified in natural language. In *Computational Science and Its Applications – ICCSA 2022 Workshops: Malaga, Spain, July 4–7, 2022, Proceedings, Part V*, page 489–501, Berlin, Heidelberg. Springer-Verlag.
- Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustafin, R., and Safina, L. (2017). Microservices: Yesterday, Today, and Tomorrow. *Present and Ulterior Software Engineering*, pages 195–216.

- Hassan, S., Bahsoon, R., and Buyya, R. (2022). Systematic Scalability Analysis for Microservices Granularity Adaptation Design Decisions. *Software - Practice and Experience*, 52.
- Hevner, A., R, A., March, S., T, S., Park, Park, J., Ram, and Sudha (2004). Design science in information systems research. *Management Information Systems Quarterly*, 28:75–.
- Hojaji, F. and Shirazi, M. (2010). Developing a More Comprehensive and Expressive SOA Governance Framework. volume 1, Chengdu.
- Horita E.A., F., Rocha, F. G., Souza, L. S., and Gonzales, G. R. (2020). Design science in digital innovation: A literature review. SBSI'20, New York, NY, USA. Association for Computing Machinery.
- Kalske, M., Mäkitalo, N., and Mikkonen, T. (2017). Challenges when Moving from Monolith to Microservice Architecture. In *International Conference on Web Engineering*, pages 32–47. Springer.
- Lee, Y. (2010). Qos metrics for service level measurement for soa environment. In *2010 6th International Conference on Advanced Information Management and Service (IMS)*, pages 509–514.
- Li, Z., Shang, C., Wu, J., and Li, Y. (2022). Microservice Extraction based on Knowledge Graph from Monolithic Applications. *Information and Software Technology*, 150.
- Munialo, S., Muketha, G., and Omieno, K. (2020). An Effort Estimation Method for Service-Oriented Architecture. *Journal of Engineering Science and Technology Review*, 13.
- Nik Daud, N. and Wan Kadir, W. (2015). Review on structural properties metrics in SOA design. *Jurnal Teknologi*, 77.
- Pulparambil, S., Baghdadi, Y., Al-Hamdani, A., and Al-Badawi, M. (2018). Service Design Metrics to Predict IT-Based Drivers of Service Oriented Architecture Adoption. *2018 9th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2018*.
- Rocha, F., Soares, M., and Rodriguez, G. (2023). Patterns in microservices-based development: A grey literature review. In *Anais do XXVI Congresso Ibero-Americano em Engenharia de Software*, pages 61–76, Porto Alegre, RS, Brasil. SBC.
- Sheikh, A. and Ambhaikar, A. (2021). Quality of services parameters for architectural patterns of IoT. *Journal of Information Technology Management*, 13.
- Toueir, A., Broisin, J., and Sibilla, M. (2011). Toward Configurable Performance Monitoring: Introduction to Mathematical Support for Metric Representation and Instrumentation of the CIM Metric Model. *5th Int. DMTF Academic Alliance Workshop on Systems and Virtualization Management: Standards and the Cloud*.
- Vale, G., Correia, F., Guerra, E., Rosa, T., Fritsch, J., and Bogner, J. (2022). Summary of the artifact accompanying the article : “designing microservice systems using patterns: An empirical study on quality trade-offs”. pages 57–57.
- Vega, F., Rodríguez, G., Rocha, F., and dos Santos, R. P. (2022). Scrum watch: a tool for monitoring the performance of scrum-based work teams. *Journal of Universal Computer Science*, 28(1):98.