

Modelo para Caracterização e Evolução de Sistemas com Arquitetura Baseada em Serviços

Thatiane de Oliveira Rosa^{1,2} (doutoranda), Alfredo Goldman¹ (orientador),
Eduardo Martins Guerra³ (co-orientador)

¹ Doutorado Acadêmico em Ciência da Computação
Programa de Pós-Graduação em Ciência da Computação
Departamento de Ciência da Computação – Instituto de Matemática e Estatística –
Universidade de São Paulo (IME/USP)
Rua do Matão, 1010 – CEP 05508-090 – São Paulo – SP – Brasil
Ingresso: 07/2017 – Qualificação: 01/2020 – Previsão Defesa: 08/2021

²Instituto Federal de Educação, Ciência e Tecnologia do Tocantins (IFTO)
Distrito Agroindustrial, BR153, KM480 – 77600-000 - Paraíso do Tocantins - TO – Brasil

³Faculdade de Ciência da Computação – Universidade Livre de Bozen-Bolzano (UNBZ)
Dominikanerplatz 3 - piazza Domenicani, 3, 39100 – Bozen-Bolzano – Itália

{thatiane,gold}@ime.usp.br, guerraem@gmail.com

Resumo. *Construir uma boa arquitetura é fundamental para obter sucesso no processo de desenvolvimento de software. Porém, essa não é uma atividade trivial, pois envolve uma série de decisões de domínio do negócio, técnicas e de estrutura. Assim, com a finalidade de apoiar a construção de uma arquitetura mais adequada a um determinado contexto, constantemente surgem novas abordagens como microsserviços e monolitos modulares, que propõem decompor um software complexo em pequenas partes fracamente acopladas. Entretanto, ao analisar o estado da arte e da prática, percebe-se que existe um limite nebuloso na classificação dessas diferentes abordagens, especialmente as baseadas em serviços. Além de existir pouco suporte para caracterizar e direcionar arquiteturas baseadas em serviços. Diante disso, com base em pesquisas bibliográficas e estudos de caso, pretende-se desenvolver um modelo para caracterização e evolução da arquitetura de sistemas baseados em serviços, adotando diretrizes de microsserviços. Esse modelo apresenta dimensões que medem características estruturais da arquitetura relacionadas ao tamanho de módulos, compartilhamento de bases de dados e acoplamento entre serviços. Como contribuição final, esse modelo facilitará o mapeamento e medição de diferentes impactos gerados na arquitetura de software, a partir de incrementos e refatorações realizados. Assim como dará suporte a decisões arquiteturais que considerem diferentes atributos de qualidade para alcançar o equilíbrio entre independência e colaboração de serviços para um determinado sistema.*

Palavras-chave: *Arquitetura de software, Sistemas baseados em serviços, Microsserviços, Modelo para caracterização e evolução.*

Eventos CBSOFT: *SBES e SBCARS*

1. Introdução

O estilo arquitetural de microsserviços tem o objetivo de viabilizar o desenvolvimento de sistemas complexos por meio de um conjunto de serviços pequenos, onde cada microsserviço é responsável por executar seus próprios processos e comunicar-se com os demais por meio de mecanismos leves [1, 2]. Estudos indicam que, desde 2014, o interesse e a adoção desse estilo arquitetural tem aumentado tanto na indústria quanto na academia [3, 4, 5, 6]. Para iniciar o desenvolvimento de sistemas baseados em microsserviços, Fowler [7] e Newman [8] recomendam começar implementando uma solução monolítica e em seguida evoluir para microsserviços, pois assim é possível compreender melhor a complexidade do sistema e identificar seus componentes e fronteiras.

Assim como microsserviços, existem outros estilos arquiteturais, tais SOA (*Service-Oriented Architecture* – Arquitetura Orientada a Serviços) [9] e Sistemas Autocontidos (*Self-Contained Systems*) [10], que seguem uma abordagem modular e têm como princípio a decomposição de sistemas complexos em serviços fracamente acoplados e que se comunicam por mensagens. No contexto deste trabalho, a arquitetura de sistemas que segue esse princípio, será denominada como “baseada em serviços”.

Arquitetura baseada em serviços surgiu com o objetivo de superar desafios tais como forte dependência entre elementos, baixa produtividade e dificuldade para escalar, que são comuns ao desenvolver, implantar e manter sistemas monolíticos [11]. Ao adotar a abordagem baseada em serviços, algumas das vantagens obtidas são maior agilidade de desenvolvimento, heterogeneidade tecnológica, escalabilidade, resiliência e reutilização [12, 8, 10, 13, 14].

Apesar dos benefícios citados, desenvolver sistemas baseados em serviços não é trivial. Estudos indicam que alguns dos desafios enfrentados estão relacionados à complexidade do processo de desenvolvimento, dificuldade em definir o tamanho e nível de acoplamento dos serviços e manter a consistência dos dados [12, 11, 10, 5]. Além disso, desenvolver uma arquitetura baseada em serviços é caro e exige uma infraestrutura robusta e automatizada [8, 4].

Diante dos desafios apresentados e ao considerar relatos de experiência de empresas, como Uber [15] e Segment [16], que adotaram uma abordagem baseada em serviços, é possível perceber que não existe um consenso sobre qual estrutura arquitetural é a mais adequada. Tal cenário é um indício de que existe pouco suporte para caracterizar, avaliar e direcionar arquiteturas baseadas em serviços. Além disso, ao comparar diferentes abordagens, existe um limite nebuloso na classificação desse tipo de arquitetura [8, 17].

Dessa forma, a questão que norteia esta pesquisa de doutorado é: “*Como caracterizar a arquitetura de sistemas baseados em serviços para guiar a tomada de decisões arquiteturais?*”. Logo, alinhado à questão de pesquisa, o objetivo geral deste trabalho consiste em: “*Desenvolver um modelo para caracterização e evolução da arquitetura de sistemas baseados em serviços, adotando diretrizes de microsserviços*”.

2. Fundamentação Teórica

Uma arquitetura de *software* quando bem definida auxilia no gerenciamento da complexidade, evita problemas durante o processo de desenvolvimento e facilita a compreensão, reutilização, desenvolvimento, análises, evolução e manutenção do *software* [18, 19].

Uma boa arquitetura também pode ajudar um sistema a atender atributos de qualidade como desempenho, confiabilidade, portabilidade, escalabilidade e interoperabilidade. Por outro lado, uma arquitetura ruim pode ser catastrófica para o projeto, gerando um *software* complexo, difícil de manter e evoluir [20, 19].

Com o propósito de auxiliar na definição de uma arquitetura adequada aos objetivos do negócio, às características e ao comportamento esperado do *software*, são adotados estilos e padrões arquiteturais [11]. Ao considerar o contexto deste trabalho, os estilos arquiteturais mais relevantes são:

- Monolítico: estilo arquitetural mais comum, em que as aplicações possuem um único componente executável ou implantável [14];
- Monolítico Modular: abordagem semelhante à monolítica, em que os limites entre diferentes domínios de um sistema são rigorosamente determinados [21];
- Arquitetura Orientada a Serviços: abordagem cliente/servidor em que uma aplicação é composta por usuários (clientes) e provedores (servidores) de serviços de *software*, que enfatiza o acoplamento flexível entre os componentes e o uso de interfaces independentes [9];
- Sistemas Autocontidos: abordagem que visa separar as funcionalidades (casos de uso) de um sistema complexo em vários sistemas autônomos (que possuem interface com usuário, lógica de negócios e banco de dados próprios), que comunicam/colaboram entre si por meio de mecanismos leves, majoritariamente de forma assíncrona [10];
- Microsserviços: abordagem que propõe o desenvolvimento de sistemas complexos a partir de um conjunto de serviços “pequenos, autônomos e que trabalham em conjunto” [8]. Diferente dos sistemas autocontidos, geralmente, os microsserviços são focados na lógica de negócios [22].

Durante este estudo são adotadas diretrizes do estilo arquitetural de microsserviços relacionadas aos atributos estruturais de tamanho e acoplamento. Tais atributos estruturais foram escolhidos pois “baixo acoplamento” e “alta coesão” são características fundamentais para obter sistemas de *software* de qualidade [23, 24, 25] e por estarem atrelados a alguns dos principais desafios de desenvolvimento de sistemas baseados em microsserviços [5].

3. Metodologia

Para atingir o objetivo proposto, a metodologia de pesquisa está sendo executada conforme ilustrado na Figura 1. Do ponto de vista dos objetivos, tal pesquisa pode ser classificada como exploratória e descritiva. Já do ponto de vista dos procedimentos, esta pesquisa pode ser classificada como bibliográfica, documental e estudo de caso.

Em um primeiro momento, foi realizada uma revisão bibliográfica *ad-hoc* onde foram consultados tanto estudos científicos primários e secundários, quanto relatos de experiência industrial. Essa revisão bibliográfica teve o objetivo de mapear conceitos-chave e compreender a importância da arquitetura de *software*. Buscou-se também entender o funcionamento e identificar pontos positivos e negativos de diferentes estilos arquiteturais, sobretudo baseados em serviços. Além disso, essa etapa teve o objetivo de compreender diretrizes de microsserviços relevantes. Logo, foi possível identificar atributos estruturais interessantes e assim refinar o problema de pesquisa. A partir disso, foi elaborada a primeira versão do modelo proposto.

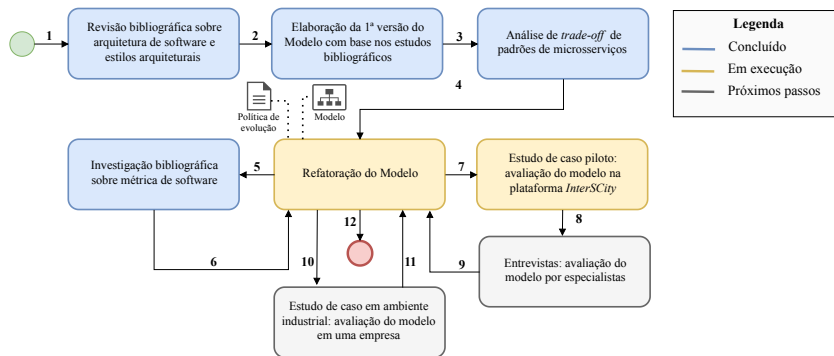


Figura 1. Etapas da metodologia.

No passo seguinte, foi realizada uma análise sistemática para identificar a influência exercida por padrões de microsserviços nos atributos estruturais relacionados a tamanho, compartilhamento de base de dados e nível de acoplamento. A partir dessa análise, foi possível mapear perdas e ganhos da adoção dos padrões analisados, assim como refinar ainda mais o problema de pesquisa e identificar pontos que poderiam ser melhorados na primeira versão do modelo proposto.

Em seguida, foi realizada uma nova pesquisa bibliográfica *ad-hoc*, consultando estudos científicos primários e secundários. Tal etapa teve como objetivo identificar métricas relevantes para o contexto desta pesquisa e relacionadas aos atributos estruturais de tamanho e acoplamento. Como resultado, obteve-se uma lista de métricas candidatas viáveis (Seção 4). Com base nesses resultados, o modelo proposto está sendo refinado e uma segunda versão está sendo desenvolvida, a qual é explorada na Seção 4.

Com o objetivo de avaliar o modelo proposto, será realizada uma pesquisa qualitativa, dividida em três etapas: estudo de caso piloto, entrevistas com especialistas e estudo de caso em ambiente industrial.

- **Estudo de caso piloto:** A segunda versão do modelo está sendo utilizada para caracterizar a arquitetura da plataforma *InterSCity* [26] para cidades inteligentes. Durante esta etapa, pretende-se identificar ajustes necessários ao modelo, assim como criar uma versão inicial de uma política de evolução, indicando incrementos e refatorações que devem ser implementados na plataforma. A conclusão desta etapa está prevista para setembro de 2020;
- **Entrevistas com especialistas:** o modelo desenvolvido será apresentado a especialistas em arquitetura de *software*, que atuem na indústria e/ou na academia. O objetivo é que esses profissionais avaliem o modelo e sugiram melhorias. Ao final desta etapa, com base no *feedback* recebido, pretende-se obter uma versão otimizada do modelo. A execução desta etapa está prevista para iniciar em outubro de 2020 e finalizar em janeiro de 2021;
- **Estudo de caso em ambiente industrial:** pretende-se realizar um estudo de caso em uma empresa que atenda a critérios predefinidos. Após escolher a empresa e o projeto, será delimitado o contexto a ser analisado durante o estudo. Esta etapa tem a finalidade de avaliar o modelo desenvolvido em um ambiente real. Além disso, espera-se entender os impactos gerados a partir de certas evoluções da arquitetura e, conseqüentemente, melhorar a política de evolução. A previsão de execução desta etapa é de janeiro até agosto de 2021.

4. Resultados Preliminares

Conforme descrito na Seção 3, uma das etapas da metodologia que já foi concluída é a pesquisa bibliográfica sobre arquitetura de *software* e estilos arquiteturais. Tal pesquisa, embasou o desenvolvimento da primeira versão do modelo proposto nesta pesquisa, que foi chamado de CEMMBA (*Characterization and Evolution Model of Microservice-Based Architectures*). Tal resultado foi publicado na trilha de Ideias Novas e Emergentes do ICOSA-2020 (*International Conference on Software Architecture*) [27].

Outro resultado preliminar deste trabalho, consiste na análise de perdas e ganhos da adoção de padrões de microsserviços. Os resultados obtidos durante essa etapa foram organizados em formato de artigo, intitulado “*A Method for Architectural Trade-off Analysis Based on Patterns: Evaluating Microservices Structural Attributes*”, que foi aceito para publicação nos Anais da EuroPLoP 2020 (*European Conference on Pattern Languages of Programs*). O objetivo geral desse artigo foi apresentar um método sistemático para identificar padrões arquiteturais que influenciam um determinado conjunto de atributos de qualidade. Para demonstrar o passo a passo de execução desse método, analisou-se a influência de padrões de microsserviços sobre os seguintes atributos estruturais: tamanho dos módulos/serviços, compartilhamento de bases de dados e acoplamento entre serviços. A partir dessa análise, foi possível identificar a influência exercida por determinados padrões de microsserviços sobre os atributos estruturais estudados.

Neste momento está sendo desenvolvida a segunda versão do modelo proposto. Essa versão é composta por quatro dimensões denominadas: tamanho dos módulos¹, compartilhamento de bases de dados entre módulos, acoplamento síncrono entre serviços² e acoplamento assíncrono entre serviços.

- **Tamanho dos módulos:** esta dimensão caracteriza a composição dos módulos, considerando serviços e operações. O objetivo é identificar quais critérios devem ser adotados para limitar o escopo dos serviços e para agrupar serviços em módulos. As métricas adotadas nesta dimensão são: número de serviços por módulo e número de operações por serviço [30];
- **Compartilhamento de bases de dados entre módulos:** esta dimensão caracteriza a distribuição de base de dados entre os módulos, apontando se uma base de dados é utilizada por um ou mais módulos. O objetivo é auxiliar na escolha da estratégia mais adequada para compartilhamento das bases de dados do sistema. As métricas adotadas nesta dimensão são: número de bases de dados por módulo e número de módulos que compartilham uma mesma base de dados;
- **Acoplamento síncrono entre os serviços:** esta dimensão caracteriza o grau de dependência síncrona entre os serviços do sistema. Em uma interação síncrona, um serviço x faz uma solicitação a um serviço y e fica bloqueado, aguardando uma resposta em tempo hábil [14]. Logo, a finalidade dessa dimensão é guiar a construção do esquema de comunicação síncrona entre os serviços de um sistema. As métricas adotadas nesta dimensão são a AIS (*Absolute Importance of the Service*), que indica o número de serviços clientes que invocam as operações de um determinado serviço e a ADS (*Absolute Dependence of the Service*), que indica o número de serviços que um dado serviço é dependente [31];

¹Módulo: conjunto coeso de serviços que são implantados juntos [28]

²Serviço: conjunto de funcionalidades que representam tarefas do negócio [29]

- **Acoplamento assíncrono entre os serviços:** esta dimensão caracteriza o grau de dependência assíncrona entre os serviços. Diferente da síncrona, na interação assíncrona o serviço x que enviou uma solicitação a um serviço y , não fica bloqueado aguardando a resposta, que pode demorar ou não chegar [14]. Assim, a finalidade desta dimensão é guiar a construção do esquema de comunicação assíncrona entre os serviços de um sistema. As métricas adotadas nesta dimensão também são a AIS, para coletar o número de diferentes tipos de mensagens publicadas por um determinado serviço e ADS para coletar o número de diferentes tipos de mensagens consumidas por um determinado serviço.

Essa segunda versão do modelo está sendo avaliada por meio de um estudo de caso piloto na plataforma *InterSCity*. Neste momento, a documentação da plataforma está sendo analisada³ e dados relevantes sobre cada dimensão estão sendo coletados. O próximo passo é realizar entrevistas semiestruturadas com os responsáveis para compreender melhor e consolidar os dados coletados. A partir da análise de perdas e ganhos da adoção de padrões de microsserviços e das entrevistas e caracterização da arquitetura da plataforma, pretende-se iniciar o desenvolvimento da política de incrementos e refatorações, considerando os pontos críticos mapeados durante esse estudo de caso.

5. Contribuições Esperadas

Com o desenvolvimento deste trabalho, as principais contribuições científicas e técnicas esperadas são:

- **[Científica] Método sistemático para análise de perdas e ganhos da adoção de padrões arquiteturais:** considera-se que o método desenvolvido apresenta um passo a passo que guia os arquitetos de *software* na identificação de padrões arquiteturais que melhor atendem às necessidades de um determinado projeto;
- **[Técnica] Análise de perdas e ganhos da adoção de padrões de microsserviços que influenciam os atributos estruturais de tamanho dos módulos/serviços, compartilhamento de bases de dados e acoplamento entre os serviços:** Acredita-se que os resultados alcançados auxiliam a tomada de decisão no contexto de sistemas baseados em serviços. Isso porque, o estudo realizado apresenta um conjunto de evidências que demonstram o impacto de determinados padrões de microsserviços sobre os atributos estruturais investigados;
- **[Científica] Modelo para caracterização e evolução da arquitetura de sistemas baseados em serviços:** supõem-se que esse modelo contribua para que os arquitetos de *software* entendam melhor as características estruturais de um determinado sistema e, conseqüentemente, identifiquem mais facilmente problemas a serem resolvidos. Ou seja, espera-se que o modelo proposto seja uma ferramenta para identificar aspectos estruturais que devem ser melhorados ou mantidos;
- **[Técnica] Política de evolução da arquitetura de sistemas baseados em serviços:** espera-se desenvolver um catálogo de evoluções, que combinado com o modelo de caracterização, sirva de guia para ajustar a arquitetura de acordo com a abordagem desejada pela organização. Logo, a proposta é que a partir da política de evolução o arquiteto possa direcionar melhor as decisões em cada uma das dimensões do modelo.

³ [26] e <https://gitlab.com/intercity/intercity-platform>

6. Trabalhos Relacionados

Foram identificados na literatura alguns trabalhos relacionados, os que mais se aproximam aos objetivos desta pesquisa são apresentados nesta Seção.

Granchelli *et al.* [32] apresentaram uma abordagem para recuperação da arquitetura de sistemas baseados em microsserviços, implementada na forma de um protótipo chamado MicroART. Tal abordagem tem o objetivo *auxiliar no gerenciamento e entendimento da complexidade da arquitetura de sistemas baseados em microsserviços*. Durante o estudo, os autores coletam e analisam dados de métricas estáticas e dinâmicas de um sistema de *benchmark* chamado Acme Air. Com base nesses dados, foi possível extrair automaticamente uma versão preliminar da arquitetura do sistema. A partir disso, essa versão preliminar foi analisada e refinada por arquitetos de *software* e então obtida a versão final da arquitetura do sistema estudado.

Engel *et al.* [33] propuseram uma abordagem para avaliar a arquitetura de sistemas baseados em microsserviços. Tal abordagem é fundamentada em princípios como serviços de tamanho pequeno e acoplamento flexível, que foram identificados a partir de pesquisas bibliográficas e entrevistas. A partir da definição desses princípios, foram derivadas as métricas a serem adotadas para *avaliar a arquitetura e identificar pontos críticos que requerem atenção*. Diante disso, foi desenvolvida a ferramenta denominada MAAT (*Microservice Architecture Analysis Tool*), que possibilita obter uma visualização da arquitetura de um sistema baseado em microsserviços. Tal visualização pode ser usada para avaliar a arquitetura com base nos princípios identificados. A ferramenta desenvolvida foi avaliada em um estudo de caso.

Já na pesquisa realizada por Bogner, Wagner e Zimmermann [34], foi proposto um Modelo de Qualidade de Manutenção para Sistemas Baseados em Serviços e Microsserviços, denominado MM4S (*Maintainability Model for Services*). Esse modelo tem a finalidade de *viabilizar a avaliação e controle da capacidade de manutenção de sistemas baseados em serviços e microsserviços*. A primeira versão do modelo MM4S possui uma estrutura hierárquica simples composta por três camadas. A primeira camada representa o atributo de qualidade focal do modelo, que é a manutenibilidade. A segunda camada é formada por propriedades de serviços, relacionadas a características de manutenção de serviços ou do sistema completo. Já a terceira camada é composta por métricas de serviços relacionadas às propriedades da segunda camada.

Logo, a partir da investigação realizada, até o momento, não foram identificados estudos que analisassem o mesmo viés discutido neste trabalho, ou seja, o desenvolvimento de um modelo para caracterização e evolução da arquitetura de sistemas baseados em serviços, visando essencialmente uma análise a nível estrutural. Além disso, ao comparar este estudo aos demais, é possível verificar que o modelo proposto visa caracterizar a arquitetura de um sistema considerando diferentes aspectos estruturais, utilizando métricas interoperáveis e viáveis para o contexto de cada dimensão. Outro diferencial relevante, é que atrelada ao modelo, será desenvolvida uma política para melhor direcionamento da evolução da arquitetura.

Referências

- [1] M. Fowler and J. Lewis, “Microservices,” 2014. Disponível em: martinfowler.com/articles/microservices.html. Acessado: em 10 dez. 2019.

- [2] A. Balalaie, A. Heydarnoori, and P. Jamshidi, “Microservices architecture enables devops: Migration to a cloud-native architecture,” *IEEE Software*, vol. 33, no. 3, pp. 42–52, 2016.
- [3] D. Taibi, V. Lenarduzzi, and C. Pahl, “Architectural patterns for microservices: a systematic mapping study,” in *8th International Conference on Cloud Computing and Services Science (CLOSER 2018)*, pp. 221–232, 2018.
- [4] N. Ford, *The State of Microservices Maturity – Survey Results*. O’Reilly Media, 2018.
- [5] J. Soldani, D. A. Tamburri, and W.-J. V. D. Heuvel, “The pains and gains of microservices: A Systematic grey literature review,” *Journal of Systems and Software*, vol. 146, pp. 215–232, 2018.
- [6] P. Di Francesco, P. Lago, and I. Malavolta, “Architecting with microservices: A systematic mapping study,” *Journal of Systems and Software*, vol. 150, pp. 77–97, 2019.
- [7] M. Fowler, “Monolith First,” 2015. Disponível em: martinfowler.com/bliki/MonolithFirst.html. Acessado em: 12 set. 2019.
- [8] S. Newman, *Building Microservices: Designing Fine-Grained Systems*. O’Reilly Media, 2015.
- [9] Y. Natis and R. Schulte, “Introduction to Service-Oriented Architecture,” tech. rep., Gartner Group, 2003.
- [10] innoQ, “Self-Contained Systems: Assembling Software From Independent Systems,” 2015. Disponível em: scs-architecture.org/index.html. Acessado em: 14 abr. 2020.
- [11] M. Richards, *Software Architecture Patterns*. O’Reilly Media, 2015.
- [12] Z. Mahmood, “The promise and limitations of service oriented architecture,” *International Journal of Computers*, vol. 1, no. 3, pp. 74–78, 2007.
- [13] N. Alshuqayran, N. Ali, and R. Evans, “A systematic mapping study in microservice architecture,” in *9th International Conference on Service-Oriented Computing and Applications (SOCA 2016)*, pp. 44–51, 2016.
- [14] C. Richardson, *Microservices Patterns*. Manning Publicatins Co., 2018.
- [15] Highscalability, “One Team At Uber Is Moving From Microservices To Macroservices,” 2020. Disponível em: highscalability.com/blog/2020/4/8/one-team-at-uber-is-moving-from-microservices-to-macroservic.html. Acessado em: 21 mai. 2020.
- [16] InfoQ, “To Microservices and Back Again,” 2020. Disponível em: www.infoq.com/news/2020/04/microservices-back-again/. Acessado: em 24 jun. 2020.
- [17] I. Nadareishvili, R. Mitra, M. McLarty, and M. Amundsen, *Microservice Architecture: Aligning Principles, parctices, and culture*. O’Reilly Media, 2016.
- [18] C. Paul, R. Kazman, and M. Klein, *Evaluating software architectures: methods and case studies*. Addison-Wesley, 2002.
- [19] H. Cervantes and R. Kazman, *Designing Software Architectures: A practical Approach*. Addison-Wesley, 2016.

- [20] D. Garlan, “Software architecture: a travelogue,” in *Future of Software Engineering (FOSE 2014)*, pp. 29–39, 2014.
- [21] K. Westeinde, “Deconstructing the Monolith: Designing Software that Maximizes Developer Productivity,” 2019. Disponível em: engineering.shopify.com/blogs/engineering/deconstructing-monolith-designing-software-maximizes-developer-productivity. Acessado em: 15 dez. 2019.
- [22] E. Wolff, “Self-contained Systems: A Different Approach to Microservices,” 2016. Disponível em: www.innoq.com/en/articles/2016/11/self-contained-systems-different-microservices/#self-contained-systems. Acessado em: 14 abr. 2020.
- [23] M. Perepletchikov, C. Ryan, K. Frampton, and Z. Tari, “Coupling metrics for predicting maintainability in service-oriented designs,” in *Australian Software Engineering Conference (ASWEC’07)*, pp. 329–338, 2007.
- [24] C. Pahl and P. Jamshidi, “Microservices: A systematic mapping study,” in *6th International Conference on Cloud Computing and Services Science (CLOSER 2016)*, pp. 137–146, 2016.
- [25] F. Neal, R. Parsons, and P. Kua, *Building Evolutionary Architectures: Support Constant Change*. O’Reilly Media, 2017.
- [26] A. de Moura Del Esposte, “A Scalable Microservice-based Open Source Platform for Smart Cities,” Master’s thesis, Universidade de São Paulo, 2018.
- [27] T. D. O. Rosa, A. Goldman, and E. M. Guerra, “How ‘micro’ are your services?,” in *IEEE International Conference on Software Architecture Companion (ICSA-C 2020)*, pp. 75–78, 2020.
- [28] R. C. Martin, *Clean Architecture: A Craftsmans Guide to Software Structure and Design*. Prentice Hall, 2018.
- [29] C. Richardson and F. Smith, *Microservices From Design to Deployment*. Nginx Inc, 2016.
- [30] B. Shim, S. Choue, S. Kim, and S. Park, “A Design Quality Model for Service-Oriented Architecture,” in *15th Asia-Pacific Software Engineering Conference*, pp. 403–410, 2008.
- [31] D. Rud and A. Schmietendorf, “Product metrics for service-oriented infrastructures,” *IWSM/MetriKon*, pp. 161–174, 2006.
- [32] G. Granchelli, M. Cardarelli, P. D. Francesco, I. Malavolta, L. Iovino, and A. D. Salle, “Towards recovering the software architecture of microservice-based systems,” in *IEEE International Conference on Software Architecture Workshops (ICSAW 2017)*, pp. 46–53, 2017.
- [33] T. Engel, M. Langermeier, B. Bauer, and A. Hofmann, “Evaluation of microservice architectures: A metric and tool-based approach,” in *Information Systems in the Big Data Era*, pp. 74–89, 2018.
- [34] J. Bogner, S. Wagner, and A. Zimmermann, “Towards a practical maintainability quality model for serviceand microservice-based systems,” in *11th European Conference on Software Architecture (ECSA 2017)*, pp. 195–198, 2017.