

Estudo Exploratório Sobre o Uso da Arquitetura de Microserviços na Indústria de Software

Tamires G. B. Deprá¹, Douglas R. Tanno¹, Aline Maria M. M. Amaral¹,
Thelma Elita Colanzi¹, Wesley Klewerton Guêz Assunção²

¹Departamento de Informática – Universidade Estadual de Maringá (UEM)
Maringá – PR – Brasil

{tami.g.bsp,douglas.tanno}@gmail.com, {ammmamaral,thelma}@uem.br

²DI – Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)
Rio de Janeiro – RJ – Brasil

wesleyklewerton@gmail.com

Resumo. *A arquitetura de microserviços, embora ainda muito recente, já demonstra grande impacto na indústria de software. Empresas dos mais variados tamanhos e segmentos vêm adotando esse estilo arquitetural. Contudo, poucos trabalhos científicos relatam sobre a adoção da arquitetura de microserviços na indústria de software. Neste trabalho buscou-se identificar se as empresas de software utilizam a arquitetura de microserviços e como a utilizam. Para este propósito, um questionário de avaliação foi aplicado a um grupo de empresas de desenvolvimento de software da cidade de Maringá - PR. Foram convidadas para participar da pesquisa 35 empresas, desse total, 23 empresas aceitaram. Com os dados resultantes da aplicação do questionário, foi possível observar que um número expressivo de empresas utilizam a arquitetura de microserviços (39,1%). Além disso, observou-se que boas práticas para desenvolvimento de microserviços discutidas na literatura são adotadas pelas empresas participantes do estudo.*

Abstract. *The microservice architecture, although very recent, already shows great impact in the software industry. Most varied companies' size and segments adopted this architectural style. However, a small number of scientific papers present discussion about the adoption of microservice architecture in the software industry. In this work, we tried to identify if the software companies use the microservices architecture and how they use it. For this purpose, an evaluation questionnaire was applied to a set of software development companies in the city of Maringá - PR. We invited to participated of this research 35 companies, of that total, 23 companies accepted. Considering the data resulting from the questionnaire application, it was possible to observe that a significant number of companies use the microservice architecture (39.1 %). In addition, it was observed that good practices for the development of microservices discussed in the literature are adopted by the companies participating in the study.*

1. Introdução

A arquitetura de software de um sistema computacional é a estrutura ou conjunto de estruturas do sistema que abrange os componentes do software, as propriedades externamente visíveis desses componentes, e as relações entre eles [Bass et al. 2003]. Existem

diversas abordagens e estilos arquiteturais para a construção de sistemas computacionais [Richards 2005]. Conhecer as vantagens e desvantagens de cada arquitetura, visando a melhor escolha para um dado cenário, é fundamental para o sucesso de um negócio. Dentre os estilos arquiteturais existentes, a arquitetura de microsserviços tem se tornado uma tendência no desenvolvimento moderno de software [Lewis and Fowler 2014].

Microsserviços são serviços pequenos e autônomos que trabalham em conjunto, por meio da comunicação utilizando-se um protocolo leve. A arquitetura de microsserviços propõe o desenvolvimento de um sistema distribuído, onde cada pequena parte do sistema é um (micro)serviço independente, facilitando a criação de novas funcionalidades para adição ao sistema e manutenção das funcionalidades que já se encontram em produção [Newman 2015].

Microsserviços surgiram na indústria, mas só recentemente tem ganhado maior atenção dos pesquisadores de engenharia de software [Lewis and Fowler 2014, Di Francesco et al. 2019]. Na literatura há estudos secundários sobre a migração de sistemas monolíticos para microsserviços [Ponce et al. 2019, Di Francesco et al. 2019, Fritsch et al. 2019], classificação de abordagens de refatoração para modernização de sistemas com microsserviços [Fritsch et al. 2018], e relatos de migração para microsserviços e *surveys* com desenvolvedores da indústria [Carvalho et al. 2019, Fritsch et al. 2019, Di Francesco et al. 2018, Wang et al. 2020]. Apesar do crescente interesse na migração de sistemas monolíticos para arquitetura de microsserviços, tanto na indústria como na academia, pouco se sabe sobre o panorama de uso de arquitetura de microsserviços na indústria de software.

Muitas questões com relação a adoção da arquitetura de microsserviços por empresas de desenvolvimento de software ainda estão em aberto, tais como: (i) quais tecnologias são utilizadas? (ii) qual estratégia para implantação está sendo realizada? (iii) quais motivos que levaram à migração para esse estilo arquitetural?, e (iii) quais critérios são utilizadas para identificação de microsserviços? Nesse sentido, o objetivo deste trabalho foi realizar um estudo exploratório sobre a adoção e uso da arquitetura de microsserviços em empresas da cidade de Maringá - PR. Para tanto, foi elaborado e aplicado um questionário de avaliação em empresas de desenvolvimento de software de Maringá. Das 35 empresas convidadas, 23 empresas responderam voluntariamente ao questionário. O presente trabalho é uma extensão de um estudo inicial [Deprá et al. 2020]. Nesta extensão, incluiu-se novas questões sobre o motivo de algumas empresas adotarem o estilo arquitetural de microsserviços e quais os provedores de infraestrutura e tecnologias as empresas utilizam.

Ao analisar as respostas obtidas, pode-se observar que a maioria das empresas ainda utilizam a arquitetura monolítica, sendo equivalente a 61,9% dos respondentes. Entretanto, grande parte das empresas possui conhecimento sobre a arquitetura de microsserviços e acredita que seria possível a migração da arquitetura monolítica para microsserviços. Entre as empresas que responderam que utilizam microsserviços (39,1%), pode-se verificar uma grande variedade de tecnologias utilizadas, como diferentes *frameworks*, linguagens de programação, serviços de infraestrutura em nuvem, entre outros.

Este trabalho está organizado da seguinte forma: a Seção 2 apresenta uma breve

fundamentação teórica sobre microsserviço; na Seção 3 o método de pesquisa do trabalho é descrito; na Seção 4 são apresentados e discutidos os resultados deste trabalho; finalmente, na Seção 5 são apresentadas as conclusões da pesquisa realizada.

2. Microsserviços

Microsserviços são serviços pequenos e autônomos que trabalham juntos. Recomenda-se que cada microsserviço possua seus próprios dados, sua própria regra de negócio e uma interface bem definida, de modo que cada serviço possa ter seu próprio repositório, controlador de versões, ciclo de vida e controlador de tarefas [Newman 2015]. Além disso, outra proposta desse estilo arquitetural é que cada microsserviço tenha sua própria equipe de desenvolvimento e implantação, possibilitando o desenvolvimento em diversas linguagens e plataformas diferentes, com banco de dados distribuídos e até equipes distribuídas [Meloca et al. 2018, Newman 2015].

É comum comparar a arquitetura de microsserviços com a arquitetura monolítica, pois a arquitetura monolítica é amplamente utilizada e difere em diversos aspectos da arquitetura de microsserviços. Uma aplicação monolítica é feita como uma única unidade lógica executável. A arquitetura de microsserviços deriva da SOA (*Service Oriented Architecture*) cuja finalidade é reduzir o acoplamento entre os diversos módulos de uma aplicação com o propósito de facilitar sua escalabilidade. Na arquitetura monolítica nenhuma funcionalidade do sistema existe e opera por si só [Fowler and Lewis 2014, Meloca et al. 2018]. A Figura 1 apresenta a estrutura de um sistema monolítico e a estrutura de um sistema que utiliza o estilo arquitetural de microsserviços. Na arquitetura monolítica, evidenciada na parte esquerda da figura, é possível observar o conceito de unidade lógica onde as 3 camadas, interface com usuário, regras do negócio e acesso a dados, se comunicam com o banco de dados. Já na arquitetura de microsserviços, na parte direita da figura, observa-se uma interface com o usuário (UI), diversos microsserviços sendo acionados pelas ações do usuário no sistema e a possibilidade de bancos de dados distintos para alguns microsserviços. A arquitetura de microsserviços é particularmente adequada

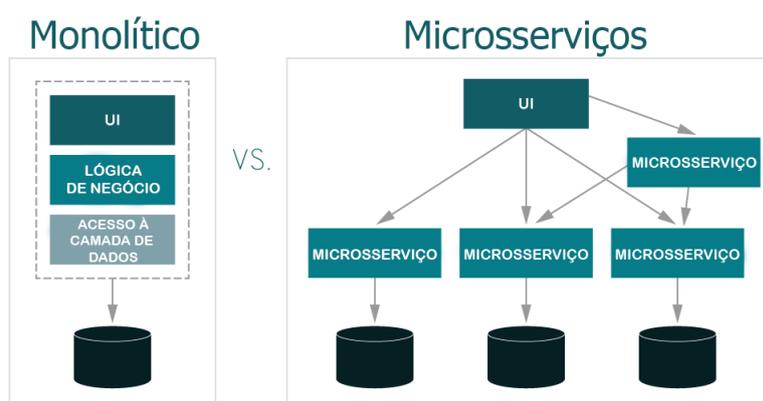


Figura 1. Comparativo entre sistema monolítico e arquitetura de microsserviços. Adaptado de [Hat 2019].

para infraestruturas de nuvem, pois se beneficia muito da elasticidade ativada pela nuvem e do rápido provisionamento de recursos. Fazer uso da arquitetura de microsserviços, no entanto, não é uma tarefa trivial, pois exige o gerenciamento de uma arquitetura distribuída e seus desafios, que incluem latência e falta de confiabilidade da rede, tolerância

a falhas, orquestração de serviços complexos, consistência de dados, gerenciamento de transações e balanceamento de carga. Infraestruturas em nuvem e novas tecnologias desempenham um papel fundamental no uso de arquiteturas de microsserviços e no gerenciamento dos desafios e complexidades associados [Di Francesco et al. 2019].

A utilização da arquitetura de microsserviços oferece diversos benefícios. Dentre eles, destacam-se [Newman 2015]:

Heterogeneidade Tecnológica: em um sistema composto por múltiplos serviços é possível utilizar diferentes tecnologias em cada serviço, o que possibilita a escolha da melhor opção para cada serviço ao invés de utilizar uma única ferramenta para toda a aplicação. Os microsserviços também possibilitam a adoção mais rápida de novas tecnologias, causando menor impacto quando comparados a sistemas monolíticos.

Resiliência: na arquitetura de microsserviços, caso um componente do sistema falhe, mas a falha não desencadeie outras falhas, o problema pode ser isolado e resolvido posteriormente. Com microsserviços, pode-se construir um sistema que controle a falha total de um serviço e degradação de uma função adequadamente, não necessitando que um mesmo serviço seja executado em diversas máquinas ao mesmo tempo para se recuperar da falha.

Escalabilidade: em um sistema monolítico é necessário escalar toda a aplicação. Com microsserviços é possível acionar apenas os serviços que necessitam ser escalados, permitindo executar as demais partes do sistema em um hardware menor e menos potente.

Facilidade no *deploy*: não importa se foi alterada uma linha ou milhares de linhas de código, em um sistema monolítico há a necessidade de realizar o *deploy* da aplicação inteira, todas as vezes que a aplicação sofre alterações. Já nos microsserviços basta realizar o *deploy* exclusivamente dos serviços alterados.

Alinhamento organizacional: com uma base de código reduzida para atuação por equipe, pode-se trabalhar com times menores, distribuídos e mais produtivos.

Composição: uma característica herdada de sistemas distribuídos e sistemas orientados a serviços SOA é a possibilidade de reúso de componentes em diversas partes do sistema.

Otimizando para substituição: toda empresa de médio ou grande porte tem uma parte de código antigo e vital para o funcionamento da aplicação, de difícil manutenção e não facilmente substituível. Com os microsserviços, devido ao tamanho da base de código, essa substituição é facilitada. Caso seja necessário remover ou substituir código, com a arquitetura de microsserviços isso é feito de maneira mais segura, sem danificar o restante do sistema.

Facilidade em realizar testes: outra vantagem da arquitetura de microsserviços é a facilidade em realizar testes unitários, uma vez que o sistema é composto por diversos pequenos serviços [Meloca et al. 2018].

3. Método de Pesquisa

A pesquisa foi conduzida em três etapas, sendo elas: definição do método para coleta dos dados, coleta de dados e análise dos resultados.

3.1. Definição do Método de Coleta de Dados

Esta etapa constitui-se em definir o meio mais adequado para coletar os dados de acordo com a proposta do trabalho. Nesse contexto, foi elaborado um questionário de avaliação, contendo perguntas objetivas e discursivas sobre a arquitetura de microsserviços. As perguntas do questionário foram construídas nos seguintes formatos: (i) múltipla escolha, restringidas a seleção de uma opção, (ii) tipo sim/não, (iii) tipo seleção, onde é possível ao respondente selecionar mais de uma opção, (iv) e questões discursivas.

Considerando o objetivo do trabalho do proposto, o questionário de avaliação¹ possui questões para (1) **caracterização dos participantes e das empresas**, respondidas por todos, (2) questões técnicas e organizacionais referentes ao **uso da arquitetura de microsserviços**, respondidas apenas pelos participantes das empresas que adotam essa arquitetura, e (3) questões exclusivas para os participantes das empresas que não utilizam microsserviços com o objetivo de avaliar seu **conhecimento e intenção de uso dessa arquitetura**.

3.2. Coleta de Dados

A coleta dos dados foi realizada através de um formulário *online*. Dessa forma, os participantes da pesquisa receberam um *link* para responder ao questionário de avaliação. O critério para escolha das empresas participantes da pesquisa foi a sua localização na cidade de Maringá-PR. Outros requisitos como número de funcionários ou ramo do desenvolvimento de software não foram considerados. O questionário foi enviado para 35 empresas, das quais 23 participaram voluntariamente da pesquisa.

Para cada empresa, solicitou-se um representante para responder o questionário. A definição do funcionário responsável por responder o questionário de avaliação ficou a cargo das empresas participantes da pesquisa. Foi solicitado que o respondente tivesse envolvimento (gerencial ou técnico) com a área de desenvolvimento de software da empresa.

3.3. Análise dos Resultados

A última etapa da pesquisa teve como foco a análise e discussão sobre os resultados obtidos com a aplicação do questionário, comparando os dados coletados com os materiais disponíveis na literatura. Essa análise foi realizada quantitativa e qualitativamente, conforme a natureza da informação obtida na resposta do questionário.

4. Resultados e Discussão

Nesta seção é apresentado e discutido o resultado de cada uma das questões do questionário de avaliação elaborado. Como já destacado anteriormente, este questionário foi enviado a 35 empresas de desenvolvimento de software de Maringá-PR, das quais um percentual de 65,7% (23) aceitaram participar da pesquisa e responderam ao questionário.

4.1. Caracterização dos Participantes e das Empresas

A primeira questão de caracterização do questionário de avaliação refere-se ao cargo ocupado pelo respondente na empresa. Nesse contexto 56,5% são desenvolvedores e 17,4%

¹Disponível em: <https://drive.google.com/file/d/1K7ZnDepPlZF-V8np9Pq7WFd9wHJTL4Da/view?usp=sharing>

são analistas de sistema. Quanto ao restante dos respondentes (26,1%), há uma grande variedade de cargos dentro da empresa, tais como: arquiteto, gerente de projetos, coordenador, testador, engenheiro de software, diretor de inovação e gerente de tecnologia.

A questão seguinte refere-se a como a empresa se organiza, se por área de desenvolvimento (interface, negócio, banco de dados, etc.) ou por área do negócio (exemplo: vendas, contabilidade, financeiro, etc.) com times multifuncionais. A maior parte das empresas, 65,2% é dividida em áreas de negócio, 30,4% são divididas por áreas de desenvolvimento e 4,4% por ambas.

O questionário também abordou uma questão sobre a distribuição das equipes de desenvolvimento nas empresas. Como discutido em [Soldani et al. 2018, Lenarduzzi and Sievi-Korte 2018] é comum que empresas que utilizam a arquitetura de microsserviços possuam equipes de trabalho alocadas em locais diferentes, pois a divisão de equipes responsáveis por cada microsserviço é facilitada. Do total das empresas respondentes, 52,2% utilizam o desenvolvimento distribuído, sendo que 55,6% dessas empresas utilizam microsserviços.

Em seguida, foi questionado se as empresas adotam a arquitetura de microsserviços. Um percentual de 39,1% empresas usa microsserviços, enquanto que 60,9% delas não utiliza esse estilo arquitetural. Após essa pergunta, o questionário foi dividido em duas seções diferentes, uma para as empresas que utilizam a arquitetura de microsserviços (Seção 4.2) e outra seção para as empresas que utilizam outras arquiteturas para o desenvolvimento de seus sistemas (Seção 4.3).

4.2. Uso da Arquitetura de Microsserviços

A primeira questão do questionário de avaliação direcionada às 9 empresas que adotam a arquitetura de microsserviços foi sobre a estratégia utilizada para implantação desse estilo arquitetural. A distribuição das respostas dadas a esta questão pode ser visualizada na Figura 2. A maioria das empresas, 55,6%, optou pela migração gradual do sistema e 11,1% alegam que estavam conduzindo a migração dos seus sistemas durante o período em que o questionário foi respondido, também de forma gradual. Tais resultados corroboram a discussão apresentada em [Balalaie et al. 2015] que afirma que a migração do sistema para a arquitetura de microsserviços não é simples e deveria ser realizada de forma incremental (*strangler pattern*) e em muitos passos sem afetar o usuário final. Nesse sentido, pode-se observar que as empresas do estudo, assim como sugerido na literatura, realizam a migração para a arquitetura de microsserviços gradualmente.

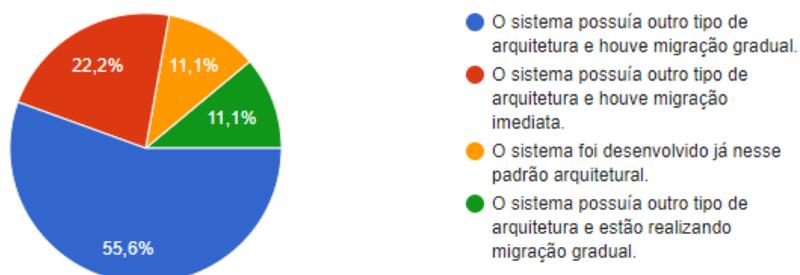


Figura 2. Estratégia para implantação da arquitetura de microsserviços.

Em seguida, as empresas foram questionadas sobre os motivos que as levaram a

migrar para a arquitetura de microsserviços. Os resultados desta questão (apresentados na Figura 3) demonstram que os fatores escalabilidade, manutenção e evolução foram os principais motivos para a migração, respondidos por 77,8% das empresas, seguidos por flexibilidade tecnológica com 66,7%, *deploy* independente e autônomo com 44,4%, tempo reduzido de entrega com 44,4%, e baixo acoplamento com 33,3%.

Complementando a questão acima apresentada, uma questão sobre critérios utilizados para a tomada de decisão sobre a extração de microsserviços também foi feita aos participantes da pesquisa. Nesse contexto, a escalabilidade também foi apontada como critério para decisão de extração, por todas as empresas que utilizam a arquitetura de microsserviços. O segundo critério mais apontado foi acoplamento, seguido pela reusabilidade. A Figura 4 apresenta o número de vezes que cada critério foi selecionado na pesquisa.

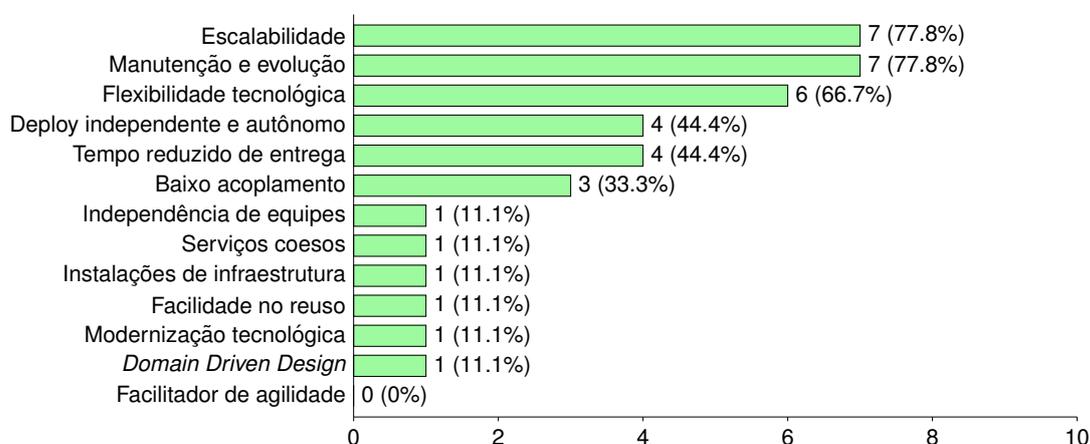


Figura 3. Motivos que levaram a empresa a migrar para microsserviço.

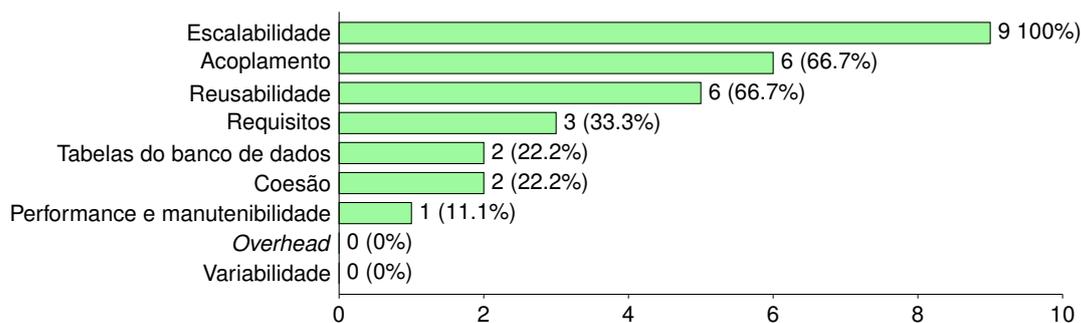


Figura 4. Critérios para decisão sobre a extração dos microsserviços.

Segundo Meloca [Meloca et al. 2018], um dos grandes benefícios da adoção da arquitetura de microsserviços é a obtenção de um bom nível de escalabilidade do sistema, pois é possível aprimorar e expandir um microsserviço sem a necessidade de modificar outras partes do sistema, acelerando processos e poupando custos, tempo e trabalho. Além disso, [Newman 2015] discute que uma das principais características da arquitetura de microsserviços é sua autonomia, uma vez que toda a comunicação entre os serviços é feita via chamadas através da rede, para forçar a separação entre eles e evitar alto acoplamento. Pode-se observar nas respostas às duas questões anteriores que tanto o formato da migração quanto os motivos/critérios de extração adotados pelas empresas participantes da pesquisa vão ao encontro do que é também discutido e proposto na literatura.

Com o objetivo de consolidar algumas das informações obtidas pelas empresas que utilizam microsserviços, a Tabela 1 apresenta a relação entre os dados sobre a escolha da arquitetura e os motivos que levaram a empresa a realizar a migração. No total, cinco empresas realizaram a migração gradual para microsserviços a partir outro tipo de arquitetura, duas empresas realizaram a migração imediata, uma empresa estava realizando a migração no período em que o questionário foi aplicado, e uma empresa construiu o software já utilizando a arquitetura de microsserviços.

Deve-se ressaltar que, independente do tipo de estratégia de migração para arquitetura de microsserviços, o principal motivo para a decisão sobre a extração de microsserviços foi a escalabilidade. Além disso, pode-se notar que o segundo fator mais relevante para extração dos microsserviços por empresas que realizaram a migração gradual foi a reusabilidade. Já para as empresas que realizaram a migração imediata foram os requisitos. O acoplamento também foi uma escolha das empresas que estão passando por migração ou que o sistema já foi construído utilizando a arquitetura de microsserviços.

Tabela 1. Estratégia e motivos para migração para microsserviços.

Estratégia de migração	Motivo da migração	Nº de empresas
Migração gradual	Manutenção e evolução facilitadas	5
	Escalabilidade	4
	Flexibilidade tecnológica	4
	<i>Deploy</i> independente e autônomo	2
	Tempo reduzido de entrega	2
	Independência dos times	1
	Serviços coesos	1
	Baixo acoplamento	1
	Instalações de infraestrutura	1
	Facilidade no reúso	1
	<i>Domain Driven Design</i>	1
Migração imediata	Escalabilidade	1
	<i>Deploy</i> independente e autônomo	1
	Flexibilidade tecnológica	1
	Tempo reduzido de entrega	1
Em processo de migração	Escalabilidade	1
	Manutenção e evolução facilitadas	1
	Baixo acoplamento	1
	Flexibilidade tecnológica	1
Usa microsserviços desde o início	Escalabilidade	1
	<i>Deploy</i> independente e autônomo	1
	Manutenção e evolução facilitadas	1
	Baixo acoplamento	1
	Flexibilidade tecnológica	1
	Tempo reduzido de entrega	1

A Tabela 2 relaciona os dados sobre a escolha da arquitetura de microsserviços e os critérios adotados pelas empresas para a escolha de quais partes de sua(s) aplicação(ões) serão extraídas como microsserviços. Assim como na Tabela 1, pode-se observar que o critério mais citado pelas empresas foi a escalabilidade, seguido novamente pelo critério de reusabilidade. Tais resultados demonstram que, para este grupo de

empresas, a adoção da arquitetura de microsserviços está diretamente relacionada com a necessidade de escalabilidade de suas aplicações e/ou serviços.

Tabela 2. Estratégia de migração e critérios para extração dos microsserviços.

Estratégia de migração	Critério para extração	Nº de empresas
Migração gradual	Escalabilidade	5
	Reusabilidade	4
	Acoplamento	3
	Coessão	1
	Requisitos	1
Migração imediata	Escalabilidade	2
	Requisitos	2
	Tabelas do banco de dados	2
	Acoplamento	1
	Coessão	1
Em processo de migração	Escalabilidade	1
	Acoplamento	1
	Performance, manutenibilidade	1
Usa microsserviços desde o início	Escalabilidade	1
	Reusabilidade	1
	Acoplamento	1

Uma questão sobre o número de linguagens de programação utilizadas no desenvolvimento de sistemas que tem como base a arquitetura de microsserviços foi também apresentada às empresas (Figura 5). Um percentual de 33,3% das empresas afirmam utilizar duas linguagens, enquanto que 33,3% utilizam apenas uma linguagem e 22,2% quatro linguagens. Um respondente afirmou desconhecer a quantidade de linguagens utilizada pela empresa, uma vez que não participou ativamente no desenvolvimento dos microsserviços. O fato de a maioria das empresas utilizar mais de uma linguagem de programação confirma a heterogeneidade tecnológica oferecida pelo uso de microsserviços [Newman 2015].

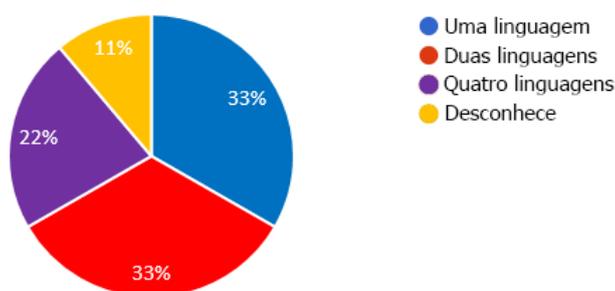


Figura 5. Número de linguagens de programação utilizadas.

A questão seguinte teve como objetivo verificar se as empresas fazem uso de serviços de infraestrutura em nuvem. Nesse contexto, 100% das empresas afirmam fazer uso deste tipo de serviço. Adicionalmente, foi questionado sobre quais provedores de infraestrutura em nuvem são utilizados. Com base nas respostas obtidas, e apresentadas na Figura 6, pode-se destacar que 66,7% utilizam Amazon Web Services (AWS), seguido

por 11,1% que fazem uso de AWS com Digital Ocean, 11,1% utilizam Microsoft Azure, e 11,1% utilizam Oracle Cloud com uma infraestrutura própria da empresa.

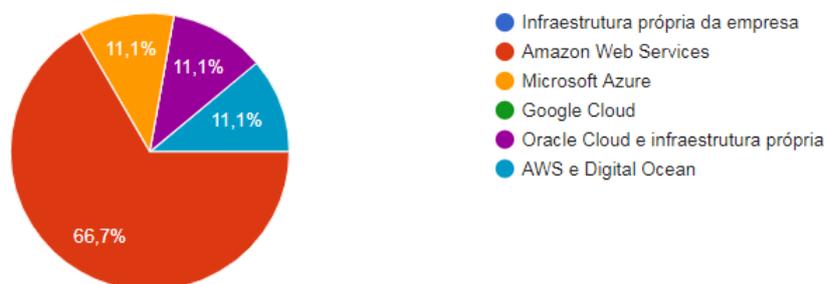


Figura 6. Utilização de serviços de infraestrutura em nuvem.

Outra questão relevante relacionada a aspectos técnicos envolvidos na adoção de microsserviços refere-se aos *frameworks* de desenvolvimento utilizados pelas empresas. Na Figura 9 pode-se observar, dentre as opções propostas no questionário de avaliação, o número de vezes que cada *framework* foi selecionado na pesquisa. O *framework* ASP .Net foi o mais utilizado, indicado por quatro das nove empresas que utilizam a arquitetura de microsserviços, e o Spring Boot foi o segundo mais utilizado, indicado por três empresas.

Ainda considerando aspectos técnicos, uma questão sobre o meio de comunicação utilizado entre os microsserviços foi feita aos participantes desta pesquisa. A Figura 7 apresenta as respostas a esta questão, que demonstram uma clara predominância do uso da comunicação por APIs REST com 66,7% das respostas. Esse resultado é retratado em trabalhos como [Balalaie et al. 2015] e [Fowler and Lewis 2014] que discutem que microsserviços visam transformar sistemas de software em pacotes de pequenos serviços, cada um entregável em uma plataforma diferente e rodando seus próprios processos através de uma comunicação com mecanismos leves como, por exemplo, RESTFull APIs ou API HTTP.

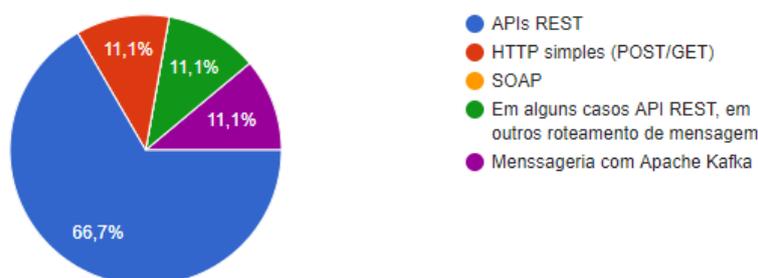


Figura 7. Tipo de comunicação utilizada entre os microsserviços.

Com o objetivo de avaliar como as empresas que utilizam microsserviços organizam sua estrutura de banco de dados, foi elaborada uma questão sobre a estrutura de banco de dados dessas empresas. Na Figura 8, observa-se que a estrutura mais utilizada é a de um banco de dados dedicado para cada microsserviço com 55,6 % das respostas dos participantes da pesquisa. Este tipo de decisão é preconizada pela literatura. Como mencionado anteriormente, cada microsserviço deve possuir seus próprios dados, sua própria regra de negócio e uma interface bem definida, de modo que cada microsserviço deve

ter seu próprio repositório, controlador de versões, ciclo de vida e controlador de tarefas [Meloca et al. 2018]. Levando isso em conta, pode-se entender que, ao optar por um banco de dados para cada microserviço, espera-se ter melhor desempenho e menor acoplamento entre os serviços, benefícios almejados ao utilizar microserviços.

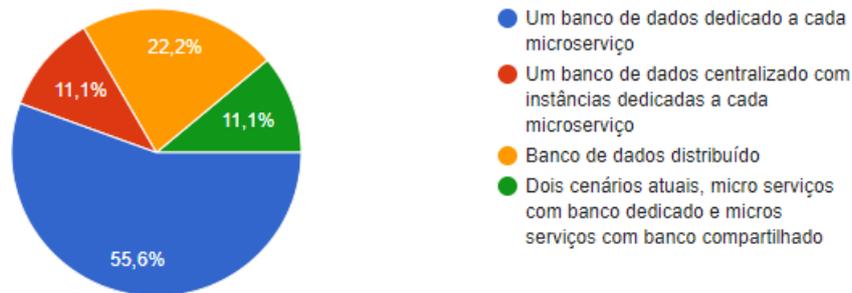


Figura 8. Estrutura de banco de dados para suporte aos microserviços.

Uma questão sobre as ferramentas de integração contínua utilizadas pelas empresas também foi incluída, sendo que as opções de resposta eram: *TeamCity*, *Jenkins*, *Azure*, *nenhuma* e *outro*, com a possibilidade do respondente citar a ferramenta utilizada. Como pode ser visto na Figura 10, a ferramenta mais utilizada é a *Jenkins*. Essa questão é relevante, pois uma característica importante da arquitetura de microserviços é a facilidade do *deploy*, o que geralmente ocorre por meio de integração contínua, e as ferramentas mencionadas têm a proposta de facilitar essa integração contínua. Segundo [Balalaie et al. 2015], a arquitetura de microserviços está intimamente ligada ao conceito de entrega contínua. Na prática, isso só é possível ao adotar ferramentas apropriadas que promovam essa integração do projeto com o *deploy*.

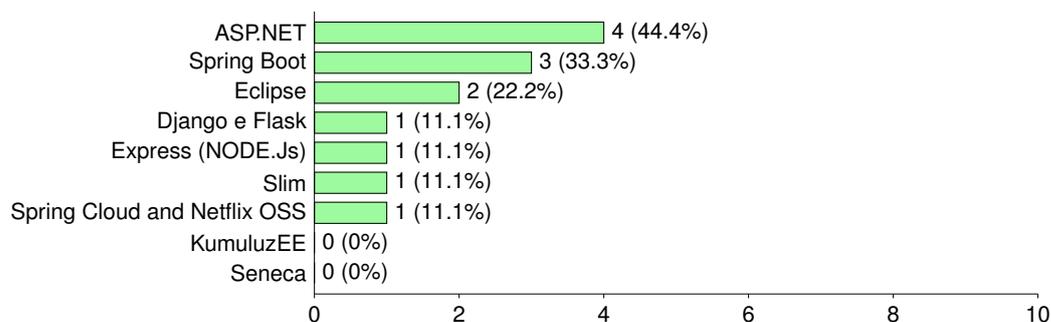


Figura 9. Frameworks utilizados para desenvolvimento de microserviços.

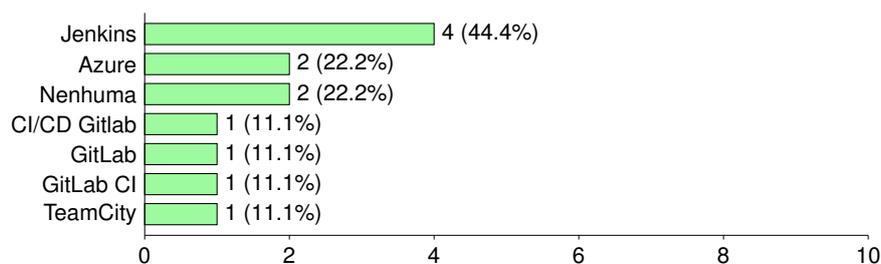


Figura 10. Ferramentas para realização de integração contínua.

O registro de *logs* é gerado e incrementado ao longo do tempo, e possui informações que permitem diagnosticar anormalidades em relação ao propósito do sis-

tema e questões de segurança e acessibilidade. Para [Brown 2016], os seguintes conceitos são importantes na implementação da arquitetura de microsserviços: ID de Correlação, Agregador de *log* e Disjuntor. Mais especificamente sobre o agregador de *log*, ele complementa os IDs de Correlação permitindo que os *logs* de diversos microsserviços diferentes sejam agregados em um único repositório que possa ser pesquisado. Juntos, eles permitem a depuração eficiente e compreensível de microsserviços, independentemente do número de serviços ou da profundidade de cada pilha de chamada.

A Figura 11 apresenta as respostas dos participantes da pesquisa a uma questão sobre a ferramenta de controle de *log* utilizada. As opções de seleção foram: *Flume*, *ELK*, *Fluentd*, *nenhuma* e *outra*, com opção de indicação da ferramenta utilizada. A maior parte das empresas, como pode ser observado na figura, utiliza alguma ferramenta de controle de *log*, porém não há destaque para nenhuma em específico. Um dos respondentes marcou a opção *outra* e descreveu a forma do trabalho como segue: “*Registros de eventos no banco de dados — log dos frameworks de roteamento*”.

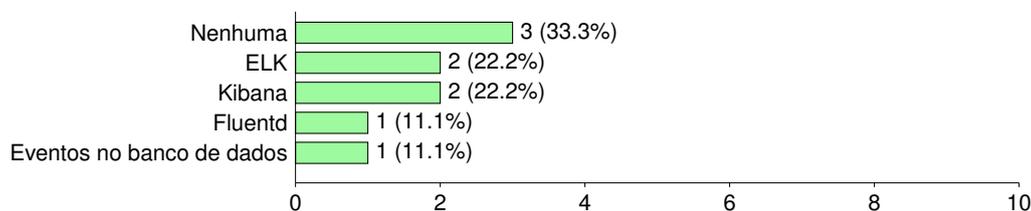


Figura 11. Ferramentas para controle de *log*.

Consolidando as respostas sobre os aspectos técnicos do desenvolvimento, comunicação e na manutenção dos microsserviços, a Tabela 3 apresenta algumas tecnologias utilizadas pelas empresas, como infraestrutura em nuvem, *frameworks*, meio de comunicação entre microsserviços, ferramentas para integração contínua e para o controle de *log*. É possível verificar que, mesmo muitas empresas utilizando tecnologias em comum, como *Amazon Web Services*, *ASP.NET*, *APIs REST*, *Jenkins* e *Kibana*, ainda existe uma grande variedade de tecnologias utilizadas pelas empresas para a adoção da arquitetura de microsserviços.

Duas questões da seção de perguntas para empresas que utilizam a arquitetura microsserviços foram discursivas, onde a empresa poderia responder de forma livre. Dentre as nove empresas que utilizam a arquitetura de microsserviços cinco responderam essas questões. Esses respondentes foram nomeados como R1, R2, R3, R4 e R5.

A primeira questão discursiva relaciona-se às preocupações da empresa com a granularidade dos microsserviços. Com base nas respostas pode-se observar que a principal preocupação das empresas é o domínio dos serviços e a manutenção do código. É possível observar esse padrão na resposta fornecida por R1 “*Manutenção de código/estrutura do banco, balanceamento de carga*” ou ainda pela resposta fornecida por R2 “*Responsabilidade única, a orientação é que cada microsserviço seja responsável por um contexto específico*”. Para [Newman 2015], microsserviços são serviços pequenos e autônomos que trabalham em conjunto. Sendo assim, um princípio base da arquitetura de microsserviços é justamente a granularidade dos serviços.

A segunda questão discursiva refere-se sobre o gerenciamento do *overhead* (balanceamento de carga). Pode-se observar que a maior parte das empresas adota como es-

Tabela 3. Tecnologias utilizadas no desenvolvimento e manutenção de microsserviços.

Tipo de tecnologia	Tecnologia utilizada	Nº de empresas
Infraestrutura em nuvem	Amazon Web Services (AWS)	6
	Microsoft Azure	1
	AWS e Digital Ocean	1
	Oracle Cloud e infraestrutura própria	1
<i>Framework</i>	ASP.NET	4
	Spring Boot	3
	Eclipse	2
	Express (Node.JS)	1
	Spring Cloud and NetFlix OSS	1
	Django e Flask	1
	Slim	1
Comunicação	APIs REST	7
	Mensageria com Apache Kafka	1
	SOAP	1
	HTTP simples (POST/GET)	1
Integração contínua	Jenkins	4
	Azure	2
	TeamCity	1
	Gitlab	1
	CI/CD Gitlab	1
	Gitlab CI	1
	Nenhuma	1
Controle de <i>log</i>	Kibana	2
	Fluentd	1
	ELK	1
	Registro de eventos no banco de dados	1
	Nenhuma	1

tratégia utilizar ferramentas para o controle do *overhead*. Na resposta de R3, por exemplo, "*Frameworks escolhidos para roteamento/balanceamento*", ou ainda na resposta de R4 "*Utilizamos Kubernetes rodando no EKS da Amazon*" e na resposta de R5 "*Nginx, AWS EC2 ELB*". Como destacado em [Balalaie et al. 2015], para ser escalável, um aplicativo deve ser capaz de distribuir a carga de um serviço individual entre suas várias instâncias. Esse é o dever de um balanceador de carga e, nesse caso, deve obter instâncias disponíveis do componente de descoberta de serviço. Como a escalabilidade é um dos benefícios da arquitetura de microsserviços, a empresa deve se preocupar com o *overhead* do sistema.

4.3. Conhecimento e Intenção de Uso de Microsserviços

Foram elaboradas questões específicas para as empresas que não utilizam a arquitetura de microsserviços com o objetivo de avaliar se esse estilo arquitetural é, pelo menos, conhecido pelos participantes da pesquisa e se, além disso, existe intenção por partes das empresas em usar microsserviços em seus sistemas futuramente.

A primeira questão foi se a empresa possui conhecimento sobre a arquitetura de microsserviços, sendo que a grande maioria, 71,4%, apesar de não utilizar, alegou ter

conhecimento sobre microsserviços, enquanto 28,6% afirmou não conhecer.

A segunda questão aplicada aos participantes da pesquisa refere-se a arquitetura utilizada atualmente pela empresa. Foi observado que todas as empresas que não adotam a arquitetura de microsserviços utilizam uma arquitetura monolítica. Nenhuma das empresas da amostra desta pesquisa utilizam SOA, apesar de ser um tipo de arquitetura popular e comumente utilizada.

A última questão objetivou verificar a possibilidade e/ou intenção de uma migração dos sistemas da empresa para a arquitetura de microsserviços. Nesse sentido, 42,9% dos respondentes acreditam que *talvez seja possível* a migração, 28,6% indicam que *é possível* a migração e outros 28,6% acreditam que *não é possível* a migração. Das empresas que não utilizam a arquitetura de microsserviços, 71,4% alegam ter conhecimento sobre a arquitetura de microsserviços, 42,9% acreditam que talvez seja possível realizar a migração de seus sistemas monolíticos para esse estilo arquitetural, e 28,9% acreditam que é possível realizar tal migração.

Essas informações indicam um cenário positivo para uma futura adoção desse estilo arquitetural pelas empresas participantes da pesquisa. Nesse contexto, várias pesquisas vêm sendo realizadas com o objetivo de oferecer suporte para que empresas migrem para a arquitetura de microsserviços, apresentando soluções técnicas para a migração [Fan and Ma 2017, Balalaie et al. 2018], abordagens para a identificação de microsserviços [Carvalho et al. 2020, Zhang et al. 2020], e relatos de experiência de migração para este estilo arquitetural [Fowler 2016, Tizzei et al. 2017, Gouigoux and Tamzalit 2017, Knoche and Hasselbring 2018, Assunção et al. 2021].

5. Conclusão

Este trabalho teve como objetivo realizar um estudo exploratório sobre o uso da arquitetura de microsserviços na indústria de software, considerando o contexto das empresas da cidade de Maringá - PR. Com base nos resultados apresentados e na análise dos mesmos, pode-se concluir que há um número expressivo de empresas de software dentre as que responderam o questionário que utilizam a arquitetura de microsserviços, sendo 9 das 23 (39,1%) que participaram da pesquisa. Além disso, conforme a análise dos dados, as empresas que utilizam a arquitetura de microsserviços, em geral, trabalham conforme as boas práticas já apresentadas e descritas na literatura. Das empresas que não utilizam esse estilo arquitetural, observou-se que há conhecimento sobre o tema e muitas empresas acreditam que uma migração seria viável.

É possível expandir a pesquisa ao âmbito nacional, possibilitando a comparação entre a forma de trabalho regional, observada na cidade de Maringá - PR, e a forma de trabalho encontrada em outras regiões do país. Também pode ser expandida esta pesquisa para outros aspectos não abordados ou não explorados profundamente nesta pesquisa, tais como aspectos da migração dos sistemas, os benefícios obtidos com o uso da arquitetura de microsserviços, entre outros tópicos. Como forma de divulgação dos resultados da pesquisa, os autores pretendem promover *lives* e palestras no futuro para discussão do tema.

Referências

- Assunção, W. K. G., Colanzi, T. E., Carvalho, L., Pereira, J. A., Lima, A. G. M. J., and Lucena, C. (2021). A multi-criteria strategy for redesigning legacy features as microservices: An industrial case study. In *The 28th IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*.
- 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.
- Balalaie, A., Heydarnoori, A., Jamshidi, P., Tamburri, D. A., and Lynn, T. (2018). Microservices migration patterns. *Software: Practice and Experience*, 48(11):2019–2042.
- Bass, L., Clements, P., and Kazman, R. (2003). *Software architecture in practice*. Addison-Wesley Professional.
- Brown, K. (2016). Além de palavras da moda: um breve histórico sobre padrões de microsserviços, <https://developer.ibm.com/br/articles/cl-evolution-microservices-patterns/>, abril.
- Carvalho, L., Garcia, A., Assunção, W. K. G., de Mello, R., and de Lima, M. J. (2019). Analysis of the criteria adopted in industry to extract microservices. In *7th International Workshop on Conducting Empirical Studies in Industry and 6th International Workshop on Software Engineering Research and Industrial Practice*, pages 22–29. IEEE.
- Carvalho, L., Garcia, A., Colanzi, T. E., Assunção, W. K. G., Pereira, J. A., Fonseca, B., Ribeiro, M., Lima, M. J., and Lucena, C. (2020). On the performance and adoption of search-based microservice identification with tomicroservices. In *The 36th IEEE International Conference on Software Maintenance and Evolution (ICSME)*.
- Deprá, T., Tanno, D., Amaral, A., and Colanzi, T. (2020). Estudo exploratório sobre o uso da arquitetura de microsserviços em empresas da cidade de Maringá-PR. In *IV Escola Regional de Engenharia de Software*, pages 11–20, Porto Alegre, RS, Brasil. SBC.
- Di Francesco, P., Lago, P., and Malavolta, I. (2018). Migrating towards microservice architectures: an industrial survey. In *International conference on software architecture*, pages 29–2909. IEEE.
- Di Francesco, P., Lago, P., and Malavolta, I. (2019). Architecting with microservices: A systematic mapping study. *Journal of Systems and Software*, 150:77–97.
- Fan, C. and Ma, S. (2017). Migrating monolithic mobile application to microservice architecture: An experiment report. In *2017 IEEE International Conference on AI & Mobile Services (AIMS)*, pages 109–112. IEEE.
- Fowler, M. and Lewis, J. (2014). Microservices. <https://martinfowler.com/articles/microservices.html>, abril.
- Fowler, S. (2016). *Production-Ready Microservices*. O'Reilly Media.
- Fritzsch, J., Bogner, J., Wagner, S., and Zimmermann, A. (2019). Microservices migration in industry: Intentions, strategies, and challenges. In *IEEE International Conference on Software Maintenance and Evolution*, pages 481–490. IEEE.

- Fritzsche, J., Bogner, J., Zimmermann, A., and Wagner, S. (2018). From monolith to microservices: a classification of refactoring approaches. In *International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*, pages 128–141. Springer.
- Gouigoux, J. and Tamzalit, D. (2017). From monolith to microservices: Lessons learned on an industrial migration to a web oriented architecture. In *International Conference on Software Architecture Workshops (ICSAW)*, pages 62–65.
- Hat, R. (2019). O que são os microsserviços?. <https://www.redhat.com/pt-br/topics/microservices/what-are-microservices>, abril.
- Knoche, H. and Hasselbring, W. (2018). Using microservices for legacy software modernization. *IEEE Software*, 35(3):44–49.
- Lenarduzzi, V. and Sievi-Korte, O. (2018). On the negative impact of team independence in microservices software development. In *Proceedings of the 19th International Conference on Agile Software Development: Companion*, New York, NY, USA. Association for Computing Machinery.
- Lewis, J. and Fowler, M. (2014). Microservices: a definition of this new architectural term.
- Meloca, R. M., Ré, R., and Schwerz, A. L. (2018). An analysis of frameworks for microservices. *2018 XLIV Latin American Computer Conference (CLEI)*, pages 542–551.
- Newman, S. (2015). *Building microservices: designing fine-grained systems*. "O'Reilly Media, Inc."
- Ponce, F., Márquez, G., and Astudillo, H. (2019). Migrating from monolithic architecture to microservices: A rapid review. In *38th International Conference of the Chilean Computer Science Society (SCCC)*, pages 1–7. IEEE.
- Richards, M. (2005). *Software Architecture Patterns*. O'Reilly Media.
- Soldani, J., Tamburri, D., and Heuvel, W.-J. (2018). The pains and gains of microservices: A systematic grey literature review. *Journal of Systems and Software*, 146.
- Tizzei, L. P., Nery, M., Segura, V. C. V. B., and Cerqueira, R. (2017). Using microservices and software product line engineering to support reuse of evolving multi-tenant SaaS. In *21st International Systems and Software Product Line Conference (SPLC)*, pages 205–214.
- Wang, Y., Kadyala, H., and Rubin, J. (2020). Promises and challenges of microservices: an exploratory study. *Empirical Software Engineering*, pages 1–45.
- Zhang, Y., Liu, B., Dai, L., Chen, K., and Cao, X. (2020). Automated microservice identification in legacy systems with functional and non-functional metrics. In *2020 IEEE International Conference on Software Architecture (ICSA)*, pages 135–145.