



Unioeste - Universidade Estadual do Oeste do Paraná
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
Colegiado de Ciência da Computação
Curso de Bacharelado em Ciência da Computação

Mapeamento Sistemático das Abordagens para Criação de Microserviços

Julian Alberto Piovesan Ruiz Diaz

CASCADEL
2018

Julian Alberto Piovesan Ruiz Diaz

Mapeamento Sistemático das Abordagens para Criação de Microserviços

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação, do Centro de Ciências Exatas e Tecnológicas da Universidade Estadual do Oeste do Paraná - Campus de Cascavel

Orientador: Prof. Ivonei Freitas da Silva

CASCADEL
2018

Julian Alberto Piovesan Ruiz Diaz

Mapeamento Sistemático das Abordagens para Criação de Microserviços

Monografia apresentada como requisito parcial para obtenção do Título de Bacharel em Ciência da Computação, pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel, aprovada pela Comissão formada pelos professores:

Prof. Ivonei Freitas da Silva (Orientador)
Colegiado de Ciência da Computação,
UNIOESTE

Prof. Edmar Bellorini
Colegiado de Ciência da Computação,
UNIOESTE

Prof. André Luiz Brun
Colegiado de Ciência da Computação,
UNIOESTE

Cascavel, 15 de dezembro de 2018

Lista de Figuras

| | | |
|-----|--|----|
| 2.1 | Aplicação Monolítica. Adaptado de (LEWIS; FOWLER, 2014) | 5 |
| 2.2 | Arquitetura de Microserviços. Adaptado de (LEWIS; FOWLER, 2014) | 6 |
| 2.3 | Gerenciamento de dados aplicação monolítica. Adaptado de (LEWIS; FOWLER, 2014) | 9 |
| 2.4 | Gerenciamento de dados nos microserviços. Adaptado de (LEWIS; FOWLER, 2014) | 9 |
| 3.1 | Crescimento de publicações sobre microserviços na ACM. | 15 |
| 3.2 | String de busca utilizada na ACM | 17 |
| 3.3 | String de busca utilizada na IEEEExplore e ScienceDirect | 18 |
| 6.1 | Primeiro cruzamento entre categorias das facetas | 53 |
| 6.2 | Segundo cruzamento entre categorias das facetas | 54 |
| 6.3 | Terceiro cruzamento entre categorias das facetas | 55 |
| 6.4 | Distribuição dos artigos entre os anos. | 56 |
| 6.5 | Distribuição dos artigos entre as bases de dados antes da filtragem. | 58 |
| 6.6 | Distribuição dos artigos entre as bases de dados após filtragem. | 59 |

Lista de Tabelas

| | | |
|-----|---|----|
| 3.1 | Tradução das palavras-chave. | 16 |
| 3.2 | Quantidade de estudos obtidos na execução do piloto | 21 |
| 4.1 | Listagem final de artigos | 28 |
| 6.1 | Distribuição das categorias das duas primeiras facetas entre os trabalhos | 50 |
| 6.2 | Distribuição das categorias das três últimas facetas entre os trabalhos | 52 |

Lista de Abreviaturas e Siglas

| | |
|---------|--|
| AIMC | <i>International Workshop on Automated Incident Management in Cloud</i> |
| AIMS | <i>International Conference on AI & Mobile Services</i> |
| API | <i>Application Programming Interface</i> |
| APSEC | <i>Asia-Pacific Software Engineering Conference</i> |
| CAIPT | <i>Conference on Computer Applications and Information Processing Technology</i> |
| CNSM | <i>International Conference on Network and Service Management</i> |
| CVDS | <i>Ciclo de Vida de Desenvolvimento de Software</i> |
| DMTF | <i>Distributed Management Task Force</i> |
| ECBS | <i>European Conference on the Engineering of Computer-Based Systems</i> |
| ETFA | <i>International Conference on Emerging Technologies and Factory Automation</i> |
| FAMS | <i>International Symposium on Frontiers in Ambient and Mobile Systems</i> |
| FRUCT | <i>Conference of Open Innovations Association</i> |
| HPCS | <i>International Conference on High Performance Computing & Simulation</i> |
| ICASI | <i>International Conference on Applied System Innovation</i> |
| ICCCA | <i>International Conference on Computing, Communication and Automation</i> |
| ICCT | <i>International Conference on Communication Technology</i> |
| ICCUBEA | <i>International Conference on Computing, Communication, Control and Automation</i> |
| ICPS | <i>Industrial Cyber-Physical Systems</i> |
| ICT-DM | <i>International Conference on Information and Communication Technologies for Disaster</i> |
| iiWAS | <i>Information Integration and Web-based Applications and Services</i> |
| IMCEC | <i>Information Management, Communicates, Electronic and Automation Control Conferen</i> |
| INDIN | <i>International Conference on Industrial Informatics</i> |
| IoT | <i>Internet of Things</i> |
| ISICO | <i>Information Systems International Conference</i> |
| ISPA | <i>International Symposium on Parallel and Distributed Processing with Applications</i> |
| IUCC | <i>International Conference on Ubiquitous Computing and Communications</i> |
| MSA | <i>Microservice Architecture</i> |
| NRDC | <i>Natural Resources Defense Council</i> |
| PaaS | <i>Platform as a Service</i> |
| QRS-C | <i>International Conference on Software Quality, Reliability and Security Companion</i> |
| REST | <i>Representational State Transfer</i> |
| RTF | <i>Rich Text Format</i> |
| RWK | <i>Real World Knowledge</i> |
| SaaS | <i>Software-as-a-Service</i> |

| | |
|--------|--|
| SACI | <i>Symposium on Applied Computational Intelligence and Informatics</i> |
| SF | <i>Service Fabric</i> |
| SOA | <i>Service Oriented Architecture</i> |
| TAR | <i>Technical Action Research</i> |
| TI | <i>Tecnologia da Informação</i> |
| UKRCON | <i>Ukraine Conference on Electrical and Computer Engineering</i> |
| WoO | <i>Web of Objects</i> |
| XSEDE | <i>Extreme Science and Engineering Discovery Environment</i> |

Sumário

| | |
|---|------------|
| Lista de Figuras | iv |
| Lista de Tabelas | v |
| Lista de Abreviaturas e Siglas | vi |
| Sumário | vii |
| Resumo | xii |
| 1 Introdução | 1 |
| 1.1 Motivação do Trabalho | 1 |
| 2 Referencial Teórico | 4 |
| 2.1 Microserviço | 6 |
| 2.2 Mapeamento Sistemático | 10 |
| 2.3 Trabalhos Relacionados | 11 |
| 3 Planejamento do Mapeamento Sistemático | 14 |
| 3.1 Pergunta de Pesquisa | 14 |
| 3.2 Palavras-chave | 15 |
| 3.3 Bases de Dados | 16 |
| 3.4 Strings de Busca | 17 |
| 3.5 Critérios de Inclusão e Exclusão | 18 |
| 3.5.1 Inclusão | 18 |
| 3.5.2 Exclusão | 19 |
| 3.6 Método de Classificação | 19 |
| 3.6.1 Facetas e Categorias | 20 |
| 3.7 Piloto | 21 |

| | | |
|----------|---|-----------|
| 4 | Execução do Mapeamento Sistemático | 23 |
| 4.1 | Resultados das Buscas | 23 |
| 4.1.1 | IEEEExplore | 23 |
| 4.1.2 | ACM Digital Library | 24 |
| 4.1.3 | ScienceDirect | 24 |
| 4.2 | Triagem dos Artigos | 24 |
| 4.2.1 | Aplicação dos Critérios de Inclusão e de Exclusão | 25 |
| 5 | Análise dos Artigos | 29 |
| 5.1 | Extração de Dados | 29 |
| 5.1.1 | <i>A microservice based reference architecture model in the context of enterprise architecture</i> | 29 |
| 5.1.2 | <i>A microservice development for document management system</i> | 30 |
| 5.1.3 | <i>A microservice-based architecture approach for the automation of modular process plants</i> | 30 |
| 5.1.4 | <i>A microservices architecture for collaborative document editing enhanced with face recognition</i> | 31 |
| 5.1.5 | <i>A reference architecture for real-time microservice API consumption</i> | 31 |
| 5.1.6 | <i>An application of microservices architecture pattern to create a modular computer numerical control system</i> | 32 |
| 5.1.7 | <i>An architecture for self-managing microservices</i> | 32 |
| 5.1.8 | <i>An open IoT framework based on microservices architecture</i> | 33 |
| 5.1.9 | <i>Asynchronous Queue Based Approach for Building Reactive Microservices</i> | 33 |
| 5.1.10 | <i>Case Study: Microservice Evolution and Software Lifecycle of the XSEDE User Portal API</i> | 34 |
| 5.1.11 | <i>Container-based microservice architecture for cloud applications</i> | 35 |
| 5.1.12 | <i>Cyber-physical microservices: An IoT-based framework for manufacturing systems</i> | 35 |
| 5.1.13 | <i>Decision guidance models for microservices: service discovery and fault tolerance</i> | 36 |

| | | |
|--------|--|----|
| 5.1.14 | <i>Design and Development of Backend Application for Public Complaint Systems Using Microservice Spring Boot</i> | 36 |
| 5.1.15 | <i>From Monolith to Microservices: A Dataflow-Driven Approach</i> | 37 |
| 5.1.16 | <i>Highly-Available Applications on Unreliable Infrastructure: Microservice Architectures in Practice</i> | 38 |
| 5.1.17 | <i>Microservice-based architecture for the NRDC</i> | 38 |
| 5.1.18 | <i>Microservices approach for the internet of things</i> | 39 |
| 5.1.19 | <i>Microservices architecture: Case on the migration of reservation-based parking system</i> | 39 |
| 5.1.20 | <i>Microservices model in WoO based IoT platform for depressive disorder assistance</i> | 40 |
| 5.1.21 | <i>Microservices-based software architecture and approaches</i> | 41 |
| 5.1.22 | <i>Migrating Monolithic Mobile Application to Microservice Architecture: An Experiment Report</i> | 41 |
| 5.1.23 | <i>Migrating web applications to clouds with microservice architectures</i> | 42 |
| 5.1.24 | <i>PipelineX: a feature animation pipeline on microservices</i> | 42 |
| 5.1.25 | <i>Publishing linked data through semantic microservices composition</i> | 43 |
| 5.1.26 | <i>Real-Time HazMat Environmental Information System: A micro-service based architecture</i> | 43 |
| 5.1.27 | <i>Scalable microservice based architecture for enabling DMTF profiles</i> | 44 |
| 5.1.28 | <i>Service fabric: a distributed platform for building microservices in the cloud</i> | 44 |
| 5.1.29 | <i>SmartVM: A Multi-Layer Microservice-Based Platform for Deploying SaaS</i> | 45 |
| 5.1.30 | <i>The ENTICE approach to decompose monolithic services into microservices</i> | 46 |
| 5.1.31 | <i>Towards a microservices development approach for the crisis management field in developing countries</i> | 47 |
| 5.1.32 | <i>Transform Monolith into Microservices using Docker</i> | 47 |
| 5.1.33 | <i>Using microservices in educational applications of IT-company</i> | 48 |

| | | |
|----------|--|-----------|
| 6 | Análise dos Resultados | 49 |
| 6.1 | Categorização dos Estudos | 49 |
| 6.1.1 | Primeiras Facetas | 49 |
| 6.1.2 | Últimas Facetas | 51 |
| 6.2 | Publicações por Ano | 56 |
| 6.3 | Distribuição entre as Bases de Dados | 57 |
| 7 | Considerações Finais | 60 |
| 7.1 | Resultados | 61 |
| 7.2 | Conclusões | 61 |
| 7.3 | Trabalhos Futuros | 62 |
| | Referências Bibliográficas | 63 |

Resumo

Este trabalho apresenta um mapeamento sistemático sobre abordagens para criação de microserviços. Nele são apresentados conceitos sobre a arquitetura de microserviços e um planejamento para execução do mapeamento sistemático. A execução do mapeamento é descrita e uma análise dos estudos selecionados é realizada, juntamente com a categorização. São gerados dados sobre a distribuição das categorias entre os trabalhos e é realizada uma análise das informações. Como resultado obteve-se uma listagem de estudos que respondem à pergunta de pesquisa e consolidou-se a relevância do estilo arquitetural de microserviços, além da visualização de tendências e informações sobre o estado da arte do assunto.

Palavras-chave: Mapeamento Sistemático; Microserviços; Arquitetura;

Capítulo 1

Introdução

A arquitetura baseada em microserviços defende a criação de um sistema a partir de uma coleção de serviços pequenos, isolados e independentes, cada um contendo seus próprios dados, sendo ainda escalável e resiliente às falhas. Os serviços integram-se entre si para formar um sistema coeso (BONÉR, 2016). É também definido por (THÖNES, 2015) como uma aplicação pequena que pode ser implantada de forma independente, escalonado de forma independente, testado de forma independente e tem somente uma função. Um microserviço normalmente é auxiliado por um *framework* de implementação e orquestração, por exemplo a nuvem, permitindo que eles sejam implantados de maneira frequente e independente, em agendas arbitrárias (PAHL; JAMSHIDI, 2016).

1.1 Motivação do Trabalho

Microserviço é um tema recente, segundo observado pelo estudo de Nuha Alshuqayran, Nour Ali e Roger Evans, o estilo arquitetural de microserviços surge na literatura a partir de 2014. Neste estudo também nota-se que, pelo estilo arquitetural ter surgido a partir de pesquisas acadêmicas, existem lacunas entre o que é aplicado no mercado e o que já foi estudado. Ou seja, nem todos os conceitos estudados são utilizados efetivamente (ALSHUQAYRAN; ALI; EVANS, 2016).

Embora o tema seja recente, houve um crescimento na quantidade de estudos relevantes ao assunto entre 2014 e 2015, demonstrando uma preocupação significativa da comunidade da computação (PAHL; JAMSHIDI, 2016). Dentre esses estudos, existem aqueles que tratam das abordagens para a construção de microserviços.

A tradicional aplicação monolítica possui problemas principalmente com relação à manutenibilidade, pequenas mudanças no sistema requerem que o sistema inteiro seja reconstruído e reimplantado, o que em algumas situações pode ser muito custoso. Com a arquitetura de microserviços pequenas mudanças podem ser feitas com facilidade, pois cada serviço pode ser implantado e escalado de maneira independente. Essa arquitetura permite ainda que diferentes serviços sejam escritos em diferentes linguagens, por diferentes equipes (LEWIS; FOWLER, 2014). Algumas empresas como Netflix e SoundCloud adotaram a arquitetura de microserviços (CALÇADO, 2014)(TONSE, 2014).

Isto posto, identifica-se a pergunta motivadora deste trabalho: Quais são as abordagens existentes para o desenvolvimento de aplicações com a arquitetura de microserviços? Abordagens podem ser caracterizadas como métodos, processos, modelos, técnicas, diretrizes, etc. Para responder a essa pergunta foi realizado um mapeamento sistemático que consiste em um procedimento sistemático e objetivo, para identificar a natureza e a extensão dos dados de um estudo, estes dados estão disponíveis para responder a uma determinada questão de pesquisa (BUDGEN et al., 2017).

Como resultado obteve-se uma listagem com estudos que respondem a pergunta de pesquisa e uma análise individual de cada um deles. A categorização desses estudos permitiu analisar tendências e trazem mais informações sobre o estado da arte das abordagens para criação de microserviços. Informações sobre melhores bases de dados para pesquisas sobre o assunto e sugestões de trabalhos futuros também são resultados deste trabalho.

O restante deste trabalho é dividido da seguinte forma: o Capítulo 2 apresenta o referencial teórico que serviu como conhecimento base para execução do trabalho; o Capítulo 3 expõe o planejamento do mapeamento sistemático, ou seja, as métricas utilizadas para realizar as pesquisas nas bases de dados, os filtros para seleção dos estudos, as categorias utilizadas para posterior análise e a execução do projeto piloto; o Capítulo 4 documenta a execução do mapeamento sistemático, nele é apresentado detalhadamente os passos realizados para obtenção da listagem final de estudos; no Capítulo 5 é realizada uma análise individual dos artigos selecionados. Nessa sessão também é feita, durante a análise, a categorização dos trabalhos; o Capítulo 6 sintetiza os dados obtidos após a análise do capítulo anterior, e documenta a análise dessas informações; no Capítulo 7 são feitas as considerações finais sobre os resultados e conclusões

obtidas, e também são sugeridos trabalhos futuros.

Capítulo 2

Referencial Teórico

Nesta seção será apresentada uma comparação entre sistemas monolíticos e arquitetura de microserviços. Na sequência serão expostas definições sobre a arquitetura de microserviços, e também sobre a metodologia do mapeamento sistemático. Ao final da seção será discutido sobre trabalhos relacionados.

Um sistema monolítico (“tudo em um”, “tudo ou nada”) é a forma que muitos sistemas foram projetados. Eles possuem todas as funções dentro de um grande “bloco” de sistema, ou seja, se uma única linha de código falhar o sistema todo irá falhar. Esses sistemas são difíceis de escalar, difíceis de manter e difíceis de entender. A arquitetura de microserviços elimina essas barreiras, pois trabalha de forma que cada função do sistema represente um sistema todo, onde as funções se comunicam via troca de mensagens (BONÉR, 2016).

Uma aplicação monolítica coloca todas as funcionalidades em um único processo, e escala a aplicação replicando o monolítico em diversos servidores. Na arquitetura de microserviços cada funcionalidade é colocada em um serviço separado, e para escalar a aplicação eles são distribuídos entre servidores, replicando conforme necessidade.

A Figura 2.1 representa a estrutura de uma aplicação monolítica. As funcionalidades do sistema são representadas pelas formas irregulares e coloridas, elas são envoltas por uma forma quadrada, que representa um processo. Os servidores são representados por uma forma cúbica, e na imagem observa-se que um processo é colocado como um todo em cada um dos servidores.

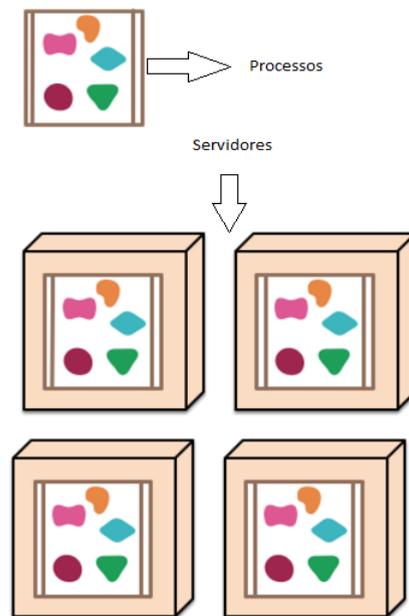


Figura 2.1: Aplicação Monolítica. Adaptado de (LEWIS; FOWLER, 2014)

Já a Figura 2.2 representa a arquitetura de microserviços, nela observa-se as formas quadradas, que representam os processos, envolvendo apenas uma forma irregular, ou seja, aquele processo possui uma única funcionalidade. Nas formas cúbicas, que representam os servidores, percebe-se uma distribuição não uniforme dos processos entre os diferentes servidores, isso ocorre porque na arquitetura de microserviços os processos são distribuídos conforme a demanda.

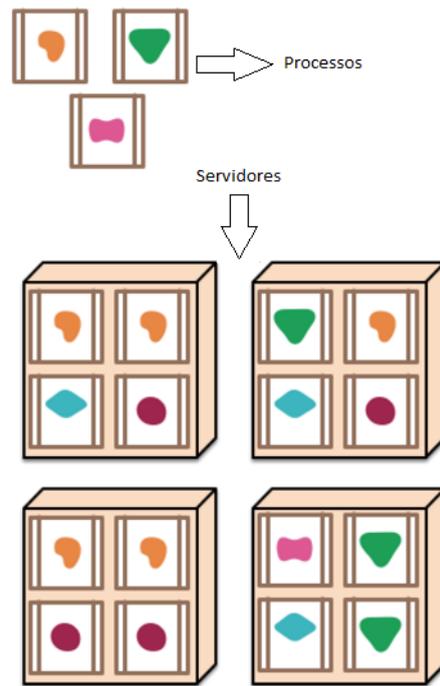


Figura 2.2: Arquitetura de Microserviços. Adaptado de (LEWIS; FOWLER, 2014)

2.1 Microserviço

Microserviço é definido como o menor processo possível, que seja independente e que interaja via troca de mensagens. Já a arquitetura de microserviço é definida como uma aplicação distribuída, onde todos os seus módulos são microserviços (DRAGONI et al., 2017).

A arquitetura de microserviços é um estilo arquitetural que dá ênfase em dividir o sistema em serviços pequenos e leves, os quais são construídos com o propósito de realizar tarefa muito específica. É uma evolução da tradicional Arquitetura Orientada a Serviços (SOA - *Service Oriented Architecture*) (LEWIS; FOWLER, 2014).

Essas características do microserviço fazem com que a divisão de trabalho em uma equipe de desenvolvimento torne-se mais natural, pois os sistemas podem ser estruturados da mesma forma que as equipes: dividindo responsabilidades entre os membros do time e dando liberdade para que façam o seu trabalho. Dessa forma as equipes podem evoluir de forma rápida, porque eles entendem o software dentro de limites bem definidos que eles controlam (BONÉR, 2016).

Em seu livro Bonér levanta um questionamento se o microserviço não é simplesmente um SOA “vestido em novas roupas”. Ele alega que a resposta é sim e não, simultaneamente. Sim, porque os objetivos iniciais são os mesmos (desacoplamento, isolamento, composição, integração e serviços autônomos). E não, porque as ideias fundamentais do SOA foram muitas vezes incompreendidas e mal utilizadas, gerando sistemas complicados que na verdade eram uma junção de vários monolíticos, que se comunicavam por protocolos muito complicados e ineficientes (BONÉR, 2016).

A arquitetura de microserviços manteve as boas ideias do SOA, porém reconstruiu todos os seus princípios baseados nos **princípios reativos**. Os quais são definidos pelo Manifesto Reativo, que diz que um sistema reativo deve ser (BONÉR et al., 2014):

- **Responsivo:** Sistemas responsivos focam em tempo de resposta rápido e consistente, estabelecendo limites superiores para que seja fornecido um serviço de qualidade, e também consistente. Esse comportamento facilita o tratamento de erros, cria confiança para o usuário e incentiva futuras interações.
- **Resiliente:** O sistema permanece responsivo diante de uma falha. Qualquer sistema que não seja resiliente não será responsivo após uma falha. A resiliência é conseguida com replicação, contenção, isolamento e delegação. As falhas estão contidas em cada componente, assim, ao isolar os componentes uns dos outros é garantido que partes do sistema possam falhar e se recuperar sem comprometer o sistema como um todo.
- **Elástico:** O sistema permanece reativo independente da carga de trabalho. Os sistemas reativos reagem à mudanças de fluxo de entrada aumentando ou diminuindo os recursos alocados para atender à demanda. Isso gera sistemas que não possuem gargalos, pois ele possui a habilidade de fragmentar ou replicar componentes para distribuir as entradas entre eles.
- **Baseado em mensagens:** Sistemas reativos dependem de troca de mensagens assíncronas para estabelecer uma barreira entre os componentes, o que garante o baixo acoplamento, o isolamento e a transparência da localização. Essa barreira fornece os meios para representar uma falha como uma mensagem.

Outra característica da arquitetura de microserviços é a necessidade de gerenciamento de dados descentralizados. Isso é comum em grandes empresas, que utilizam diferentes sistemas com diferentes propósitos. Dessa forma, atributos presentes em um sistema podem não estar presentes em outro, ou pior, presentes porém com semântica ligeiramente diferente. Isso torna os dados difíceis de gerenciar (LEWIS; FOWLER, 2014).

Esse problema é comum quando se lida com diferentes aplicações, mas também pode ocorrer internamente em aplicações, especialmente quando essa aplicação é dividida em componentes separados. Os microserviços descentralizam armazenamento de dados. Enquanto aplicações monolíticas preferem um único banco de dados lógico para persistir informações, as empresas geralmente utilizam um único banco de dados para várias aplicações. Os microserviços permitem que cada serviço gerencie seu próprio banco de dados, seja instâncias diferentes de um mesmo banco de dados ou sistemas de banco de dados totalmente diferentes.

A Figura 2.3 demonstra a forma como os dados são centralizados em uma aplicação monolítica. Nela é possível observar na parte inferior um único banco de dados fornecendo informações para uma aplicação, com seus diferentes serviços representados pelas formas irregulares visualizadas dentro da forma cúbica que representa o sistema. A Figura 2.4 expõe a descentralização do armazenamento de dados. Na figura são exibidos quatro microserviços, sendo os dois mais à direita representantes de um mesmo serviço. Percebe-se que cada serviço possui o seu próprio banco de dados, e que as duas instâncias do mesmo serviço utilizam a mesma base de informações.

O fato dos dados serem gerenciados de forma descentralizada tem implicações no gerenciamento de atualizações das informações. A maneira comumente utilizada para lidar com isso tem sido usar transações para garantir consistência ao atualizar vários recursos. Esta abordagem é usada frequentemente em monolíticos, porém, apesar do uso dessas transações ajudarem na consistência, acabam impondo um acoplamento temporal significativo, que se torna problemático quando há vários serviços. Para resolver isso, os microserviços buscam enfatizar a coordenação entre os serviços sem registrar transações, utilizando operações de compensação para lidar com as inconsistências (LEWIS; FOWLER, 2014).

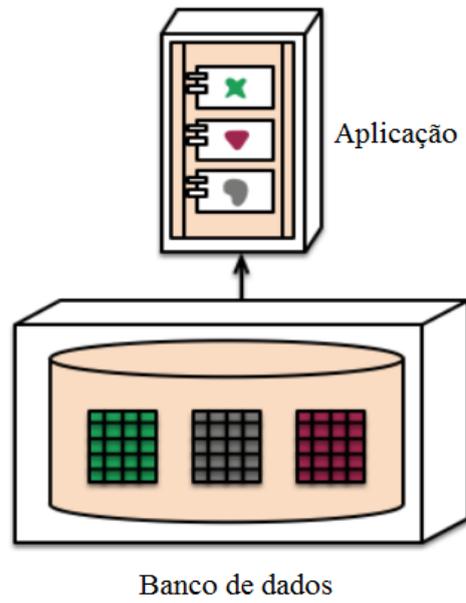


Figura 2.3: Gerenciamento de dados aplicação monolítica. Adaptado de (LEWIS; FOWLER, 2014)

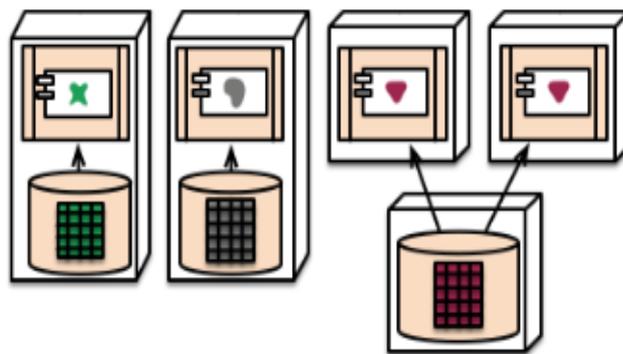


Figura 2.4: Gerenciamento de dados nos microserviços. Adaptado de (LEWIS; FOWLER, 2014)

2.2 Mapeamento Sistemático

Os Estudos de Mapeamento Sistemático reduzem o viés através de uma seqüência rigorosa de etapas metodológicas para a literatura de pesquisa. Eles dependem de protocolos de revisão bem definidos e avaliados para extrair, analisar e documentar os resultados (PAHL; JAMSHIDI, 2016).

O protocolo de revisão desse mapeamento é dividido em três etapas: **planejamento**, **execução** e **documentação**, cada etapa com suas subetapas específicas, baseado na adaptação do processo de mapeamento apresentado em (PETERSEN et al., 2008).

Dentro do **planejamento**:

- Definição do protocolo;
 - Identificação de palavras chave;
 - Definição dos critérios de seleção de artigos;
 - Definição das bases de dados para consulta;

Na **execução**:

- Pesquisa das palavras chave nas bases de dados;
- Leitura dos artigos selecionados;
- Identificação de abordagens utilizadas;
- Categorização;

E a **documentação** é apresentada no formato de Trabalho de Conclusão de Curso, sendo dividida nas seguintes etapas:

- Escrita do protocolo utilizado;
- Análise dos dados;
- Documentação dados:
 - Texto final;

Mais detalhes sobre essas etapas e subetapas serão apresentados nas próximas seções deste trabalho.

2.3 Trabalhos Relacionados

O artigo de Nuha Alshuqayran, Nour Ali e Roger Evans intitulado *A Systematic Mapping Study in Microservice Architecture* é um mapeamento sistemático sobre arquitetura de micro-serviços que buscou identificar desafios arquiteturais, diagramas/visões arquiteturais e atributos de qualidade relacionados aos sistemas de microserviços (ALSHUQAYRAN; ALI; EVANS, 2016).

No artigo os autores explicam que o rápido progresso da velocidade de rede, confiabilidade e segurança criam uma demanda crescente para que o software e os serviços não sejam armazenados e processados localmente, nas máquinas dos usuários, mas gerenciados por serviços de terceiros. Isso criou a necessidade de desenvolver novos métodos de desenvolvimento de software e estilos arquitetônicos que atendam a essas novas demandas.

Eles citam o recente surgimento da arquitetura de microserviços para atender às necessidades de manutenção e escalabilidade dos provedores online como exemplo da demanda pelos novos métodos de arquitetura de software. Como a arquitetura de microserviços é uma nova área de pesquisa, eles consideram crucial a necessidade de um estudo de mapeamento sistemático para resumir o progresso até o momento e identificar as lacunas e requisitos para estudos futuros.

Ainda no mesmo artigo, os autores apresentam o estudo sistemático de mapeamento de arquiteturas de microserviços e sua implementação, onde analisaram detalhadamente os trabalhos disponíveis sobre a arquitetura de microserviços e os desafios arquitetônicos relevantes. O estudo utiliza dois métodos de síntese qualitativa e quantitativa e aborda três principais perguntas de pesquisa. A primeira pergunta aborda os desafios arquitetônicos enfrentados pelos sistemas de microserviço, onde os pesquisadores puderam explorar todos os artigos e estudos publicados que destacaram as lacunas na pesquisa de microserviços, e fizeram sugestões sobre como abordar algumas das futuras soluções e iniciativas. A segunda questão de pesquisa investiga quais diagramas e visualizações de arquitetura, além de quais métodos ou modelos, são usados para representar arquiteturas de microserviços. Além disso, a última questão de pesquisa, afirma os

possíveis atributos de qualidade relacionados aos microserviços que são apresentados na literatura. Os autores sugerem como trabalho adicional a realização de uma revisão sistemática da literatura, que leva em conta outras considerações sobre a arquitetura de microserviços.

Outro trabalho relacionado é o artigo de Claus Pahl e Pooyan Jamshidi, intitulado *Microservices: A Systematic Mapping Study*, que também é um mapeamento sistemático sobre microserviços. Nele os autores buscaram identificar, classificar e comparar sistematicamente o corpo de pesquisa sobre a utilização de microserviços na nuvem.

No trabalho é apresentado que os microserviços emergiram recentemente como um estilo arquitetônico, abordando como construir, gerenciar e desenvolver arquiteturas a partir de unidades pequenas e independentes. Particularmente na nuvem, a abordagem da arquitetura de microserviços parece ser um complemento ideal da tecnologia de contêineres no nível de PaaS (*Platform as a Service*) (PAHL; JAMSHIDI, 2016).

Os autores realizaram o estudo de mapeamento sistemático em 21 estudos selecionados, publicados nos anos de 2014 e 2015, desde o surgimento do padrão de microserviços. Eles classificaram e compararam os estudos selecionados com base em um quadro de características. Isso resultou em uma discussão das preocupações emergentes dentro do estilo arquitetural de microserviços, posicionando a arquitetura de microserviços dentro de um contexto de desenvolvimento contínuo, mas também aproximando ela da tecnologia da nuvem e de contêineres.

O terceiro estudo apresentado como trabalho relacionado é do autor Gabriel Campeanu, intitulado *A mapping study on microservice architectures of Internet of Things and cloud computing solutions*. Nele o autor comenta que a Internet das Coisas é um paradigma relativamente novo adotado pela indústria, que oferece a conectividade, via sistemas sem fio, de todos os dispositivos que nos rodeiam. Um dos desafios da IoT está relacionado aos recursos necessários para armazenar e calcular a enorme quantidade de dados resultante das conexões dos dispositivos. A computação em nuvem é uma solução para os desafios da IoT, ele fornece recursos sob demanda de maneira fácil de acessar.

Ele também cita que outra tendência no mundo corporativo é o uso de arquiteturas de microserviço. Sendo um paradigma recém-desenvolvido, e embora seus princípios sejam definidos, é difícil ter uma visão das soluções de pesquisa existentes baseadas em microserviço. Este artigo, através da metodologia de estudo de mapeamento sistemático, fornece uma visão geral do atual

estado da arte em relação ao uso de arquiteturas de microserviço em soluções de computação em nuvem e IoT. Foram sintetizados dados de 364 estudos selecionados e descritos os tipos de pesquisa, número de publicações e seus principais locais (CAMPEANU, 2018).

Como pôde ser observado, os trabalhos citados não realizaram um estudo focado nas abordagens para criação de microserviços, por conta disso acredita-se que este trabalho pode contribuir nessa área. Outro fator importante é o crescimento do número de pesquisas relacionadas a microserviços no período após a publicação dos estudos comentados e a realização deste trabalho. Estes novos estudos podem conter contribuições relevantes, que pretendemos indentificar e categorizar, dentro do escopo da pergunta de pesquisa.

Capítulo 3

Planejamento do Mapeamento Sistemático

Esta seção visou documentar a forma que a pesquisa foi executada para fins de replicação do estudo. O planejamento serviu de guia para a execução do trabalho, e deve ser bem definido antes do início das pesquisas.

Para validar se os itens apresentados eram realmente as melhores opções para responder a pergunta de pesquisa, é realizada a execução do protocolo de maneira reduzida e observado se os resultados da pesquisa trazem estudos relevantes. Neste momento, chamado de piloto, são validadas as palavras-chave utilizadas, as bases de dados onde serão realizadas as buscas e os critérios de seleção dos estudos. É um momento de teste, onde ajustes são feitos buscando obter melhores resultados. Detalhes sobre a execução do piloto que validou os itens apresentados a seguir podem ser vistos na Seção 3.7.

3.1 Pergunta de Pesquisa

Conforme citado na Seção 1.1, devido ao crescente número de estudos publicados pela comunidade científica sobre o tema de microserviços, percebe-se um interesse sobre a área. Isso é apresentado na Figura 3.1, que possui dados extraídos da base de dados ACM, e demonstra um aumento significativo de publicações nessa plataforma a cada ano. No ano de 2015 foram publicados 7 estudos sobre o assunto, em 2016 o gráfico registra 40 estudos e no ano de 2017 houve 62 publicações. No ano de 2018, até o mês de agosto, há 32 artigos publicados.

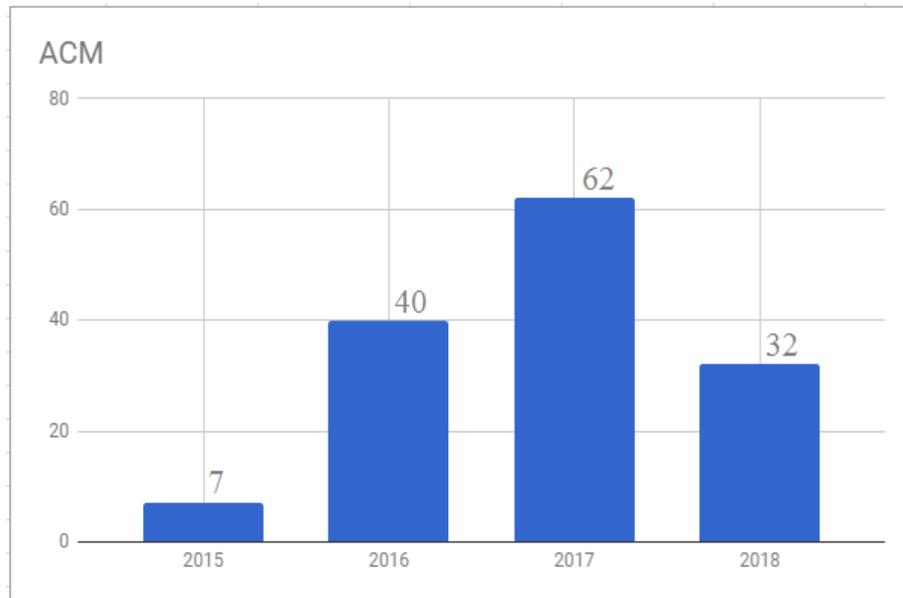


Figura 3.1: Crescimento de publicações sobre microserviços na ACM.

A arquitetura de microserviço demonstra ter potencial para solução de problemas de manutenibilidade existentes no mercado (THÖNES, 2015). Dessa forma, categorizar técnicas utilizadas para projetar microserviços pode ser uma contribuição para a comunidade.

A pergunta de pesquisa é: “Quais são as abordagens existentes para o desenvolvimento de aplicações com a arquitetura de microserviços?”. O termo “abordagens” pode ser caracterizado como métodos, processos, modelos, técnicas, diretrizes, *frameworks*, etc.

3.2 Palavras-chave

A pergunta de pesquisa apresentada na Seção 3.1 é o guia de todo o trabalho. Portanto, as palavras chaves foram extraídas dela e são: microserviço, arquitetura, abordagem, método, processo, modelo, técnica, diretriz, *framework*.

As bases de dados (Seção 3.3) onde foram realizadas as pesquisas utilizam inglês como padrão de linguagem. Portanto, para a composição da string de busca (Seção 3.4) que foi utilizada foi necessário que as palavras-chave também estivessem escritas em inglês. Por esse motivo foi realizada a tradução, conforme a Tabela 3.1. Essas palavras traduzidas são as que foram efetivamente utilizadas na composição das strings de busca.

| Original | Tradução |
|--------------|---------------------|
| microserviço | <i>microservice</i> |
| arquitetura | <i>architecture</i> |
| abordagem | <i>approach</i> |
| método | <i>method</i> |
| processo | <i>process</i> |
| modelo | <i>model</i> |
| técnica | <i>technique</i> |
| diretriz | <i>guideline</i> |
| framework | <i>framework</i> |

Tabela 3.1: Tradução das palavras-chave.

Observa-se que no mapeamento sistemático de (PAHL; JAMSHIDI, 2016), que está apontado como trabalho relacionado, foi utilizada a palavra-chave “*microservice*” e uma variação “*micro-service*”. Decidiu-se expandir a pesquisa utilizando os termos propostos, buscando abranger uma maior quantidade de estudos que possam ser do interesse para responder a pergunta de pesquisa.

Foram realizadas pesquisas utilizando outras palavras-chave, como por exemplo “SOA” e “*service oriented architecture*”, porém a grande quantidade de resultados retornados inviabilizavam a execução deste trabalho. Detalhe este que será aprofundado na seção Piloto.

3.3 Bases de Dados

Para pesquisa dos artigos foram usadas bases de dados digitais, que são listadas a seguir:

- ACM (<http://dl.acm.org/>)
- IEEEExplore (<http://ieeexplore.ieee.org/>)
- ScienceDirect (<http://sciencedirect.com/>)

Essas bases de dados são uma adaptação das bases utilizadas pelo trabalho relacionado apresentado por (ALSHUQAYRAN; ALI; EVANS, 2016), que realizaram as pesquisas nas bases ACM, IEEEExplore e Scopus. No caso deste trabalho a Scopus foi substituída pela ScienceDirect, pois traz apenas resumos e citações da literatura, enquanto sua substituta selecionada

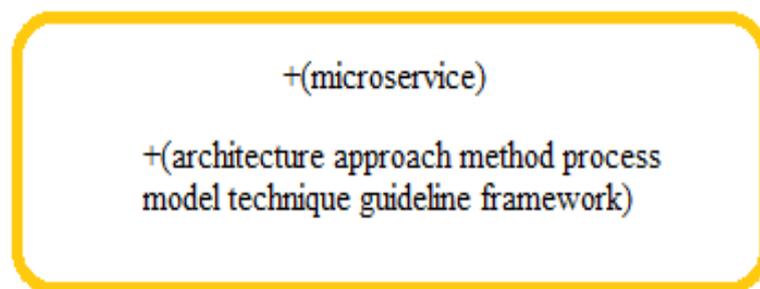
retorna publicações confiáveis de textos completos. As duas bases são plataformas de pesquisa da Elsevier¹.

Outra razão que justifica a seleção destas bases de dados é a disponibilidade de acesso ao conteúdo dos artigos científicos presentes, que é fornecida pela Universidade Estadual do Oeste do Paraná - Unioeste.

3.4 Strings de Busca

As strings de busca devem ser organizadas de acordo com a sintaxe de cada base de dados. As bases IEEEExplore e ScienceDirect utilizam uma composição das palavras chaves com operadores lógicos AND ou OR. Já a ACM possui uma sintaxe mais específica, conforme exemplo apresentado pela Universidade de Akron². Nela é necessário utilizar um símbolo “+” antes de conjuntos de palavras, e o espaço entre cada palavra representa um operador lógico OR.

A string composta para a busca na base de dados ACM é apresentada na Figura 3.2. Para busca nas bases IEEEExplore e na ScienceDirect utilizou-se a mesma string, por possuírem a mesma sintaxe, e ela pode ser vista na Figura 3.3. Essas strings buscam encontrar artigos que contenham necessariamente a palavra-chave *microservice*, que se encontra dentro de um primeiro conjunto de parênteses, juntamente com uma ou mais palavras-chave presentes no segundo conjunto de parênteses.



+(microservice)
+(architecture approach method process
model technique guideline framework)

Figura 3.2: String de busca utilizada na ACM

¹Plataformas de pesquisa: <https://www.elsevier.com/pt-br/research-platforms>

²Exemplo de busca na ACM: <https://www.uakron.edu/dotAsset/659aac25-dcf2-478a-ab86-d5074f35c4dc.pdf>

```
(microservice)
AND
(architecture OR approach OR method OR process
OR model OR technique OR guideline OR framework)
```

Figura 3.3: String de busca utilizada na IEEEExplore e ScienceDirect

3.5 Critérios de Inclusão e Exclusão

Os critérios de inclusão e exclusão dos artigos neste trabalho visam obter os melhores resultados baseados na pergunta de pesquisa. Baseiam-se também nos mapeamentos sobre micro-serviços de (PAHL; JAMSHIDI, 2016) e (ALSHUQAYRAN; ALI; EVANS, 2016).

Em uma busca nas bases de dados digitais pôde-se obter uma quantidade significativa de resultados (dezenas, centenas, ou até milhares de artigos). Se não existirem formas de reduzir a quantidade de estudos a serem analisados o mapeamento pode se tornar inviável. Os critérios de inclusão e exclusão, apresentados a seguir, servem para filtrar trabalhos que podem não ser relevantes para responder a pergunta de pesquisa, e que portanto, não devem ser considerados em etapas posteriores.

Em um primeiro momento, os critérios foram aplicados observando-se somente o título e as palavras-chave do artigo, quando enquadrado em pelo menos um critério de exclusão, o estudo foi excluído das próximas análises. Os estudos que não foram excluídos tiveram os mesmos critérios aplicados, mas agora somente no *abstract* do artigo.

3.5.1 Inclusão

- Estudos a respeito de abordagens para criação de microserviços;

Esse primeiro critério de inclusão busca selecionar artigos que falem não somente sobre microserviços, mas sobre abordagens para sua criação, visto que artigos selecionados devem contribuir para responder a pergunta de pesquisa que é o foco deste trabalho.

- Estudos que incluem os termos de busca no título, no *abstract* ou nas palavras chave;

O segundo critério de inclusão parte do pressuposto que artigos que não apresentem os termos de busca nas seções citadas provavelmente não serão relevantes para responder a pergunta de pesquisa.

- Estudos publicados até o mês de Junho de 2018;

O terceiro critério busca traçar um limite de prazo para a inclusão dos artigos. Novos estudos continuam sendo publicados, mas o resultado desse trabalho é referente ao momento em que foi executado o mapeamento. Caso o estudo seja replicado em outro período, só será possível obter os mesmos resultados considerando os mesmos estudos que existiam até aquele momento.

- Estudos escritos em inglês;

O último critério de inclusão busca selecionar apenas artigos escritos na língua inglesa. As bases de dados digitais podem retornar estudos escritos em outras línguas, o que faria necessário possuir conhecimento sobre outros idiomas, ou a realização de tradução dos trabalhos. Devido ao tempo disponível para execução do mapeamento optou-se por focar apenas em um idioma.

3.5.2 Exclusão

- Materiais em forma de resumo;

O primeiro critério de exclusão acarretará na seleção apenas de artigos científicos completos, com respaldo da comunidade científica.

- Estudos com os termos de busca usados com diferente significado;

O segundo critério de exclusão pretende eliminar do mapeamento artigos que contenham os termos de busca de interesse, porém com outro sentido, visto que dessa forma não tem potencial para auxiliar a responder a pergunta de pesquisa.

O procedimento de seleção dos estudos é apresentado na Seção 4.

3.6 Método de Classificação

Após a execução do protocolo de mapeamento e da aplicação dos critérios de inclusão e exclusão, os artigos selecionados foram classificados em categorias. Os estudos podem se enquadrar em zero, uma, ou mais de uma categoria.

3.6.1 Facetas e Categorias

Uma faceta deste trabalho, que representa um conjunto de categorias, é “**Tipo de Pesquisa**”. Dentro dessa faceta existirão categorias onde os estudos serão qualificados à elas após análise.

A seguir, apresentamos as categorias desta faceta com a sua descrição associada (WIERINGA et al., 2006):

- **Artigo de experiência:** Artigos dessa natureza explicam como algo foi feito na prática, e devem conter uma experiência pessoal do autor.
- **Artigo filosófico:** Esses artigos esboçam uma nova forma de olhar para algo que já existe, uma nova estrutura conceitual, etc.
- **Artigo de opinião:** Expressam a opinião pessoal do autor sobre determinada técnica, dizendo se ela é boa ou ruim, ou como deveria ser feita.
- **Pesquisa de avaliação:** Técnicas são implementadas na prática, e uma avaliação da técnica é feita.
- **Pesquisa de validação:** As técnicas apresentadas no artigo não foram implementadas na prática, por exemplo, através de um experimento.
- **Proposta de solução:** Uma solução para um problema é proposta, ela pode ser inédita ou uma extensão de uma técnica existente. Os benefícios e aplicações da solução podem ser apresentadas com um exemplo ou com uma boa argumentação.

Outra faceta é “**Tipo de Contribuição**”, a qual pode significar o tipo de abordagem, que por conseguinte, ajudaria a responder a questão de pesquisa. Nela identificamos as seguintes categorias: Processo, Método, Modelo, *Framework*, Abordagem, Técnica e Diretriz.

“**Tipo de desenvolvimento**” é uma faceta deste trabalho. Nela observamos duas categorias, a primeira é referente à migração da arquitetura monolítica de um sistema existente para arquitetura de microserviços. A segunda categoria é referente ao desenvolvimento de uma nova aplicação baseada na arquitetura de microserviços. As categorias serão nomeadas de Migração e Novo sistema, respectivamente.

“**Motivação para adoção de microserviços**” é a quarta faceta apresentada. As categorias presentes são: Escalabilidade, Evolução, Manutenção, Desempenho, Confiabilidade e Princípios reativos. O objetivo é entender qual a justificativa dos trabalhos para utilização da arquitetura de microserviços.

Outra faceta é “**Fases do processo de software**”. Nela será observado em qual fase do processo de desenvolvimento do software o estudo analisado se enquadra. As categorias são: Requisitos, Arquitetura, Implementação, Testes, Implantação.

3.7 Piloto

Essa seção representa a execução do protocolo de mapeamento proposto neste trabalho. Porém, com amostragem pequena, buscando apenas validar o planejamento deste mapeamento.

A partir da execução desse piloto foi possível calibrar e validar as palavras-chave propostas nesse planejamento. A princípio “SOA” e “*Service Oriented Architecture*” foram consideradas, mas ao inserí-las no conjunto, a quantidade de estudos retornados se tornava muito grande, inviabilizando o mapeamento (Tabela 3.2). Portanto, essas palavras-chave foram removidas do conjunto para possibilitar a realização do trabalho.

Os dois conjuntos de palavras-chave que foram testados durante a execução do piloto são: Conjunto 1 (microservice, architecture, approach, method, process, model, technique, guideline, framework) e Conjunto 2 (soa, “service oriented architecture”, microservice, architecture, approach, method, process, model, technique, guideline, framework). A Tabela 3.2 representa os valores obtidos utilizando as duas variações dos conjuntos de palavras-chave nas bases de dados.

| Base de dados | Resultados: Conjunto 1 | Resultados: Conjunto 2 |
|---------------|------------------------|------------------------|
| ACM | 129 | 1,381 |
| IEEEExplore | 325 | 20,429 |
| ScienceDirect | 190 | 32,418 |

Tabela 3.2: Quantidade de estudos obtidos na execução do piloto

A sintaxe de busca das palavras-chave nas bases de dados mudam conforme especificado por elas. A sintaxe utilizada para cada base foi apresentada anteriormente neste capítulo. Essa sintaxe representa o Conjunto 1 de palavras-chave. Dentre as variações possíveis de sintaxe de

busca, foram utilizadas as que retornavam mais estudos, buscando assim a maior quantidade de referências possíveis.

A base de dados Google Scholar³ foi considerada inicialmente, mas durante a execução do piloto sua utilização se mostrou inviável, pois com o Conjunto 1 de palavras-chave foram obtidos 7,720 (sete mil setecentos e vinte) resultados, e com o Conjunto 2 retornaram 2,110,000 (dois milhões cento e dez mil) resultados.

Dentre os resultados obtidos na coluna do Conjunto 1 na Tabela 3.2, realizamos uma amostragem com os 20 primeiros artigos retornados na pesquisa em cada base de dados, aplicando os critérios de inclusão e exclusão nos títulos para identificarmos quantos desses descrevem sobre abordagens para criação de microserviços, e obtivemos o seguinte resultado: ACM: 5 artigos; IEEEExplore: 4 artigos; ScienceDirect: 4 artigos. Artigos repetidos entre as bases de dados não foram considerados na execução do piloto, portanto, é possível que nos números apresentados existam artigos iguais entre as bases.

³<https://scholar.google.com.br/>

Capítulo 4

Execução do Mapeamento Sistemático

Este capítulo documenta os passos e os dados obtidos a partir da execução do planejamento apresentado no capítulo anterior. Nele é visto detalhadamente os valores encontrados em cada bases de dados, e as alterações percebidas após aplicação dos critérios de inclusão e exclusão. A forma utilizada para organizar e analisar os artigos também é apresentada ao longo do texto.

4.1 Resultados das Buscas

Os resultados apresentados a seguir são referentes à quantidade de artigos encontrados em cada base de dados durante a busca, utilizando as strings de busca apresentadas no capítulo anterior. Em caso de replicação do estudo, esses valores apresentados podem ser diferentes, visto que novos estudos são adicionados às bases de dados digitais diariamente. Porém, a quantidade de estudos encontrados após aplicação dos critérios de inclusão e exclusão apresentados nesse trabalho devem ser iguais, salvo a exclusão de algum estudo de alguma das bases de dados.

Após realização da pesquisa nas bases de dados com as strings de busca apresentadas, em um primeiro momento elas retornaram 722 artigos no total, ainda não considerando artigos repetidos nas diferentes bases de dados.

4.1.1 IEEEExplore

No momento da consulta a essa base de dados retornou 366 artigos. Para organizar e analisar os trabalhos foi utilizado um recurso da própria base de dados, onde é possível exportar o resultado da busca em alguns diferentes formatos. Para este trabalho foi utilizada a exportação

em formato “.csv¹”.

Após a execução dos procedimentos apresentados obteve-se uma lista dos títulos dos artigos retornados pela IEEEExplore. A string de busca utilizada foi “*(microservice) AND (architecture OR approach OR method OR process OR model OR technique OR guideline OR framework)*”.

4.1.2 ACM Digital Library

A busca nessa base de dados retornou 145 artigos. Para organizar e analisar os artigos não foi possível utilizar do mesmo método aplicado na IEEEExplore. A base de dados também possui um recurso para exportação dos resultados da busca, e também é encontrado o formato “.csv”.

Após executar os procedimentos acima obteve-se uma listagem dos títulos dos artigos retornados pelas duas bases de dados, ACM Digital Library e IEEEExplore. A string de busca utilizada foi “*+(microservice) +(architecture approach method process model technique guideline framework)*”.

4.1.3 ScienceDirect

A pesquisa nessa base de dados retornou 211 artigos. Para organização e posterior análise dos trabalhos foi necessário realizar a exportação de maneira manual, pois essa base de dados não fornece recursos para exportação dos resultados como as outras utilizadas nesse trabalho. Portanto foi feita a cópia dos títulos dos artigos e foi colado nas células da planilha de controle.

Após realização desses procedimentos constitui-se uma listagem dos títulos a serem analisados nesse trabalho. A string de busca utilizada foi “*(microservice) AND (architecture OR approach OR method OR process OR model OR technique OR guideline OR framework)*”.

4.2 Triagem dos Artigos

Nessa seção serão apresentados os resultados obtidos após a aplicação dos critérios de inclusão e exclusão apresentados no capítulo anterior. Também será apresentada a forma como foi executada a obtenção dos resultados apresentados. Por razões práticas, os critérios não foram aplicados necessariamente na ordem apresentada no planejamento. Os critérios foram aplicados na listagem final obtida que continha 722 artigos, e são descritos na Seção 4.2.1.

¹Comma-separated values

4.2.1 Aplicação dos Critérios de Inclusão e de Exclusão

Inicialmente foi considerado o terceiro critério de inclusão (Estudos publicados até o mês de Junho de 2018). Para realizar a aplicação desse critério foi utilizada uma funcionalidade das bases de dados que ordena a listagem de resultados da busca. Existem algumas opções para a ordenação dos resultados, porém, para o nosso objetivo, foi utilizada a função que ordena os resultados por data, dos mais novos para os mais antigos. Isso foi feito antes da exportação dos artigos explicada no início desse capítulo, portanto foram selecionados para exportação apenas os artigos que foram incluídos por esse critério. Ou seja, artigos publicados nas plataformas de busca até 30 de Junho de 2018. Detalhes sobre as datas dos artigos nas bases de dados serão apresentados individualmente na sequência.

Após aplicação do terceiro critério de inclusão nos 722 estudos restaram 586 trabalhos a serem considerados. Foram encontrados 20 artigos repetidos, que também foram eliminados, assim, restando 566. Para eliminação dos artigos repetidos entre as bases de dados foi utilizada a listagem que continha todos os artigos a serem analisados após aplicação do terceiro critério de inclusão.

As bases de dados retornaram apenas estudos escritos em inglês, portanto o quarto critério de inclusão (Estudos escritos em inglês) não reduziu a quantidade de artigos. Nenhum artigo foi eliminado pelo primeiro critério de exclusão (Materiais em forma de resumo).

A seguir são apresentados detalhadamente os resultados obtidos após aplicação do terceiro critério de inclusão em cada base de dados.

IEEEExplore

Na base de dados IEEEExplore é informada a data da publicação científica e a data em que foi disponibilizada para consulta. Aplicamos o terceiro critério de inclusão baseado na data em que a publicação ficou disponível para consulta, pois em datas anteriores numa pesquisa na base de dados ainda não seria encontrado esse artigo.

No momento da pesquisa foram 366 artigos listados, após a aplicação do terceiro critério de inclusão restaram 297 arquivos da base de dados IEEEExplore a serem analisados.

ACM Digital Library

A base de dados ScienceDirect informa datas de recebimento, revisão, aceite e a data em que o artigo foi disponibilizado online. Para aplicação do terceiro critério de inclusão foi utilizada a data em que foi disponibilizado online, pois antes disso não seria possível encontrar este artigo seguindo nosso planejamento.

A base de dados retornou 211 artigos, e após aplicação do terceiro critério de inclusão, restaram 167.

ScienceDirect

Assim como nas demais bases, a data utilizada para análise do critério de inclusão foi a data da disponibilização do artigo online nessa plataforma. Após aplicação do terceiro critério de inclusão, dos 145 artigos listados inicialmente restaram 122 para análise.

Resultado parcial

Nos 566 artigos restantes foi aplicado o segundo critério de exclusão (Estudos com os termos de busca usados com diferente significado). Neste momento foi possível observar diversos artigos que apenas citam a palavra “*microservice*”, sem necessariamente falar de maneira mais aprofundada sobre ele. Em alguns casos, a palavra aparece apenas uma vez ao longo do trabalho. Após aplicação do segundo critério de exclusão restaram 201 artigos para análise, e foi notado que o segundo critério de inclusão (Estudos que incluem os termos de busca no título, no abstract ou nas palavras-chave) foi cumprido, pois todos os artigos restantes continham pelo menos um termo de busca nas áreas observadas do artigo. Assim, restando apenas a aplicação do primeiro critério de inclusão.

Aplicação do primeiro critério de inclusão

Para aplicar o primeiro critério de inclusão (Estudos a respeito de abordagens para criação de microserviços) foi feita uma observação nos abstracts dos 201 artigos restantes, pois era necessário saber do que realmente falam a respeito àqueles estudos. Apenas estudos sobre criação, ou obtenção, de microserviços e arquiteturas baseadas em microserviços seriam incluídos na posterior análise.

Ao final da aplicação deste critério de inclusão chegamos à lista final, que representa todos os artigos científicos que foram analisados e categorizados dentro do planejamento apresentado anteriormente. São 33 artigos que conferem com todos os critérios apresentados, e seus títulos são apresentados em ordem alfabética na segunda coluna da Tabela 4.1. Ela visa facilitar o acesso ao leitor interessado em obter apenas a listagem de artigos, eliminando a necessidade de folhar por todas as páginas de análise do Capítulo 5. A primeira coluna dessa tabela representa a seção onde são encontradas as análises de cada estudo.

| Seção | Título do artigo |
|--------|---|
| 5.1.1 | <i>A microservice based reference architecture model in the context of enterprise architecture</i> |
| 5.1.2 | <i>A microservice development for document management system</i> |
| 5.1.3 | <i>A microservice-based architecture approach for the automation of modular process plants</i> |
| 5.1.4 | <i>A microservices architecture for collaborative document editing enhanced with face recognition</i> |
| 5.1.5 | <i>A reference architecture for real-time microservice API consumption</i> |
| 5.1.6 | <i>An application of microservices architecture pattern to create a modular computer numerical control system</i> |
| 5.1.7 | <i>An architecture for self-managing microservices</i> |
| 5.1.8 | <i>An open IoT framework based on microservices architecture</i> |
| 5.1.9 | <i>Asynchronous Queue Based Approach for Building Reactive Microservices</i> |
| 5.1.10 | <i>Case Study: Microservice Evolution and Software Lifecycle of the XSEDE User Portal API</i> |
| 5.1.11 | <i>Container-based microservice architecture for cloud applications</i> |
| 5.1.12 | <i>Cyber-physical microservices: An IoT-based framework for manufacturing systems</i> |
| 5.1.13 | <i>Decision guidance models for microservices: service discovery and fault tolerance</i> |
| 5.1.14 | <i>Design and Development of Backend Application for Public Complaint Systems Using Microservice Spring Boot</i> |
| 5.1.15 | <i>From Monolith to Microservices: A Dataflow-Driven Approach</i> |
| 5.1.16 | <i>Highly-Available Applications on Unreliable Infrastructure: Microservice Architectures in Practice</i> |
| 5.1.17 | <i>Microservice-based architecture for the NRDC</i> |
| 5.1.18 | <i>Microservices approach for the internet of things</i> |
| 5.1.19 | <i>Microservices architecture: Case on the migration of reservation-based parking system</i> |
| 5.1.20 | <i>Microservices model in WoO based IoT platform for depressive disorder assistance</i> |
| 5.1.21 | <i>Microservices-based software architecture and approaches</i> |
| 5.1.22 | <i>Migrating Monolithic Mobile Application to Microservice Architecture: An Experiment Report</i> |
| 5.1.23 | <i>Migrating web applications to clouds with microservice architectures</i> |
| 5.1.24 | <i>PipelineX: a feature animation pipeline on microservices</i> |
| 5.1.25 | <i>Publishing linked data through semantic microservices composition</i> |
| 5.1.26 | <i>Real-Time HazMat Environmental Information System: A micro-service based architecture</i> |

| | |
|--------|---|
| 5.1.27 | <i>Scalable microservice based architecture for enabling DMTF profiles</i> |
| 5.1.28 | <i>Service fabric: a distributed platform for building microservices in the cloud</i> |
| 5.1.29 | <i>SmartVM: A Multi-Layer Microservice-Based Platform for Deploying SaaS</i> |
| 5.1.30 | <i>The ENTICE approach to decompose monolithic services into microservices</i> |
| 5.1.31 | <i>Towards a microservices development approach for the crisis management field in developing countries</i> |
| 5.1.32 | <i>Transform Monolith into Microservices using Docker</i> |
| 5.1.33 | <i>Using microservices in educational applications of IT-company</i> |

Tabela 4.1: Listagem final de artigos

Capítulo 5

Análise dos Artigos

Neste Capítulo serão apresentadas detalhadamente informações dos estudos selecionados após execução do planejamento e da filtragem, a partir da aplicação dos critérios de inclusão e exclusão. A listagem final dos artigos é apresentada no Capítulo 4, e as próximas seções são organizadas sequencialmente, na mesma ordem apresentada na Tabela 4.1.

5.1 Extração de Dados

As informações contidas nos artigos selecionados que são relevantes para essa pesquisa são expostas a seguir. Cada artigo é analisado individualmente, e posteriormente será feita uma análise quantitativa das categorias observadas nos estudos e apresentados gráficos da distribuição delas entre os estudos.

5.1.1 *A microservice based reference architecture model in the context of enterprise architecture*

Neste artigo é apresentado um modelo de arquitetura baseada em microserviço com objetivo de servir de referência para construção, implementação e gerenciamento de microserviços corporativos, no contexto da arquitetura corporativa. O modelo possui definições claras do conceito de microserviço e de seus principais componentes arquitetônicos (YU; SILVEIRA; SUNDARAM, 2016).

Obtido na base de dados IEEEExplore, o trabalho de Yale Yu, Haydn Silveira e Max Sundaram foi publicado no *IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)* em 2016. E nele também são abordados pro-

blemas sobre questões arquiteturais importantes, e fornecidas recomendações correspondentes para melhor aproveitamento dos microserviços dentro de uma empresa de nível corporativo.

A partir da análise do estudo é possível atribuir as seguintes categorias a ele: Artigo de experiência, Pesquisa de avaliação, Proposta de solução, Processo, Modelo, Novo sistema, Escalabilidade, Manutenção, Princípios reativos e Arquitetura.

5.1.2 *A microservice development for document management system*

Este artigo propõe um método de desenvolvimento para um sistema eletrônico de governo, usando a arquitetura de microserviço, para solucionar problemas enfrentados pelos autores, que são escalabilidade e evolução do sistema. Para demonstração da abordagem foi implementado parte de um importante sistema do governo Coreano (SRIKAEW; KIM, 2017).

Obtido na base de dados IEEEExplore, o trabalho de Pankamol Srikaew e Inkyu Kim foi publicado na *4th International Conference on Computer Applications and Information Processing Technology (CAIPT)* em 2017.

Nele é apresentado um ciclo de desenvolvimento do sistema de gerenciamento de documentos, que se adapta do desenvolvimento ágil e DevOps, com as regras do microserviço. Cada processo de desenvolvimento pode ser revertido para o processo anterior, conforme necessário.

Neste artigo foram identificadas após análise as categorias: Artigo de experiência, Pesquisa de avaliação, Proposta de solução, Método, Abordagem, Migração, Escalabilidade, Evolução, Requisitos, Implementação, Testes e Implantação.

5.1.3 *A microservice-based architecture approach for the automation of modular process plants*

O trabalho que tem como autores Henry Bloch, Alexander Fay, Torsten Knohl, Mario Hornicke, Jens Bernshausen, Stephan Hensel, Anna Hahn e Leon Urbas foi publicado em 2017 na *22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, foi obtido pela base de dados IEEEExplore.

Nele é apresentada uma abordagem baseada em microserviços, e analisado se a arquitetura de microserviços é a mais adequada para automação modular de processos. Também é apresentada brevemente as diferenças entre SOA e microserviços (BLOCH et al., 2017).

Foi possível enquadrar este estudo nas categorias: Artigo de experiência, Pesquisa de avaliação, Proposta de solução, Abordagem, Novo sistema, Confiabilidade e Arquitetura.

5.1.4 *A microservices architecture for collaborative document editing enhanced with face recognition*

Obtido na base de dados IEEEExplore, o trabalho de Cristian Gadea, Mircea Trifan, Dan Ionescu, Marius Cordea e Bogdan Ionescu foi publicado no IEEE *11th International Symposium on Applied Computational Intelligence and Informatics (SACI)* em 2016.

Este artigo apresenta a arquitetura e a implementação de um editor de *Rich Text Format* (RTF) colaborativo que utiliza microserviços para permitir e aprimorar sua funcionalidade de coedição de maneira escalável. Microserviços são usados para sincronizar texto não estruturado, a partir de transformações operacionais, para a funcionalidade de bate-papo e para detectar e reconhecer rostos em imagens adicionadas ao editor (GADEA et al., 2016a).

O sistema é demonstrado apresentando como os microserviços possibilitam que vários usuários editem simultaneamente um documento, onde imagens contendo faces são adicionadas e reconhecidas como parte do conteúdo, assim dando suporte ao processo de criação de documentos.

Após análise do trabalho foram identificadas as categorias: Artigo de experiência, Proposta de solução, Abordagem, Novo sistema, Escalabilidade, Evolução, Manutenção, Princípios reativos, Arquitetura e Implementação.

5.1.5 *A reference architecture for real-time microservice API consumption*

Neste trabalho é apresentada uma arquitetura de referência construída sobre a ideia de um banco de dados escalável NoSQL, que permite que vários assinantes recebam notificações instantâneas de mudanças no banco de dados através do uso de uma “*livequery*”.

Ao manter uma conexão *WebSocket* aberta entre cada navegador da *Web* cliente e um *Object Synchronization Server*, este documento mostra como os dados de várias APIs REST diferentes podem ser organizados e transmitidos aos clientes interessados por meio do banco de dados (GADEA et al., 2016b).

Obtido na base de dados ACM, o trabalho de Cristian Gadea, Mircea Trifan, Dan Ionescu e Bogdan Ionescu foi publicado no *CrossCloud '16: 3rd Workshop on CrossCloud Infrastructures*

& Platforms em 2016.

As categorias percebidas neste trabalho são: Artigo de experiência, Método, Modelo, Técnica, Novo sistema, Escalabilidade, Princípios reativos e Arquitetura.

5.1.6 *An application of microservices architecture pattern to create a modular computer numerical control system*

Este trabalho sugere uma nova abordagem para o desenvolvimento de equipamentos industriais modulares usando um padrão de arquitetura de microserviços. Recursos de arquitetura de microserviços são abordados, bem como suas vantagens e desvantagens.

Ele tem como autores Maxim Ya. Afanasev, Yuri V. Fedosov, Anastasiya A. Krylova e Sergey A. Shorokhov e foi publicado em 2017 na *20th Conference of Open Innovations Association (FRUCT)*, foi obtido pela base de dados IEEExplore.

Uma rede de computadores heterogêneos, onde os nós se comunicam por meio de uma fila de mensagens, é proposta como base para o sistema de controle. A tolerância a falhas é fornecida por módulos com autonomia total e mensagens confiáveis. Nele é descrito a estrutura de configuração comum, além dos principais módulos de hardware e software (AFANASEV et al., 2017).

Após análise, o artigo foi enquadrado nas categorias: Artigo de experiência, Pesquisa de avaliação, Abordagem, Novo sistema, Manutenção, Princípios reativos e Arquitetura.

5.1.7 *An architecture for self-managing microservices*

O artigo explica que para executar eficientemente aplicações na nuvem é necessário mais do que apenas a implantação do software em máquinas virtuais. As aplicações executadas na nuvem precisam ser gerenciadas continuamente, ajustando seus recursos à carga recebida e enfrentando falhas momentâneas enquanto realiza replicação e reinicia componentes, para oferecer resiliência em infra-estrutura não confiável. O gerenciamento contínuo procura reagir de maneira automatizada e responsiva às falhas e mudanças nas condições do ambiente de execução da aplicação, minimizando a intervenção humana (TOFFETTI et al., 2015).

Ainda no artigo é exposto que o uso de serviços de terceiros, que fornecem infra-estrutura para funcionalidades de gerenciamento, resulta em limitações que impedem que aplicações em nuvem se adaptem e sejam executados na máquina mais eficaz para o trabalho.

No trabalho os autores se posicionam e propõem uma nova arquitetura para possibilitar o autogerenciamento escalonável e resiliente de aplicações de microserviços na nuvem. Os autores desse estudo são Giovanni Toffetti, Sandro Brunner, Martin Blöchlinger, Florian Dudouet e Andrew Edmonds e foi obtido através da base de dados ACM. Publicado em 2015 no AIMC '15: *1st International Workshop on Automated Incident Management in Cloud*.

As categorias que podem ser associadas a este trabalho são: Artigo de opinião, Proposta de solução, Modelo, Abordagem, Novo sistema, Escalabilidade, Confiabilidade, Princípios reativos e Arquitetura.

5.1.8 *An open IoT framework based on microservices architecture*

O trabalho que tem como autores Long Sun, Yan Li e Raheel Ahmed Memon foi publicado em 2017 na revista *China Communications*, foi obtido pela base de dados IEEEExplore. Nele é explicado que com o contínuo desenvolvimento e evolução da Internet das Coisas (IoT), a aplicação monolítica se torna muito maior em escala e ainda mais complexa em estrutura. Levando a uma baixa escalabilidade, extensibilidade e dificultando a manutenção. Em resposta a esses desafios, a arquitetura de microserviço foi introduzida no campo da aplicação de IoT, devido à sua flexibilidade e facilidade de acoplamento.

No entanto, a estrutura de microserviços de IoT existente concentra-se principalmente em um domínio específico, portanto, isso limita muito sua aplicação. Neste artigo, é proposta uma estrutura geral do sistema de microserviço para aplicação em IoT, que é uma arquitetura melhor escalonável, extensível e de fácil manutenção. É apresentado o design do sistema e os microserviços relacionados, enfatizando o serviço principal e a comunicação. Com o *framework* apresentado é possível se obter com mais facilidade a integração de novas aplicações (SUN; LI; MEMON, 2017).

É possível enquadrar o estudo nas seguintes categorias: Pesquisa de avaliação, Proposta de solução, Framework, Novo sistema, Escalabilidade, Evolução, Manutenção e Arquitetura.

5.1.9 *Asynchronous Queue Based Approach for Building Reactive Microservices*

O artigo destaca que na abordagem de microserviços, em termos de escopo, eficiência e confiabilidade, um dos tópicos mais difíceis é como migrar ou desenvolver um microserviço.

Neste sentido, o trabalho propõe um novo modelo arquitetural, baseando a estrutura interna de um novo microserviço no padrão de arquitetura de alto nível da programação reativa (BRILHANTE; COSTA; MARITAN, 2017).

Ainda no artigo é explicado que o novo modelo de microserviços é coordenado internamente por filas assíncronas, o que permite preservar a compatibilidade com a maioria dos componentes monolíticos e fornece um processo de encapsulamento para permitir sua continuidade. Além do que já foi citado, ainda é apresentado um estudo comparativo entre a abordagem padrão e a arquitetura proposta, para medir a eventual melhoria de desempenho da nova estratégia.

O trabalho que tem como autores Jonathan Brilhante, Rostand Costa e Tiago Maritan foi publicado em 2017 no *WebMedia '17: 23rd Brazillian Symposium on Multimedia and the Web*, foi obtido pela base de dados ACM. As categorias observadas nele são: Proposta de solução, Modelo, Abordagem, Migração, Evolução, Desempenho, Confiabilidade, Princípios reativos e Arquitetura.

5.1.10 Case Study: Microservice Evolution and Software Lifecycle of the XSEDE User Portal API

Obtido na base de dados ACM, o trabalho de Walter Scarborough, Carrie Arnold e Maytal Dahan foi publicado no XSEDE16: *Conference on Diversity, Big Data, and Science at Scale* em 2016.

O artigo apresenta uma evolução da arquitetura monolítica para interface de programação de aplicativos (API) alimentada por um conjunto de microserviços. Durante essa transição foram desenvolvidas diretrizes para serviços de API, que buscam equilibrar as necessidades de complexidade e de reuso com os requisitos de flexibilidade (SCARBOROUGH; ARNOLD; DAHAN, 2016).

Os autores também criaram um conjunto próprio de práticas recomendadas sobre como fazer a conversão para o uso de microserviços. Utilizaram como estudo de caso o desenvolvimento da API de um portal, chamado XSEDE, para explicar a abordagem, e para relatar a experiência de trabalhar com microserviços em um ambiente de produção real.

No estudo percebe-se as categorias: Artigo de experiência, Pesquisa de avaliação, Proposta de solução, Método, Técnica, Diretriz, Migração, Evolução, Requisitos, Implementação e Implantação.

5.1.11 *Container-based microservice architecture for cloud applications*

O trabalho que tem como autores Vindeep Singh e Sateesh K Peddoju foi publicado em 2017 na *International Conference on Computing, Communication and Automation (ICCCA)*, foi obtido pela base de dados IEEEExplore. Ele apresenta uma análise de diferentes desafios na implantação, e integração contínua de microserviços. E também propõe e projeta um sistema automatizado para superar esses desafios.

No trabalho foi realizado um estudo de caso após a implementação da arquitetura de microserviços proposta, e os resultados são apresentados realizando uma comparação entre o desempenho da abordagem monolítica e de microserviço, usando vários parâmetros, como tempo de resposta, taxa de transferência, tempo de implementação, etc.

Nos resultados é observado que a aplicação desenvolvida usando a abordagem de microserviço, e implantada usando o design proposto reduziu o tempo e o esforço para implantação e integração contínua da aplicação. Os resultados também mostram que a aplicação baseada em microserviço supera o *design* monolítico devido ao seu baixo tempo de resposta e alto rendimento (SINGH; PEDDOJU, 2017).

As categorias que se enquadram à este artigo são: Artigo de experiência, Pesquisa de avaliação, Processo, Modelo, Abordagem, Novo sistema, Evolução, Manutenção, Desempenho, Arquitetura, Implementação e Implantação.

5.1.12 *Cyber-physical microservices: An IoT-based framework for manufacturing systems*

Neste artigo, é adotado o conceito de microserviço e descrito um *framework* para sistemas de manufatura que tem o microserviço ciber-físico como base. Os processos da fábrica são definidos como composições de microserviços ciber-físicos primitivos, adotando o padrão de orquestração ou de coreografia (THRAMBOULIDIS; VACHTSEVANOU; SOLANOS, 2018).

Ainda no artigo é citado que as tecnologias de *Internet of Things* (IoT) são usadas para integração de sistemas e a engenharia baseada em modelos é utilizada para semi-automatizar o processo de desenvolvimento para o engenheiro industrial, que não está familiarizado com microserviços e com IoT. Dois estudos de caso demonstram a viabilidade da abordagem proposta.

Os autores desse estudo são Kleanthis Thramboulidis, Danai C. Vachtsevanou e Alexan-

dros Solanos e foi obtido através da base de dados IEEEExplore. Publicado em 2018 na IEEE *Industrial Cyber-Physical Systems* (ICPS).

O trabalho é compatível com as categorias: Pesquisa de validação, Proposta de solução, Modelo, Framework, Abordagem, Novo sistema, Escalabilidade, Manutenção, Arquitetura e Implantação.

5.1.13 *Decision guidance models for microservices: service discovery and fault tolerance*

O trabalho que tem como autores Stefan Haselböck, Rainer Weinreich e Georg Buchgeher foi publicado em 2017 na ECBS '17: *Fifth European Conference on the Engineering of Computer-Based Systems*, foi obtido pela base de dados ACM.

O estudo fala sobre a introdução de um sistema de microserviço ser uma tarefa desafiadora que requer a exploração e documentação de várias áreas relacionadas do *design*. A exploração e a documentação do projeto de arquitetura de software são apoiadas por modelos de orientação de decisão na arquitetura de software. No artigo, são apresentados modelos de orientação para várias áreas de projeto de sistemas de microserviço, incluindo descoberta de serviço e tolerância a falhas (HASELBÖCK; WEINREICH; BUCHGEHER, 2017).

Os modelos apresentados pelos autores foram criados com base na literatura de microserviço existente e foram validados e aperfeiçoados em oficinas de *design* com parceiros de negócios como parte de um estudo de pesquisa de ação técnica (*Technical Action Research* - TAR).

As categorias presentes neste artigo, percebidas após a análise são: Artigo de experiência, Pesquisa de validação, Modelo, Diretriz, Novo sistema, Princípios reativos, Requisitos e Arquitetura.

5.1.14 *Design and Development of Backend Application for Public Complaint Systems Using Microservice Spring Boot*

O objetivo da pesquisa apresentada é desenvolver um sistema que ofereça um serviço de reclamações públicas, baseadas em aplicações *web* que utilizem arquitetura de microserviço *springboot*. A arquitetura de microserviço foi usada para dividir a funcionalidade da aplicação em muitas partes, ou muitos microserviços, baseados em processos de negócios. Os serviços são interconectados, tornando-se um único sistema com um processo de negócios completo. Uma

das vantagens dessa arquitetura é que mais microserviços podem ser adicionados sem afetar outros (SURYOTRISONGKO; JAYANTO; TJAHYANTO, 2017).

O trabalho que tem como autores Hatma Suryotrisongko, Dedy Puji Jayanto e Aris Tjahyanto foi publicado em 2017 na *4th Information Systems International Conference (ISICO)*, foi obtido pela base de dados ScienceDirect.

As categorias deste estudo são: Artigo de experiência, Artigo filosófico, Processo, Modelo, Abordagem, Novo sistema, Evolução, Manutenção, Requisitos e Arquitetura.

5.1.15 *From Monolith to Microservices: A Dataflow-Driven Approach*

O trabalho que tem como autores Rui Chen, Shanshan Li e Zheng Li foi publicado em 2017 na *24th Asia-Pacific Software Engineering Conference (APSEC)*, foi obtido pela base de dados IEEEExplore.

O estudo fala sobre o desafio e a complexidade de realizar uma decomposição orientada para microserviços a partir de um sistema monolítico, e sobre o papel crucial dessa decomposição no desenvolvimento de sistemas de software baseados em microserviço. E para enfrentar esse desafio e reduzir a complexidade, os autores propuseram uma abordagem de análise *top-down* e desenvolveram um algoritmo de decomposição orientado a fluxo de dados (CHEN; LI; LI, 2017).

Para demonstrar o mecanismo de identificação de microserviço, são apresentados dois casos de uso, também são feitas comparações com uma ferramenta de identificação de microserviço existente. A comparação e a avaliação mostram que, o mecanismo de identificação orientado a fluxo de dados é capaz de fornecer candidatos de microserviço mais racionais, objetivos, compreensíveis e consistentes, por meio de um procedimento de implementação mais rigoroso e prático.

Após análise, o artigo foi enquadrado nas categorias: Artigo de experiência, Pesquisa de avaliação, Proposta de solução, Framework, Abordagem, Migração, Evolução, Requisitos, Arquitetura e Implementação.

5.1.16 *Highly-Available Applications on Unreliable Infrastructure: Microservice Architectures in Practice*

O trabalho que tem como autores Daniel Richter, Marcus Konrad, Katharina Utecht e Andreas Polze foi publicado em 2017 na IEEE *International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, foi obtido pela base de dados IEEEExplore.

O artigo cita que uma empresa ferroviária alemã (*Deutsche Bahn AG*) utiliza um sistema chamado EPA¹ para reservas de assentos em serviços ferroviários interurbanos. Apesar de sua alta disponibilidade, o sistema EPA em seu estado atual tem várias desvantagens, como alto custo operacional, necessidade de hardware especial, dependências tecnológicas e atualizações caras e demoradas (RICHTER et al., 2017).

Os autores utilizaram um protótipo para avaliar as propriedades gerais de uma arquitetura de microserviço, e a sua confiabilidade em relação ao sistema legado. Eles focaram nos requisitos de um sistema baseado em microserviço e no processo de migração. Os requisitos seriam serviços e dados, containerização, comunicação via filas de mensagens, como também alcançar tolerância a falhas e alta disponibilidade com a ajuda da replicação dentro da arquitetura resultante.

As categorias deste estudo são: Artigo de experiência, Pesquisa de avaliação, Proposta de solução, Modelo, Migração, Escalabilidade, Evolução, Manutenção, Confiabilidade, Princípios reativos, Arquitetura, Implementação e Testes.

5.1.17 *Microservice-based architecture for the NRDC*

Este trabalho apresenta uma nova solução de arquitetura para o NRDC. Com base em microserviços, a solução visa garantir escalabilidade, confiabilidade e capacidade de manutenção deste *data center*. O histórico do NRDC é fornecido no documento, juntamente com detalhes sobre a especificação do software da solução proposta, projeto e implementação do protótipo. Uma discussão sobre os benefícios da arquitetura baseada em microserviço, e um esboço das direções para trabalho futuro também são apresentados (LE et al., 2015).

Os autores deste artigo são Vinh D. Le, Melanie M. Neff, Royal V. Stewart, Richard Kelley, Eric Fritzinger, Sergiu M. Dascalu e Frederick C. Harris e foi obtido através da base de

¹Informações sobre a empresa que utiliza o sistema disponível em: <https://www.deutschebahn.com/>

dados IEEEExplore. Publicado em 2015 na IEEE *13th International Conference on Industrial Informatics* (INDIN).

Ao analisar o estudo, as categorias compatíveis são: Artigo de experiência, Proposta de solução, Processo, Modelo, Novo sistema, Escalabilidade, Manutenção, Confiabilidade, Requisitos, Arquitetura e Implementação.

5.1.18 *Microservices approach for the internet of things*

Obtido na base de dados IEEEExplore, o trabalho de Björn Butzin, Frank Golatowski e Dirk Timmermann foi publicado na IEEE *21st International Conference on Emerging Technologies and Factory Automation* (ETFA) em 2016.

O estudo cita que na Internet das coisas, as aplicações precisam ser organizadas a partir de um conjunto de serviços pequenos e independentes. Assim, a criação de novos serviços exigiria a combinação de serviços de diferentes fornecedores para aproveitar totalmente a heterogeneidade da IoT. Embora a direção seja diferente, muitos dos requisitos de microserviços são semelhantes aos da Internet das coisas (BUTZIN; GOLATOWSKI; TIMMERMANN, 2016).

Neste artigo são investigados padrões e melhores práticas que são usados na abordagem de microserviços, e como eles podem ser usados na internet das coisas. Os autores buscam entender como as empresas que usam microserviços fizeram considerações sobre a forma como os serviços devem ser projetados para funcionarem juntos adequadamente. As aplicações de IoT podem adotar várias dessas decisões de design para melhorar a capacidade de criar aplicações a partir de uma infinidade de serviços.

As categorias adequadas à este trabalho, percebidas após análise são: Artigo de experiência, Proposta de solução, Método, Abordagem, Diretriz, Novo sistema, Evolução, Requisitos e Arquitetura.

5.1.19 *Microservices architecture: Case on the migration of reservation-based parking system*

Este artigo explora a migração de um sistema de estacionamento baseado em reservas para a arquitetura de microserviço. Ele cita que no início o sistema de estacionamento era uma arquitetura monolítica, com tecnologia baseada na *web*. Ele relata a jornada para encontrar

a arquitetura de microserviço se baseando na arquitetura monolítica (YUGOPUSPITO; PANDUWINATA; SUTRISNO, 2017).

O estudo enfatiza a existência de microserviços como um refinamento da arquitetura orientada a serviços, e também apresenta o impacto das mudanças nos processos de negócios devido às experiências do usuário.

O trabalho que tem como autores Pujianto Yugopuspito, Frans Panduwinata e Sutrisno Sutrisno foi publicado em 2017 na *IEEE 17th International Conference on Communication Technology* (ICCT), foi obtido pela base de dados IEEEExplore.

As categorias presentes no estudo são: Artigo de experiência, Pesquisa de avaliação, Processo, Migração, Escalabilidade, Confiabilidade e Princípios reativos.

5.1.20 *Microservices model in WoO based IoT platform for depressive disorder assistance*

Sajjad Ali, Muhammad Golam Kibria, Muhammad Aslam Jarwar, Sunil Kumar e Ilyoung Chong publicaram seu artigo na *International Conference on Information and Communication Technology Convergence* (ICTC) em 2017. O artigo foi encontrado na base de dados IEEEExplore.

Os autores comentam no estudo que um grande número de dispositivos heterogêneos conectados, e uma quantidade enorme de dados gerados por esses dispositivos não são apenas difíceis de gerenciar, e que ao fornecer serviços com eles em um ambiente dinâmico de IoT, tornam o desafio ainda maior. Eles afirmam haver uma necessidade de mecanismos que não apenas atendam à heterogeneidade dos dispositivos de IoT, mas também forneçam serviços dinâmicos com eficiência, com base nos requisitos do usuário (ALI et al., 2017).

Ainda no estudo, é citado que a plataforma *Web of Objects* (WoO) permite serviços IoT virtualizando objetos do mundo real usando ontologia semântica. Ele fornece uma base para o desenvolvimento de microserviços interoperáveis que suportam recursos dinâmicos, utilizando fluxos de dados sensoriais. Este artigo propõe um modelo de microserviços na plataforma IoT baseada em WoO, para fornecer serviços assistenciais ao transtorno depressivo. Além disso, um modelo de Conhecimento do Mundo Real (*Real World Knowledge* - RWK) de situações e emoções do usuário também foi desenvolvido.

Ao analisar o trabalho, as categorias percebidas são: Artigo de experiência, Proposta de

solução, Modelo, Novo sistema, Escalabilidade, Desempenho, Princípios reativos, Requisitos e Arquitetura.

5.1.21 *Microservices-based software architecture and approaches*

Kapil Bakshi publicou seu artigo na *IEEE Aerospace Conference* em 2017. Ele foi encontrado na base de dados IEEEExplore e apresenta uma discussão sobre vários aspectos da arquitetura baseada em microserviços, incluindo vários casos de uso com possível utilização na indústria aeroespacial. As características da arquitetura baseada em microserviço, tais como componentização, organização, *endpoints* e mecanismos de mensagens são discutidas (BAKSHI, 2017).

Também é apresentada a implementação técnica de microserviços, revisando a conteneurização, a comunicação de serviços e os componentes arquiteturais relacionados. Projetos e componentes específicos de software livre que podem ser utilizados para construir arquitetura baseada em microserviços são expostos no trabalho.

As categorias presentes são: Pesquisa de validação, Modelo, Abordagem, Novo sistema, Princípios reativos, Arquitetura e Implementação.

5.1.22 *Migrating Monolithic Mobile Application to Microservice Architecture: An Experiment Report*

Chen-Yuan Fan e Shang-Pin Ma publicaram seu artigo na *IEEE International Conference on AI & Mobile Services (AIMS)* em 2017. O artigo foi encontrado na base de dados IEEEExplore.

O estudo cita que as técnicas de *DevOps* são comumente usadas para automatizar o processo de desenvolvimento e operação por meio de integração e implantação contínua. O monitoramento de sistemas de software criados pelo *DevOps* permite que a arquitetura de microserviço (MSA) obtenha o *feedback* necessário para melhorar o sistema de maneira rápida e fácil. No entanto, métodos sistemáticos, orientados por ciclo de vida de desenvolvimento de software (CVDS) são insuficientes para facilitar a migração de sistemas de software de uma arquitetura monolítica tradicional para a MSA (FAN; MA, 2017).

Para facilitar o processo de migração os autores propõem um processo de migração com base no CVDS, incluindo todos os métodos e ferramentas necessários durante o projeto, desenvolvimento e implementação. Um aplicativo móvel, *EasyLearn*, foi usado como exemplo ilustrativo

para demonstrar a eficácia do processo de migração proposto. Os autores comentam acreditar que o trabalho deles pode fornecer referências valiosas para outras equipes de desenvolvimento que buscam facilitar a migração de aplicativos existentes para a MSA.

São enquadradas neste estudo as seguintes categorias: Artigo de experiência, Proposta de solução, Processo, Migração, Evolução, Requisitos, Arquitetura Implementação, Testes e Implantação.

5.1.23 *Migrating web applications to clouds with microservice architectures*

No trabalho os autores comentam que com os recentes avanços das tecnologias de *Cloud Computing*, aplicações executadas na nuvem têm se tornado populares pelo seu rico conjunto de recursos. As aplicações na nuvem são reconhecidas como uma tendência para a próxima geração de aplicativos *Web* e, portanto, como migrar esses sistemas da *Web* para as nuvens torna-se um campo desejado na literatura. Com intenção de suprir essa necessidade, eles apresentam um método de migração que emprega as arquiteturas de microserviço conhecidas para oferecer suporte a uma migração efetiva de aplicativos da *Web* para as nuvens (LIN; LIN; HUANG, 2016).

Obtido na base de dados IEEEExplore, o trabalho de Jyhjong Lin, Lendy Chaoyu Lin e Shiche Huang foi publicado na *International Conference on Applied System Innovation (ICASI)* em 2016, e as categorias presentes nele são: Artigo de experiência, Proposta de solução, Processo, Método, Migração, Escalabilidade, Manutenção, Requisitos e Arquitetura.

5.1.24 *PipelineX: a feature animation pipeline on microservices*

Dan Golembeski, Ray Forziati, Ben George e Doug Sherman publicaram seu artigo no *DigiPro '17: ACM SIGGRAPH Digital Production Symposium* em 2017. O estudo foi encontrado na base de dados ACM e nele é apresentada uma abordagem exclusiva para a criação de um *pipeline* de animação de recursos altamente escalável, multifuncional e amigável à produção em uma infraestrutura central composta de microserviços (GOLEMBESKI et al., 2017).

No trabalho também é discutido o *design* básico da camada de serviço, bem como os benefícios e desafios de mover processos de produção, usados por décadas, de um estúdio de animação inteiro para um novo *pipeline* que opera com tecnologia compartimentada. O objetivo do estudo

é limpar a desordem de um *pipeline* legado e permitir um ambiente de produção mais flexível, usando tecnologia moderna baseada na *web*.

O estudo contém as seguintes categorias: Artigo de experiência, Pesquisa de avaliação, Proposta de solução, Abordagem, Migração, Escalabilidade, Evolução, Requisitos, Arquitetura e Implementação.

5.1.25 *Publishing linked data through semantic microservices composition*

Obtido na base de dados ACM, o trabalho de Ivan Salvadori, Alexis Huf, Ronaldo dos Santos Mello e Frank Siqueira foi publicado na *iiWAS '16: 18th International Conference on Information Integration and Web-based Applications and Services* em 2016.

No artigo, os autores citam que os microserviços estão substituindo os aplicativos monolíticos, dividindo-os em artefatos pequenos e independentes que colaboram entre si. Focado em gerenciar informações altamente coesas, os microserviços podem ser compostos para fornecer informações mais ricas e vinculadas entre si (SALVADORI et al., 2016).

No estudo é apresentado um método de composição, visando compor microserviços semânticos para alcançar a integração de dados com base nos princípios de *Linked Data*. Além disso, o método proposto aproveita a independência do desenvolvimento e capacidade de composição dos microserviços. Este artigo também apresenta um *framework* e um estudo de caso para o método proposto.

As categorias deste trabalho são: Proposta de solução, Método, Framework, Novo sistema, Evolução, Manutenção, Desempenho, Requisitos e Arquitetura.

5.1.26 *Real-Time HazMat Environmental Information System: A micro-service based architecture*

No trabalho apresentado os autores citam que os produtos perigosos são importantes para as empresas e a vida de uma cidade. Milhares de toneladas de óleo, materiais tóxicos, químicos, corrosivos, inflamáveis e radioativos são transportados a cada dia. No entanto, um acidente envolvendo materiais perigosos poderia acarretar em sérias consequências para os usuários das estradas, para a infraestrutura e para o meio ambiente.

Para reduzir os riscos de transportar materiais perigosos é proposto um sistema de auxílio ao planejamento do transporte, e nele é adotada uma arquitetura baseada em microserviços. Esse

tipo de arquitetura consiste em um conjunto de serviços fracamente acoplados e de implementação independente, para se obter aplicações mais escalonáveis. Devido à natureza distribuída dos micros serviços, o sistema será desenvolvido como um conjunto de pequenos serviços, que serão alinhados ao processo de gerenciamento de riscos. Cada serviço será executado em sua própria máquina lógica ou tecnologia de contêineres, como o Docker por exemplo (CHERRADI et al., 2017).

Ghyzlane Cherradi, Adil El Bouziri, Azedine Boulmakoul e Karine Zeitouni publicaram seu artigo no *The 7th International Symposium on Frontiers in Ambient and Mobile Systems (FAMS)* em 2017. O artigo foi encontrado na base de dados ScienceDirect e nele percebe-se as seguintes categorias: Artigo filosófico, Proposta de solução, Modelo, Diretriz, Novo sistema, Escalabilidade, Evolução, Manutenção, Arquitetura, Implementação e Testes.

5.1.27 Scalable microservice based architecture for enabling DMTF profiles

Os autores desse estudo são Divyanand Malavalli e Sivakumar Sathappan e foi obtido através da base de dados IEEEExplore. Publicado em 2015 na *11th International Conference on Network and Service Management (CNSM)*.

Este artigo propõe uma arquitetura para implementação de perfis de gerenciamento DMTF na camada de *middleware* de um console de gerenciamento, usando micros serviços. Microserviço é um estilo de arquitetura de software, ganhando popularidade para realizar o desenvolvimento de aplicativos escaláveis na Internet. Este documento fornecerá uma comparação com o atual design de método predominante, e também, discutirá como esse microserviço pode ser exposto utilizando o REST, tornando-o mais escalável, leve, etc (MALAVALLI; SATHAPPAN, 2015).

Após analisar o trabalho, as categorias que se enquadram à ele são: Pesquisa de avaliação, Proposta de solução, Modelo, Técnica, Novo sistema, Escalabilidade, Desempenho, Princípios reativos, Arquitetura e Implementação.

5.1.28 Service fabric: a distributed platform for building microservices in the cloud

Neste trabalho, os autores descrevem o *Service Fabric (SF)*, uma plataforma distribuída da *Microsoft* para construir, executar e manter aplicações de microserviço na nuvem. SF é

executado em produção há mais de 10 anos, alimentando muitos serviços críticos na *Microsoft*. Este artigo descreve as principais filosofias de *design* em SF.

É adotada uma abordagem *bottom-up* para descrever componentes de baixo nível em sua arquitetura, com foco no uso modular e suporte para semântica forte, como tolerância a falhas e consistência dentro de cada componente do SF. É realizada uma discussão sobre lições aprendidas e apresentados resultados experimentais a partir de dados de produção (KAKIVAYA et al., 2018).

Os autores desse estudo são Gopal Kakivaya, Lu Xun, Richard Hasha, Shegufta Bakht Ahsan, Todd Pfeiger, Rishi Sinha, Anurag Gupta, Mihail Tarta, Mark Fussell, Vipul Modi, Mansoor Mohsin, Ray Kong, Anmol Ahuja, Oana Platon, Alex Wun, Matthew Snider, Chacko Daniel, Dan Mastrian, Yang Li, Aprameya Rao, Vaishnav Kidambi, Randy Wang, Abhishek Ram, Sumukh Shivaprakash, Rajeet Nair, Alan Warwick, Bharat S. Narasimman, Meng Lin, Jeffrey Chen, Abhay Balkrishna Mhatre, Preetha Subbarayalu, Mert Coskun e Indranil Gupta e foi obtido através da base de dados ACM. Publicado em 2018 na *EuroSys '18: Thirteenth EuroSys Conference*.

As categorias presentes neste artigo são: Artigo de experiência, Pesquisa de avaliação, Framework, Novo sistema, Princípios reativos, Arquitetura, Implementação e Implantação.

5.1.29 *SmartVM: A Multi-Layer Microservice-Based Platform for Deploying SaaS*

Xi Zheng, Jiaojiao Jiang, Yuqun Zhang, Yao Deng, Min Fu, Tianlei Zheng e Xiao Liu publicaram seu artigo no IEEE *International Symposium on Parallel and Distributed Processing with Applications* e na IEEE *International Conference on Ubiquitous Computing and Communications* (ISPA/IUCC) em 2017. O artigo foi encontrado na base de dados IEEEExplore.

No estudo é explicado que com o advento do software como serviço (*Software-as-a-Service - SaaS*), os desenvolvedores de SaaS estão enfrentando muitos desafios associados à multilocação e ao aumento drástico do número de usuários (ZHENG et al., 2017).

No artigo os autores apresentam uma nova plataforma, denominada SmartVM, e ela foi criada para permitir que o desenvolvedor de SaaS crie, personalize e implante soluções SaaS em multi-camadas, uma abordagem baseada em microserviço.

Foi desenvolvido um protótipo de e-commerce SaaS para avaliar a eficácia e a eficiência

do SmartVM. Os resultados mostram que as implementações do SmartVM superam as implantações monolíticas, e até as de microserviço convencionais, nos quesitos de monitoramento inteligente, redução de custos e otimização de recursos.

As categorias presentes neste trabalho são: Artigo de experiência, Pesquisa de avaliação, Framework, Abordagem, Novo sistema, Escalabilidade, Evolução, Requisitos, Arquitetura, Implementação e Testes.

5.1.30 *The ENTICE approach to decompose monolithic services into micro-services*

Obtido na base de dados IEEEExplore, o trabalho de Gabor Kecskemeti, Attila Csaba Marosi e Attila Kertesz foi publicado na *International Conference on High Performance Computing & Simulation* (HPCS) em 2016.

O artigo apresenta que apesar da existência de tecnologias capacitadoras, as aplicações orientadas a serviços em grande escala ainda são inelásticos em grande parte. Tais aplicações freqüentemente usam serviços monolíticos que limitam a elasticidade (obstruindo a replicabilidade de partes de um serviço monolítico, por exemplo). A decomposição desses serviços (levando a serviços menores, mais direcionados e mais modulares) poderiam possibilitar elasticidade, mas o processo de decomposição ocorre principalmente de forma manual (KECSKEMETI; MAROSI; KERTESZ, 2016).

Este artigo apresenta uma metodologia para decompor serviços monolíticos em vários microserviços. A metodologia proposta aplica vários resultados do projeto ENTICE² (ou seja, suas ferramentas de síntese e otimização de imagens). Finalmente, o artigo fornece percepções sobre como esses resultados ajudam a revitalizar os serviços monolíticos e quais técnicas são aplicadas para ajudar futuros desenvolvedores de microserviços.

Após analisar o trabalho é possível enquadrá-lo nas categorias: Artigo de experiência, Proposta de solução, Método, Abordagem, Técnica, Migração, Escalabilidade, Princípios reativos, Requisitos, Arquitetura, Implementação, Testes e Implantação.

²Projeto ENTICE. Disponível em: <http://www.entice-project.eu/>

5.1.31 *Towards a microservices development approach for the crisis management field in developing countries*

Este artigo descreve uma primeira tentativa de alinhar recursos de tecnologia da informação com processos de negócios, no campo de gerenciamento de crise em países em desenvolvimento. Os autores propõem fornecer aos projetistas/desenvolvedores de serviços uma estrutura abrangente para o projeto, implementação e implantação de sistemas de software baseados em microserviços. A proposição é ilustrada por um estudo de caso na Argélia (IDOUGHI; ABDELOUHAB; KOLSKI, 2017).

Djilali Idoughi, Karima Ait Abdelouhab e Christophe Kolski publicaram seu trabalho na *4th International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)* em 2017. O estudo foi encontrado na base de dados IEEEExplore.

As categorias observadas são: Artigo de experiência, Pesquisa de validação, Proposta de solução, Framework, Novo sistema, Requisitos, Arquitetura, Implementação, Testes e Implantação.

5.1.32 *Transform Monolith into Microservices using Docker*

Neste artigo, é apresentada uma abordagem sobre como transformar um sistema monolítico existente em arquitetura de microserviços, implantar e fornecer usando o *Docker*, da mesma forma que uma organização faz com desenvolvimento ágil, escalabilidade e capacidade de migrar para a nuvem (SARITA; SEBASTIAN, 2017).

As contribuições do trabalho são: a) Uma abordagem para transformar aplicações existentes em arquitetura de microserviços b) Estratégia de implantação de microserviços usando o *Docker* c) Padrão de descoberta de serviço d) Estratégia proposta para construir e enviar microserviços usando o *Docker*.

Sarita e Sunil Sebastian publicaram este estudo na *International Conference on Computing, Communication, Control and Automation (ICCUBEA)* em 2017. O artigo foi encontrado na base de dados IEEEExplore e as categorias percebidas foram: Artigo de experiência, Proposta de solução, Modelo, Abordagem, Técnica, Migração, Requisitos, Arquitetura, Implementação, Testes e Implantação.

5.1.33 *Using microservices in educational applications of IT-company*

Os sistemas educacionais das empresas de TI normalmente fornecem um espaço de rede no qual alunos e professores podem se unir para ensinar e aprender. O uso da arquitetura de microserviço oferece vantagens no desenvolvimento dessas aplicações (BERKUNSKYI et al., 2017).

Os autores explicam que a análise das soluções arquitetônicas existentes mostrou que a arquitetura de microserviço é um mecanismo para aumentar a eficácia da construção de plataformas educacionais corporativas por meio do uso de soluções flexíveis, em comparação com a abordagem clássica baseada na arquitetura monolítica.

Yevhen Berkunskyi, Kateryna Knyrik, Tetyana Farionova e Tetiana Smykodub publicaram seu artigo na *IEEE First Ukraine Conference on Electrical and Computer Engineering (UKR-CON)* em 2017. O artigo foi encontrado na base de dados IEEEExplore e as categorias presentes são: Pesquisa de avaliação, Proposta de solução, Modelo, Abordagem, Novo sistema, Evolução e Manutenção.

Ao final da análise de todos os artigos, obtemos uma visão um pouco mais aprofundada sobre cada um dos estudos selecionados pós-filtragem. Neste momento temos também quais categorias foram compatíveis com cada um dos trabalhos. No Capítulo 6 será realizada uma análise da forma como as categorias foram distribuídas entre os trabalhos, a partir da observação de tabelas e gráficos gerados a partir destes dados.

Capítulo 6

Análise dos Resultados

Neste Capítulo serão apresentados dados obtidos após a análise dos estudos selecionados, que é apresentada na Seção 5. Após observação serão realizados comentários sobre o que pode ser percebido a partir da relação entre as informações.

6.1 Categorização dos Estudos

Esta sessão apresenta como as categorias propostas no planejamento do mapeamento sistemático se distribuíram entre os estudos selecionados. Para facilitar a leitura a distribuição é analisada em dois momentos: o primeiro na sessão 6.1.1 observando a distribuição das categorias das duas primeiras facetas apresentadas (“Tipo de Pesquisa” e “Tipo de Contribuição”); o segundo na sessão 6.1.2 analisando como as categorias das três últimas facetas se distribuíram (“Tipo de desenvolvimento”, “Motivação para adoção de microserviços” e “Fases do processo de software”).

6.1.1 Primeiras Facetas

Das categorias propostas no Capítulo de planejamento, todas se enquadraram em pelo menos um estudo analisado. Na Tabela 6.1 é apresentado como ocorreu a distribuição das categorias das facetas “Tipo de Pesquisa” e “Tipo de Contribuição” entre os artigos. A primeira coluna da tabela representa a categoria, a segunda coluna representa qual a porcentagem dos artigos em que aquela categoria se enquadra, e a terceira coluna representa o valor real de quantos estudos aquela categoria apareceu.

| Categoria | Porcentagem dos estudos | Quantidade de estudos |
|-----------|-------------------------|-----------------------|
|-----------|-------------------------|-----------------------|

| | | |
|-----------------------|------|----|
| Artigo de experiência | 72,7 | 24 |
| Artigo filosófico | 6,1 | 2 |
| Artigo de opinião | 3,0 | 1 |
| Pesquisa de avaliação | 45,5 | 15 |
| Pesquisa de validação | 12,1 | 4 |
| Proposta de solução | 72,7 | 24 |
| Processo | 21,2 | 7 |
| Método | 21,2 | 7 |
| Modelo | 48,5 | 16 |
| Framework | 21,2 | 7 |
| Abordagem | 51,5 | 17 |
| Técnica | 15,2 | 5 |
| Diretriz | 12,1 | 4 |

Tabela 6.1: Distribuição das categorias das duas primeiras facetas entre os trabalhos

A porcentagem apresentada na Tabela 6.1 contém apenas uma casa decimal de precisão, o que pode ocasionar pequenos erros de arredondamento na representação, mas que não exercem influência significativa na análise dos dados.

As categorias que aparecem em mais estudos são “Artigo de experiência” e “Proposta de solução”, ambas sendo compatíveis com 72,7% dos artigos. A partir deste dado é possível perceber que, entre os artigos que falam sobre abordagens para criação de microserviços, a maioria ainda busca solucionar um problema. Seja um problema encontrado com a própria utilização da arquitetura de microserviços, ou problemas arquiteturais de sistemas legados monolíticos onde a solução proposta é a utilização da arquitetura de microserviços. O fato de haver tantos artigos relatando experiências demonstra que há interesse em solucionar os desafios da arquitetura para que ela possa ser amplamente utilizada.

A categoria “Pesquisa de avaliação” ocorreu em 45,5%. Nestes estudos foram apresentadas técnicas implementadas, sempre dentro do contexto de abordagens para criação de microserviços, e realizada uma avaliação daquela técnica. A partir disso percebe-se como as abordagens ainda não estão maduras, pois ainda são realizados testes para saber se aquela técnica é uma boa solução. Seria necessário ainda realizar uma análise comparativa da qualidade das diferentes técnicas apresentadas entre os diferentes estudos para se chegar a um consenso de qual a melhor técnica para cada situação.

Artigos contendo as categorias “Modelo” e “Abordagem” representam, respectivamente,

48,5% e 51,5% dos estudos analisados. Porém foi percebido que essas duas categorias apareciam em quase todos os estudos, exceto 8. Dessa forma 75,8% dos trabalhos apresentam modelos ou abordagens para criação de microserviços, o que pode demonstrar que os critérios de seleção dos estudos serviram ao seu propósito, e que os selecionados realmente são relevantes para responder a pergunta de pesquisa.

Dentre os 8 que não continham essas duas categorias, 4 continham a categoria “*Framework*” e apresentavam uma ferramenta desenvolvida com intenção de automatizar o processo de criação de microserviços. A utilização dessas ferramentas pode ser considerada uma abordagem para criação de microserviços, assim, pode-se dizer que 87,9% dos estudos falam explicitamente sobre o que se esperava encontrar. Os 12,1% restantes obviamente também falam sobre a arquitetura de microserviços, caso contrário não teriam passado pelo primeiro critério de inclusão, porém se baseiam na estrutura da arquitetura de microserviços para utilização em outros meios.

Ainda sobre a categoria “*Framework*”, 21,2% dos trabalhos apresentam ferramentas prontas para utilizar no processo de criação de microserviços, o que é significativo. Parte delas foca no processo de migração de uma arquitetura monolítica para a arquitetura de microserviços, o que se mostrou ainda um dos maiores desafios dessa área. Acredita-se que com o surgimento de mais *frameworks* possa-se chegar a uma solução para este ponto.

Nas categorias “Artigo filosófico” e “Artigo de opinião” percebe-se os menores índices de ocorrência, com 6,1% e 3,0%, respectivamente. Pela natureza do artigo filosófico apresentar uma nova forma de olhar para algo que já existe, ou uma nova estrutura conceitual, ficam claros os motivos da baixa ocorrência. Como o tema é muito recente, e ainda não possui definições claras, trabalhos com perfil de filosófico são poucos pois a maioria deles apresentam soluções inéditas.

Quanto aos artigos de opinião, que expressam opinião pessoal do autor sobre determinada técnica, a ocorrência é menor ainda. Existem estudos que avaliam técnicas, porém foram implementadas pelos próprios autores e esses se enquadraram em uma pesquisa de avaliação.

6.1.2 Últimas Facetas

A Tabela 6.2 demonstra a distribuição das categorias das facetas “Tipo de desenvolvimento”, “Motivação para adoção de microserviços” e “Fases do processo de software” entre os trabalhos.

Na primeira coluna é exibida a categoria, na segunda coluna a porcentagem de estudos em que aquela categoria se enquadrou, e na terceira o valor total de estudos.

| Categoria | Porcentagem dos estudos | Quantidade de estudos |
|---------------------|-------------------------|-----------------------|
| Migração | 33,3 | 11 |
| Novo sistema | 66,7 | 22 |
| Escalabilidade | 51,5 | 17 |
| Evolução | 48,5 | 16 |
| Manutenção | 39,4 | 13 |
| Desempenho | 15,2 | 5 |
| Confiabilidade | 18,2 | 6 |
| Princípios reativos | 42,4 | 14 |
| Requisitos | 48,5 | 16 |
| Arquitetura | 87,9 | 29 |
| Implementação | 51,5 | 17 |
| Testes | 24,2 | 8 |
| Implantação | 27,3 | 9 |

Tabela 6.2: Distribuição das categorias das três últimas facetas entre os trabalhos

O primeiro detalhe a ser abordado sobre a Tabela 6.2 é a distribuição dos estudos entre as categorias “Migração” e “Novo sistema”. Aproximadamente 33% dos artigos descrevem formas de obter microserviços a partir de uma aplicação já existente, realizando a migração da arquitetura monolítica para a arquitetura de microserviços. O restante dos trabalhos, quase 67%, apresentam abordagens para criação de microserviços a partir do desenvolvimento de sistemas novos. Nos trabalhos são destacados os desafios de realizar a migração citada e, pela observação dos dados, percebe-se uma preferência por desenvolver toda a arquitetura desde o início para se obter microserviços.

Percebe-se a maior ocorrência na categoria “Arquitetura”, que se enquadrou em 87,9% dos estudos. Naturalmente essa fase do processo de software apresenta-se em uma quantidade maior de trabalhos, visto que o assunto principal deles é sobre a arquitetura de microserviços. Os artigos em que não se apresentou essa categoria são estudos sobre a organização estrutural de outros domínios baseadas na estrutura da arquitetura de microserviços.

Quanto às categorias da faceta “Motivação para adoção de microserviços” nota-se que “Escalabilidade”, “Evolução” e “Princípios reativos” tem as maiores ocorrências, com 51,5%, 48,5% e 42,4%, respectivamente. Problemas nessas áreas são comuns nas arquiteturas monolíticas, e se mostram como a principal motivação para adoção da arquitetura de microserviços.

Destaque também para a categoria “Manutenção” com 39,4% de ocorrência. Essas informações demonstram o porquê do interesse na área e reforçam o potencial dessa arquitetura em resolver problemas comuns encontrados com as estruturas utilizadas atualmente.

A Figura 6.1 demonstra o cruzamento entre as categorias das facetas “Tipo de desenvolvimento” e “Motivação para adoção de microserviços”. Nela são apresentados dados que quantificam estudos que possuem as duas categorias da interseção, por exemplo, a partir da visualização da imagem é possível perceber que 11 estudos possuem as categorias “Escalabilidade” e “Novo sistema” simultaneamente.

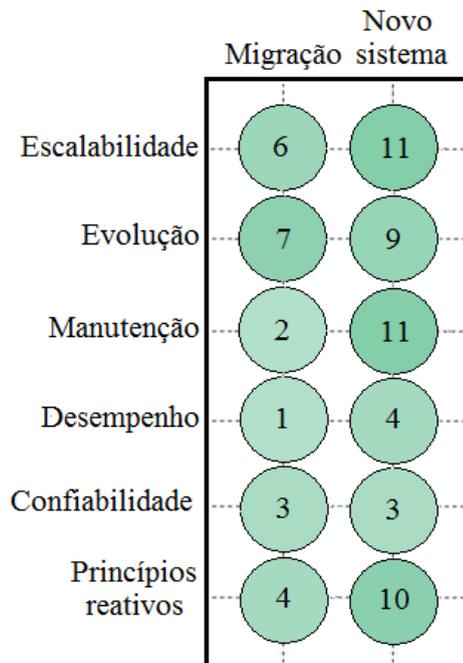


Figura 6.1: Primeiro cruzamento entre categorias das facetas

A partir do comparativo entre motivações para adoção de microserviços e a abordagem apresentada ser uma migração ou um sistema novo, apresentado na Figura 6.1, percebe-se uma tendência a criar uma nova solução ao invés de realizar migração. Dentre os trabalhos que procuram solucionar problemas de escalabilidade, evolução e manutenção nota-se uma preferência por criar um sistema novo, e não por migrar a antiga estrutura para arquitetura de microserviços. Apesar de haver o interesse em melhora de desempenho e confiabilidade, não são esses os principais motivos que levam a escolha do estilo arquitetural.

A Figura 6.2 exibe os dados obtidos a partir do cruzamento das categorias das facetas “Tipo de desenvolvimento” e “Fases do processo de software”. Por exemplo, na imagem percebe-se que 3 artigos possuem as categorias “Testes” e “Novo sistema” simultaneamente.

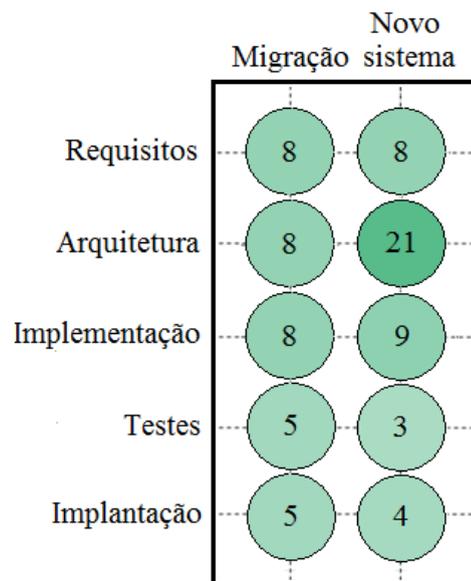


Figura 6.2: Segundo cruzamento entre categorias das facetas

Nessa figura nota-se uma distribuição equilibrada entre a preferência por migração ou desenvolvimento de um novo sistema, quando cruza-se com as fases do processo de software envolvidas nas abordagens. Ou seja, tanto para realizar migração quanto para criar um novo sistema passa-se por todas as fases apresentadas na faceta “Fases do processo de software”.

Destaque para as categorias “Arquitetura” e “Novo sistema” que juntas apareceram em 21 estudos, e que demonstra um foco maior na fase de arquitetura quando desenvolvendo um novo sistema. E também destaca-se a maior quantidade de ocorrências nas fases iniciais do processo de desenvolvimento de software, notando-se uma baixa nas fases de testes e implantação. Isso demonstra como a arquitetura de microserviços ainda está em processo de “amadurecimento” da sua estrutura.

A Figura 6.3 apresenta a quantidade de trabalhos que estão presentes simultaneamente nas categorias das facetas “Motivação para adoção de microserviços” e “Fases do processo de software”. Por exemplo, observa-se que nenhum trabalho se enquadrou nas categorias “Desempe-

no” e “Testes” conjuntamente.

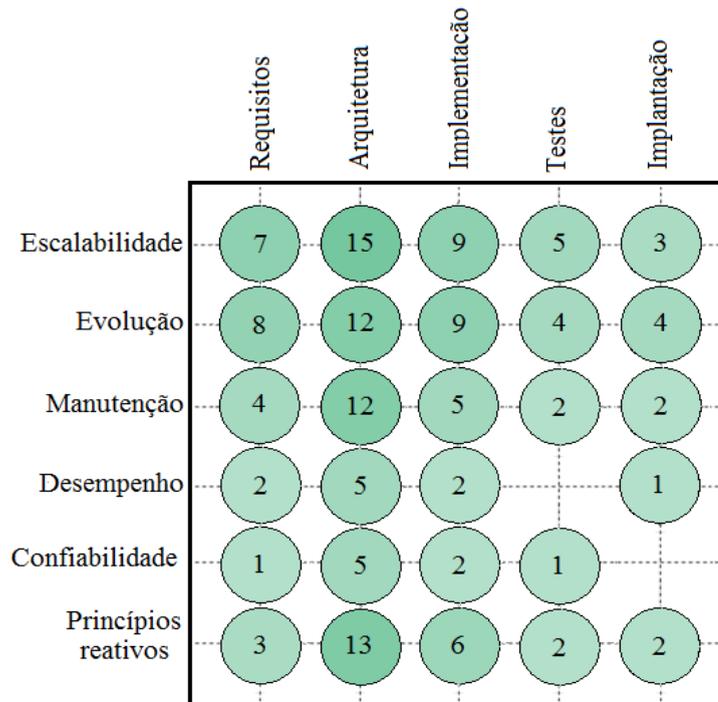


Figura 6.3: Terceiro cruzamento entre categorias das facetas

Nessa figura também percebe-se uma concentração dos valores maiores na parte superior esquerda, com exceção do cruzamento entre as categorias “Princípios reativos” e “Arquitetura” que com 13 ocorrências é a segunda maior interseção. Esse cruzamento com grande quantidade de ocorrências é natural, visto que os princípios reativos são característica da arquitetura de micros serviços.

A concentração citada acima ocorre justamente nas etapas iniciais das fases de desenvolvimento de software e nos principais motivadores para escolha da arquitetura, que são categorias que se enquadram em uma grande quantidade de estudos. Isso demonstra em qual momento das fases de desenvolvimento busca-se contemplar tais atributos, reforçando a importância do estilo arquitetural para obtenção deles, devido a maior quantidade de interseções ocorrer na fase de arquitetura.

6.2 Publicações por Ano

Na Figura 6.4 é possível observar a distribuição dos artigos entre os anos apresentados. Nota-se que não entraram, após a nossa filtragem, estudos do ano de 2014, que é o ano em que começou-se a falar sobre microserviços. Acredita-se que isso ocorre pois ainda não existiam estudos suficientes para serem usados como referência para criação de abordagens para desenvolvimento de microserviços, pois a partir do ano de 2015 são apresentados artigos sobre o assunto de interesse. No nosso estudo foram encontrados 3 trabalhos relativos à este ano.

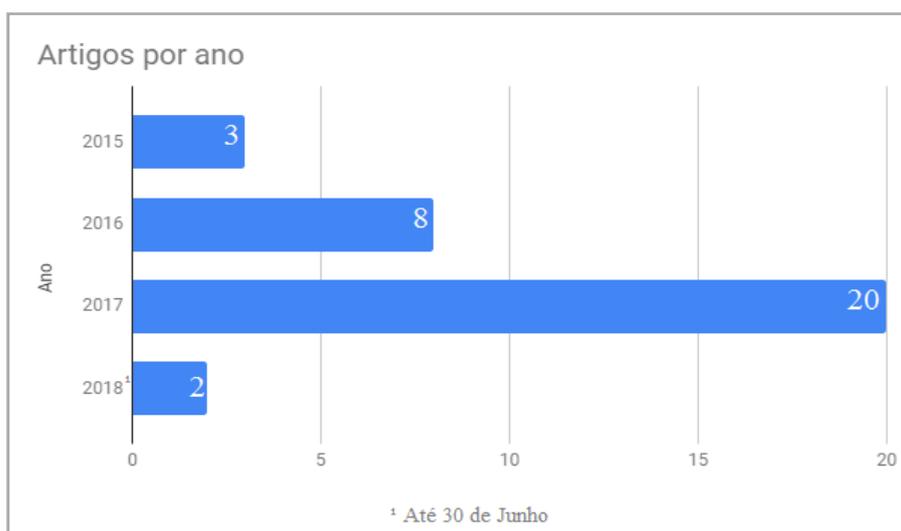


Figura 6.4: Distribuição dos artigos entre os anos.

No ano de 2016 foram selecionados, pelos critérios propostos neste trabalho, 8 artigos sobre abordagens para criação de microserviços, mais que o dobro que o ano anterior. Isso demonstra o aumento do interesse em definir formas para utilização dessa abordagem já neste período.

Já no ano de 2017 foram 20 estudos selecionados relativos ao assunto abordado neste trabalho. Novamente a taxa de aumento da quantidade de artigos foi mais que o dobro que o ano anterior. Percebe-se um aumento de aproximadamente 667% na quantidade de estudos referentes à este assunto desde 2015.

Ainda sobre a Figura 6.4, ao observar a seleção de apenas 2 estudos no ano de 2018 inicialmente interpretou-se uma possível diminuição do interesse no assunto. Porém, importante ressaltar que nos critérios de inclusão e exclusão foi determinado que apenas estudos publica-

dos e disponibilizados nas bases de dados até 30 de Junho de 2018 seriam selecionados para análise, assim limitando a quantidade de estudos possíveis de serem selecionados neste ano. Para esclarecimento desse ponto foi necessário realizar uma nova pesquisa nas bases de dados, utilizando as mesmas métricas apresentadas anteriormente, para identificação de quantos dos artigos publicados após o período determinado pelo critério de exclusão seriam considerados na contagem de 2018.

Nesta nova pesquisa observou-se que entre os meses de Julho e Outubro de 2018 houve a publicação de 6 novos estudos que passariam para o processo de análise, por se enquadrarem em todos os critérios propostos. Mesmo considerando-se estes artigos, o resultado de 8 estudos em 2018 ainda representa uma diminuição na quantidade de artigos com relação ao ano anterior.

Apesar da queda na quantidade de estudos sobre abordagens para criação de microserviços, percebe-se entre os 702 artigos que passaram pela filtragem estudos apresentando soluções para problemas encontrados nas camadas mais internas da arquitetura de microserviços. Trabalhos apresentando melhores formas de gerenciar os microserviços, ou fazendo análises de performance da arquitetura buscando otimização ou solução de problemas também são notados.

6.3 Distribuição entre as Bases de Dados

Na Figura 6.5 é demonstrada a distribuição numérica dos artigos entre as 3 bases de dados digitais selecionadas para pesquisa. Essa representação é de antes da aplicação de qualquer critério que reduzisse a quantidade total de estudos. A parte azul, que representa a base de dados IEEEExplore, demonstra a maior parcela do gráfico, o que quer dizer que essa base tem a disposição uma vasta quantidade de artigos sobre o assunto, possuindo mais da metade dos estudos a serem averiguados neste trabalho, 50,7% para ser exato, contra 20,1% da ScienceDirect e 29,2% da ACM.

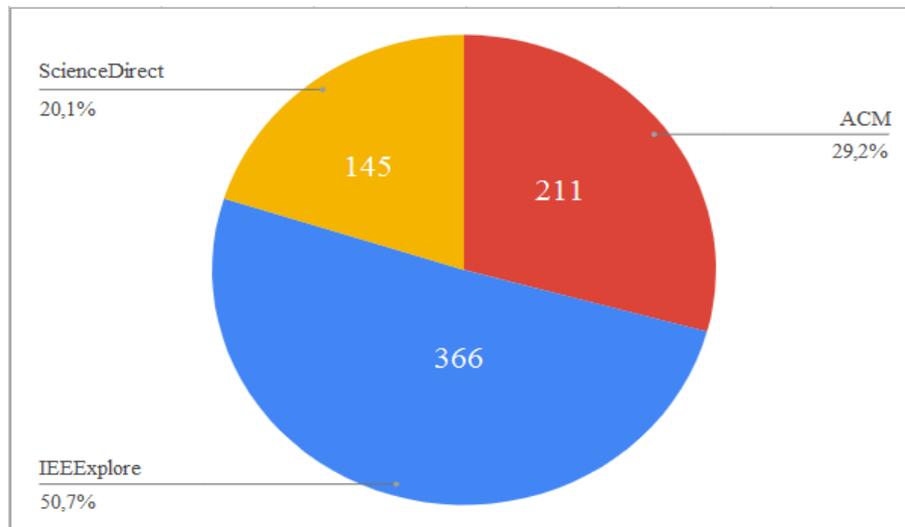


Figura 6.5: Distribuição dos artigos entre as bases de dados antes da filtragem.

Já a Figura 6.6 demonstra como os artigos ficaram distribuídos entre as bases de dados após a aplicação de todos os critérios de inclusão e exclusão deste trabalho. A listagem final de estudos relevantes à pesquisa possui a grande maioria dos artigos obtidos a partir da base de dados IEEEExplore, com 69,7% deles. Na ACM foram obtidos 24,2% dos trabalhos presentes na listagem final, e na ScienceDirect apenas 6,1%.

A partir dos dados apresentados na Figura 6.6 percebe-se a relevância da base de dados IEEEExplore para obtenção de estudos a respeito de abordagens para criação de microserviços. Acredita-se que isso ocorre por conta da grande quantidade de eventos e periódicos associados à IEEE. Importante notar o baixo aproveitamento dos artigos da ScienceDirect para este tema, e o bom resultado obtido da base de dados ACM, que não atingiu um patamar excelente como a IEEEExplore, porém trouxe uma contribuição significativa no resultado deste trabalho.

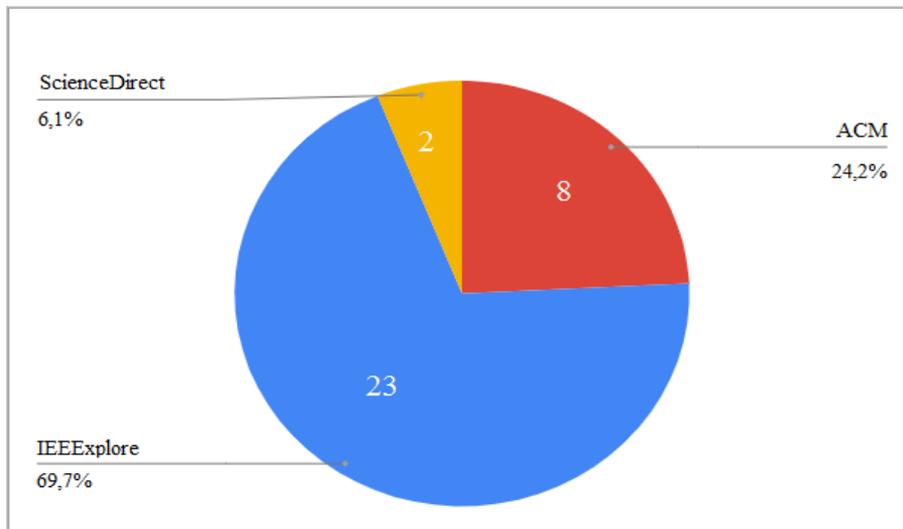


Figura 6.6: Distribuição dos artigos entre as bases de dados após filtragem.

Capítulo 7

Considerações Finais

Este documento apresenta um mapeamento sistemático, buscando responder quais são as abordagens para criação de microserviços através da pergunta de pesquisa. Para realização de tal mapeamento, foi proposto um planejamento de passos a serem seguidos e de métricas que definiriam o perfil dos estudos a serem analisados.

Durante o planejamento, palavras-chave foram definidas, derivadas da pergunta de pesquisa, critérios de inclusão e exclusão que serviriam como filtro para se chegar a uma listagem final contendo apenas trabalhos relevantes ao assunto de interesse também foram planejados. Bases de dados onde seriam realizadas as pesquisas, e se obteria acesso aos estudos foram escolhidas, ainda durante o planejamento, e *strings* de busca que continham uma sintaxe específica para cada base de dados foram montadas. Métodos de classificação, onde foram apresentadas facetas e categorias que auxiliariam na análise, foram definidos dentro do planejamento.

Para validação de todas as métricas que serviriam de guia para obtenção de um conjunto de estudos de interesse dessa pesquisa, realizou-se a execução de um projeto piloto. Nele o planejamento foi colocado em prática e à prova, de forma amostral, para se ter certeza que a maneira como foi esboçado o planejamento seria ideal para obtenção dos resultados. Neste momento foi possível ajustar as métricas do planejamento e se ter certeza que seria haveriam estudos relevantes para a pesquisa.

Finalmente, foi possível executar efetivamente o planejamento proposto e iniciar o processo de filtragem da grande quantidade de estudos oferecidas pelas bases de dados digitais. Após averiguação de mais de 700 trabalhos, publicados em eventos, revistas, entre outros, chegou-se a listagem final de artigos.

A partir da listagem final iniciou-se a análise dos estudos individualmente, e a distribuição

das categorias definidas anteriormente entre eles. Isso gerou dados que foram posteriormente analisados, gerando percepções importantes que servem como contribuição científica àqueles que tem interesse em aprofundar conhecimentos sobre esse mesmo assunto, ou temas semelhantes.

7.1 Resultados

O trabalho traz como contribuição uma análise de artigos que podem servir como ponto de partida para uma vasta variedade de outros estudos. Os resultados demonstram que os critérios e métricas definidos inicialmente serviram ao seu propósito de selecionar estudos relevantes à esta pesquisa.

Uma variedade de abordagens, técnicas e *frameworks* são contempladas dentre os trabalhos selecionados a partir do mapeamento realizado. E estas devem servir de guia para novas pesquisas, ou até mesmo para aplicação em situações reais compatíveis.

Portanto, a pergunta de pesquisa é repondida na forma de listagem de artigos relevantes sobre o assunto e ainda, a análise e categorização realizados servem para instruir o leitor interessado sobre onde encontrar informações relevantes e confiáveis sobre abordagens para criação de microserviços.

7.2 Conclusões

Os dados obtidos a partir das análises demonstraram um verdadeiro interesse da comunidade científica, e até do mercado da indústria de software, na arquitetura de microserviços e nos seus desafios de criação e manutenção.

Um tema recente o qual uma grande quantidade de estudos foram publicados buscando definir ou solucionar problemas referentes aos microserviços.

Percebe-se que ainda há muito “território” a ser explorado nesta área e que existem diversas propostas diferentes, mostrando que, pelo menos por enquanto, a arquitetura de microserviços não está 100% consolidada. Porém demonstra também ter potencial para não só resolver problemas existentes, como abrir novas possibilidades dentro do conceito de computação na nuvem.

7.3 Trabalhos Futuros

Como trabalhos futuros sugere-se análise dos modelos existentes de arquiteturas de micro-serviço, buscando identificar um padrão e definir uma estrutura genérica da arquitetura.

Outra sugestão seria realizar uma análise qualitativa das abordagens existentes, buscando avaliar quais as melhores práticas, dentre as que já são bem definidas.

A implementação de um sistema baseado em arquitetura de microserviços, para servir de estudo de caso, e avaliação empírica da eficiência da técnica também é uma sugestão para trabalhos futuros.

Referências Bibliográficas

AFANASEV, M. Y. et al. An application of microservices architecture pattern to create a modular computer numerical control system. In: *20th Conference of Open Innovations Association (FRUCT)*. Russia: [s.n.], 2017. p. 10–19.

ALI, S. et al. Microservices model in woo based iot platform for depressive disorder assistance. In: *International Conference on Information and Communication Technology Convergence (ICTC)*. South Korea: [s.n.], 2017. p. 864–866.

ALSHUQAYRAN, N.; ALI, N.; EVANS, R. A systematic mapping study in microservice architecture. In: *2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA)*. China: [s.n.], 2016. p. 44–51.

BAKSHI, K. Microservices-based software architecture and approaches. In: *IEEE Aerospace Conference*. USA: [s.n.], 2017. p. 1–8.

BERKUNSKYI, Y. et al. Using microservices in educational applications of it-company. In: *IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*. Ukraine: [s.n.], 2017. p. 1208–1211.

BLOCH, H. et al. A microservice-based architecture approach for the automation of modular process plants. In: *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. Cyprus: [s.n.], 2017. p. 1–8.

BONÉR, J. *Reactive Microservices Architecture: Design Principles for Distributed Systems*. 1. ed. EUA: O'Reilly Media, 2016.

BONÉR, J. et al. *The Reactive Manifesto*. 2014. Consultado na Internet: <http://www.reactivemanifesto.org/>, 21/08/2017.

BRILHANTE, J.; COSTA, R.; MARITAN, T. Asynchronous queue based approach for building reactive microservices. In: *WebMedia '17: Proceedings of the 23rd Brazillian Symposium on Multimedia and the Web*. Brasil: [s.n.], 2017. p. 373–380.

BUDGEN, D. et al. *Using Mapping Studies in Software Engineering*. 2017. Consultado na Internet: <http://www.ppig.org/papers/20th-budgen.pdf>, em: 15/05/2017.

BUTZIN, B.; GOLATOWSKI, F.; TIMMERMANN, D. Microservices approach for the internet of things. In: *IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*. Germany: [s.n.], 2016. p. 1–6.

- CALÇADO, P. *Building Products at SoundCloud—Part III: Microservices in Scala and Finagle*. 2014. Consultado na Internet: <https://developers.soundcloud.com/blog/building-products-at-soundcloud-part-3-microservices-in-scala-and-finagle>, 21/08/2017.
- CAMPEANU, G. A mapping study on microservice architectures of internet of things and cloud computing solutions. In: *7th Mediterranean Conference on Embedded Computing (MECO)*. Montenegro: [s.n.], 2018. p. 1–4.
- CHEN, R.; LI, S.; LI, Z. From monolith to microservices: A dataflow-driven approach. In: *24th Asia-Pacific Software Engineering Conference (APSEC)*. China: [s.n.], 2017. p. 466–475.
- CHERRADI, G. et al. Real-time hazmat environmental information system: A micro-service based architecture. In: *The 7th International Symposium on Frontiers in Ambient and Mobile Systems (FAMS2017)*. France: [s.n.], 2017. p. 982–987.
- DRAGONI, N. et al. *Microservices: yesterday, today, and tomorrow*. 2017. Consultado na Internet: <https://arxiv.org/pdf/1606.04036.pdf>, 20/07/2017.
- FAN, C.-Y.; MA, S.-P. Migrating monolithic mobile application to microservice architecture: An experiment report. In: *IEEE International Conference on AI & Mobile Services (AIMS)*. USA: [s.n.], 2017. p. 109–112.
- GADEA, C. et al. A microservices architecture for collaborative document editing enhanced with face recognition. In: *IEEE 11th International Symposium on Applied Computational Intelligence and Informatics (SACI)*. Romania: [s.n.], 2016. p. 441–446.
- GADEA, C. et al. A reference architecture for real-time microservice api consumption. In: *Proceedings of the 3rd Workshop on CrossCloud Infrastructures & Platforms*. United Kingdom: [s.n.], 2016. p. Artigo 2.
- GOLEMBESKI, D. et al. Pipelinex: a feature animation pipeline on microservices. In: *DigiPro '17: Proceedings of the ACM SIGGRAPH Digital Production Symposium*. USA: [s.n.], 2017. p. Artigo 1.
- HASELBÖCK, S.; WEINREICH, R.; BUCHGEHER, G. Decision guidance models for microservices: service discovery and fault tolerance. In: *ECBS '17: Proceedings of the Fifth European Conference on the Engineering of Computer-Based Systems*. Cyprus: [s.n.], 2017. p. Article 4.
- IDOUGH, D.; ABDELOUHAB, K. A.; KOLSKI, C. Towards a microservices development approach for the crisis management field in developing countries. In: *4th International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*. Germany: [s.n.], 2017. p. 1–6.
- KAKIVAYA, G. et al. Service fabric: a distributed platform for building microservices in the cloud. In: *EuroSys '18: Proceedings of the Thirteenth EuroSys Conference*. Portugal: [s.n.], 2018. p. Article 33.

- KECSKEMETI, G.; MAROSI, A. C.; KERTESZ, A. The entice approach to decompose monolithic services into microservices. In: *International Conference on High Performance Computing & Simulation (HPCS)*. Austria: [s.n.], 2016. p. 591–596.
- LE, V. D. et al. Microservice-based architecture for the nrdc. In: *IEEE 13th International Conference on Industrial Informatics (INDIN)*. Reino Unido: [s.n.], 2015. p. 1659–1664.
- LEWIS, J.; FOWLER, M. *Microservices*. 2014. Consultado na Internet: <https://martinfowler.com/articles/microservices.html>, 20/07/2017.
- LIN, J.; LIN, L. C.; HUANG, S. Migrating web applications to clouds with microservice architectures. In: *International Conference on Applied System Innovation (ICASI)*. Japan: [s.n.], 2016. p. 1–4.
- MALAVALLI, D.; SATHAPPAN, S. Scalable microservice based architecture for enabling dmtf profiles. In: *11th International Conference on Network and Service Management (CNSM)*. Spain: [s.n.], 2015. p. 428–432.
- PAHL, C.; JAMSHIDI, P. Microservices: A systematic mapping study. In: *6th International Conference on Cloud Computing and Services Science (CLOSER)*. Itália: [s.n.], 2016. p. 137–146.
- PETERSEN, K. et al. Systematic mapping studies in software engineering. In: *12th international conference on Evaluation and Assessment in Software Engineering (EASE)*. Itália: [s.n.], 2008. p. 1–10.
- RICHTER, D. et al. Highly-available applications on unreliable infrastructure: Microservice architectures in practice. In: *IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*. Czech Republic: [s.n.], 2017. p. 130–137.
- SALVADORI, I. et al. Publishing linked data through semantic microservices composition. In: *iiWAS '16: Proceedings of the 18th International Conference on Information Integration and Web-based Applications and Services*. Singapore: [s.n.], 2016. p. 443–452.
- SARITA; SEBASTIAN, S. Transform monolith into microservices using docker. In: *International Conference on Computing, Communication, Control and Automation (ICCUBEA)*. India: [s.n.], 2017. p. 1–5.
- SCARBOROUGH, W.; ARNOLD, C.; DAHAN, M. Case study: Microservice evolution and software lifecycle of the xsede user portal api. In: *XSEDE16: Proceedings of the XSEDE16 Conference on Diversity, Big Data, and Science at Scale*. USA: [s.n.], 2016. p. Article 47.
- SINGH, V.; PEDDOJU, S. K. Container-based microservice architecture for cloud applications. In: *International Conference on Computing, Communication and Automation (ICCCA)*. India: [s.n.], 2017. p. 847–852.
- SRIKAEW, P.; KIM, I. A microservice development for document management system. In: *4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*. Indonesia: [s.n.], 2017. p. 1–4.

- SUN, L.; LI, Y.; MEMON, R. A. An open iot framework based on microservices architecture. *China Communications*, v. 2, n. 14, p. 154–162, Fev 2017.
- SURYOTRISONGKO, H.; JAYANTO, D. P.; TIAHYANTO, A. Design and development of backend application for public complaint systems using microservice spring boot. In: *4th Information Systems International Conference 2017, ISICO 2017*. Indonesia: [s.n.], 2017. p. 736–743.
- THRAMBOULIDIS, K.; VACHTSEVANOU, D. C.; SOLANOS, A. Cyber-physical microservices: An iot-based framework for manufacturing systems. In: *IEEE Industrial Cyber-Physical Systems (ICPS)*. Russia: [s.n.], 2018. p. 232–239.
- THÖNES, J. Microservices. *IEEE SOFTWARE*, v. 32, n. 1, p. 116–116, Janeiro/Feveiro 2015.
- TOFFETTI, G. et al. An architecture for self-managing microservices. In: *AIMC '15: Proceedings of the 1st International Workshop on Automated Incident Management in Cloud*. France: [s.n.], 2015. p. 19–24.
- TONSE, S. *MicroServices at Netflix - challenges of scale*. 2014. Consultado na Internet: <https://pt.slideshare.net/stonse/microservices-at-netflix>, 21/08/2017.
- WIERINGA, R. et al. Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. v. 11, p. 102–107, 03 2006.
- YU, Y.; SILVEIRA, H.; SUNDARAM, M. A microservice based reference architecture model in the context of enterprise architecture. In: *IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*. China: [s.n.], 2016. p. 1856–1860.
- YUGOPUSPITO, P.; PANDUWINATA, F.; SUTRISNO, S. Microservices architecture: Case on the migration of reservation-based parking system. In: *IEEE 17th International Conference on Communication Technology (ICCT)*. China: [s.n.], 2017. p. 1827–1831.
- ZHENG, X. et al. Smartvm: A multi-layer microservice-based platform for deploying saas. In: *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*. China: [s.n.], 2017. p. 470–474.