

CESAR SCHOOL
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

PAULO FERNANDO DE BARROS CAVALCANTI

**ANÁLISE DE PONTOS CRÍTICOS DA MIGRAÇÃO DE
UM SISTEMA MONOLÍTICO PARA UMA ARQUITETURA
DE MICROSERVIÇOS UTILIZANDO A TÉCNICA DO
ESTRANGULAMENTO**

RECIFE
2023

PAULO FERNANDO DE BARROS CAVALCANTI

**ANÁLISE DE PONTOS CRÍTICOS DA MIGRAÇÃO DE
UM SISTEMA MONOLÍTICO PARA UMA ARQUITETURA
DE MICROSERVIÇOS UTILIZANDO A TÉCNICA DO
ESTRANGULAMENTO**

Trabalho de Conclusão de Curso
apresentado ao Curso de Graduação do
Centro de Estudos e Sistemas Educacionais
do Recife – CESAR School, como requisito
para a obtenção do título de Bacharel em
Ciência da Computação.

Orientação: Prof. Carlos Diego Cavalcanti
Pereira

**RECIFE
2023**

Análise de Pontos Críticos da Migração de um Sistema Monolítico para uma Arquitetura de Microsserviços Utilizando a Técnica do Estrangulamento

Resumo: A contribuição deste estudo é apresentar um panorama atual da análise dos pontos críticos percebidos mais comuns durante o processo de migração de um sistema monolítico para uma arquitetura de microsserviços, utilizando a técnica do estrangulamento. Baseando-se em uma revisão da literatura e na experiência de pesquisa relatada pelos autores, registra-se um roteiro embrionário de etapas para os interessados na estratégia de estudo do caso. Em adição, espera-se que este artigo possa contribuir com pesquisadores iniciantes que almejam aprofundar os conhecimentos relativos ao tema de pesquisa e também com profissionais da área que buscam um conhecimento prévio de potenciais pontos sensíveis e de alta dificuldade durante a execução de uma migração em seu ambiente profissional.

Palavras-chave: Migração de Arquitetura; Técnica do Estrangulamento; Monolito; Microsserviços; Sistema Distribuído; Revisão Bibliográfica; Pontos Críticos; Principais Desafios; Maiores Dificuldades;

Abstract: The contribution of this article is to present a current overview of the most common perceived critical points analysis during the process of migrating from a monolithic system to a microservices architecture using the strangler fig technique. Based on a literature review and the authors' research experience, an embryonic roadmap of steps is recorded for those interested in the case study strategy. Additionally, it is expected that this article can contribute to novice researchers aiming to deepen their knowledge in the research topic, as well as professionals in the field seeking prior understanding of potential sensitive and

challenging points during the execution of a migration in their professional environment.

Keywords: Architecture Migration; Strangler Fig Pattern; Monolith; Microservices; Distributed Systems; Literature Review; Critical Points; Key Challenges; Main Issues;

1. Introdução

1.1 Motivação e Contexto

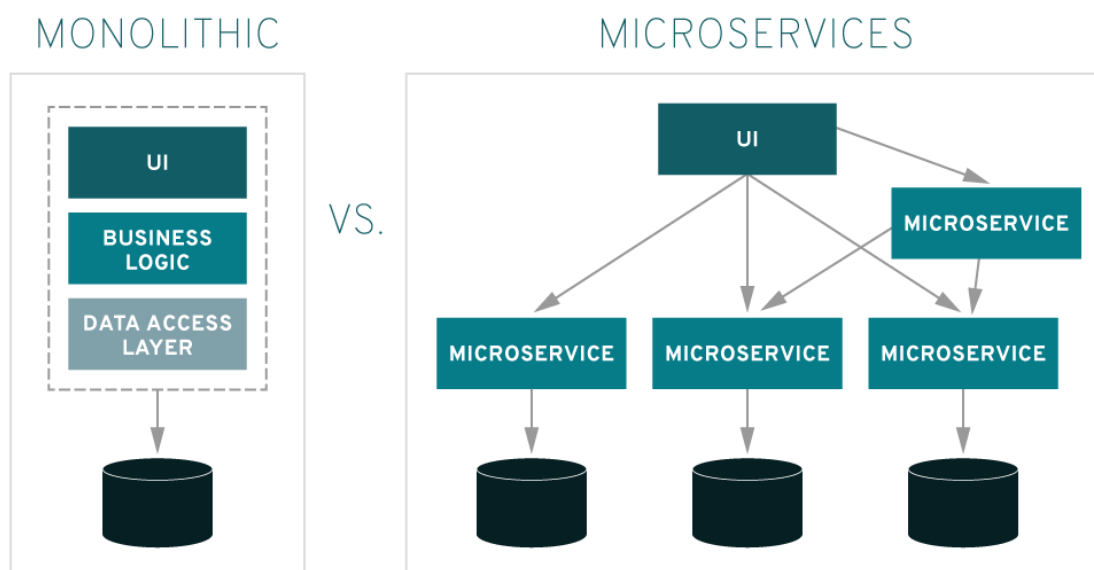
A arquitetura de software de um programa ou sistema computacional é a estrutura do sistema, que abrange os componentes do software, as propriedades externamente visíveis desses componentes e as relações entre eles (BASS et al., 2003).

Nos últimos anos, grandes corporações como Netflix, Amazon, Spotify, LinkedIn, IBM, Uber, eBay e Paypal vêm migrando de arquiteturas de software legadas para a arquitetura de microsserviços. Esse movimento acontece porque, a migração entre um monolito e uma arquitetura de microsserviços representa uma vantagem competitiva de mercado se analisado as vantagens adquiridas. Para Taibi, o que leva um desenvolvedor de software a escolher uma arquitetura em detrimento a outra são os resultados de experiências anteriores ou os benefícios percebidos da nova arquitetura (TAIBI et al., 2017). Esse impacto que a arquitetura escolhida exerce se estende a uma ampla gama de fatores como ciclo de vida do software, os atributos de qualidade do software, quão rápido o mesmo deve ser para atender determinada demanda ou expectativa do usuário e até mesmo o quão fácil é modificá-lo e escalá-lo.

Baseado na definição de um sistema monolítico como a implantação em conjunto de todas as funcionalidades de um sistema, a escolha de migração para um sistema distribuído como o de microsserviços se dá pois o estilo arquitetural deste modelo se baseia fortemente em uma abordagem computacional orientada a serviços (DRAGONI et al., 2017). Seu principal objetivo é a construção e gestão eficiente de sistemas de software complexos. Tais vantagens estão

atreladas primordialmente à busca do aumento da qualidade do serviço, a agilidade para *deploy* de máquinas e a redução de custos.

Figura 1: Representação arquitetural monolítica e de microsserviços.



Fonte: Medium. Acesso em 15 de out. de 2022.

No passado, uma quantidade significativa de sistemas com arquiteturas de software legadas começaram os processos de migração para o ambiente de *cloud* mas não adaptaram suas respectivas arquiteturas para se adequarem às novas infraestruturas de armazenamento na nuvem. Este tipo de abordagem não é vantajoso em nenhum aspecto se comparado ao resultado de um processo de migração arquitetural bem sucedido entre os sistemas legados.

Em geral, grandes organizações (como as citadas anteriormente) que começam o processo de migração para microsserviços possuem um ponto de partida extremamente complexo, com grandes aplicações monolíticas que começam a sinalizar problemas. Normalmente estas dificuldades que começam a surgir estão atreladas à capacidade de realizar mudanças funcionais. Como solução para este cenário, optou-se pela introdução gradual de novos serviços independentes, capazes de substituir frações dos sistemas até que uma parte

predominante da corporação esteja sob a arquitetura de microsserviços. Este processo contínuo e controlado de migração é popularmente conhecido como o estrangulamento monolítico com base no *Strangler pattern* descrito por Fowler (FOWLER, 2004).

Dentro das diferentes técnicas para uma migração efetiva de arquitetura, a mais utilizada é a técnica de estrangulamento (como citado acima). Essa forma de migração se baseia no processo de substituição incremental em etapas das funcionalidades dispostas no repositório do sistema monolítico por um novo serviço distribuído em microsserviços. Uma vez que a nova funcionalidade migrada estiver pronta, o componente da arquitetura legada é "estrangulado", o novo serviço é colocado em uso como padrão e o antigo componente é descontinuado. Qualquer novo desenvolvimento de funcionalidade será implementado dentro dos conformes da estrutura de arquitetura distribuída e eventualmente toda a aplicação será atualizada para o novo modelo.

1.2 Problema de Pesquisa

Uma vez contemplado o cenário disposto acima, apresentamos o problema de pesquisa deste projeto que é: **Quais são os pontos críticos de uma migração de um sistema monolítico para uma arquitetura de microsserviços utilizando a técnica do estrangulamento?** Com os benefícios da identificação de pontos críticos durante uma migração entre as arquiteturas trabalhadas, visamos esclarecer quais características possuem um maior impacto durante o processo de migração. Dessa forma, contribuir com conhecimentos que venham a agregar em um processo futuro de transformação entre os paradigmas arquiteturais da migração de sistemas monolíticos para os de microsserviços.

1.3 Justificativa

Ao contribuir com o mapeamento de pontos críticos que dificultem ou contribuam para a falha de uma migração de arquitetura monolítica para microsserviços, este trabalho irá servir como um grande apoio teórico para que diferentes sistemas sejam atualizados com maior facilidade e com uma maior taxa de sucesso. Diversos autores (DRAGONI et al., 2017; NEWMAN, 2015) constataam que os métodos que facilitam a migração de sistemas monolíticos para microsserviços são escassos. Logo, a proposta deste trabalho se torna necessária e justificável como forma de contribuição para um futuro otimizado dos processos de migração.

1.4 Objetivos

1.4.1 Objetivo geral

O presente trabalho tem como principal objetivo identificar características críticas comuns a um processo de migração bem-sucedido entre as arquiteturas monolíticas e de microsserviços usando a técnica do estrangulamento.

1.4.2 Objetivos específicos

- Executar buscas em bases acadêmicas de estudos que abordam o processo de migração de um monólito para microsserviços;
- OE2: Aprofundar o conhecimento relativo ao estado da arte no tocante ao uso de técnicas de estrangulamento como método de migração;

- OE3: Analisar pontos críticos comuns que impactam o processo de implementação de migrações usando a técnica de estrangulamento.

2. Fundamentação Teórica

Neste capítulo, será descrito de maneira resumida quais assuntos são indispensáveis para o entendimento da análise de pontos críticos no processo de migração arquitetural de um monólito para um microsserviço utilizando, especificamente, o método do estrangulamento (*strangler fig*).

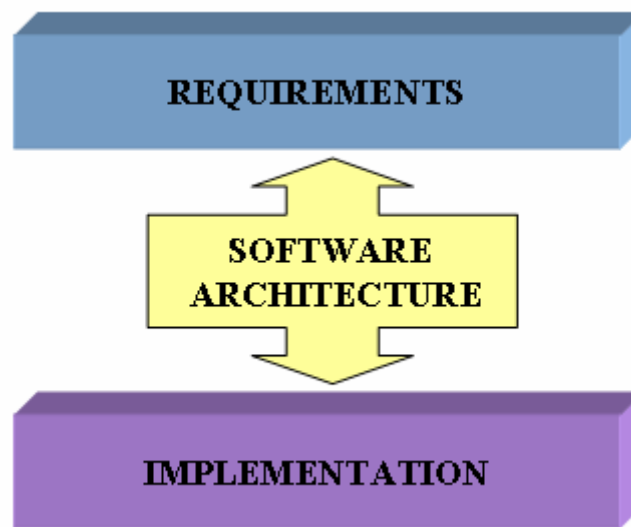
2.1 Arquitetura de Software

Segundo Joseph Ingeno, em seu livro "*Software Architect's Handbook*", a definição padrão de arquitetura de software é o resultado de um esforço em conjunto entre a ISO (*International Organization for Standardization*) e o IEEE (*Institute of Electrical and Electronics Engineers*) como o conjunto de conceitos fundamentais ou propriedades de um sistema em seu ambiente e seus elementos, relações e seus princípios de seu design e evolução (INGENO, 2018).

A padronização traz a tona os seguintes pontos principais:

- A arquitetura de software é uma parte fundamental do sistema do software;
- Um sistema de software pertence a um ambiente, e sua arquitetura de software leve em consideração o ambiente no qual o software irá operar;
- A *architecture description* documenta a própria arquitetura e comunica aos *stakeholders* como essa arquitetura implementada supre com as necessidades do sistema;
- *Architecture views* são criadas a partir da *architecture description*, e toda *view* cobre uma ou mais características que interessam aos *stakeholders*;

Figura 2: Arquitetura de software como uma ponte.



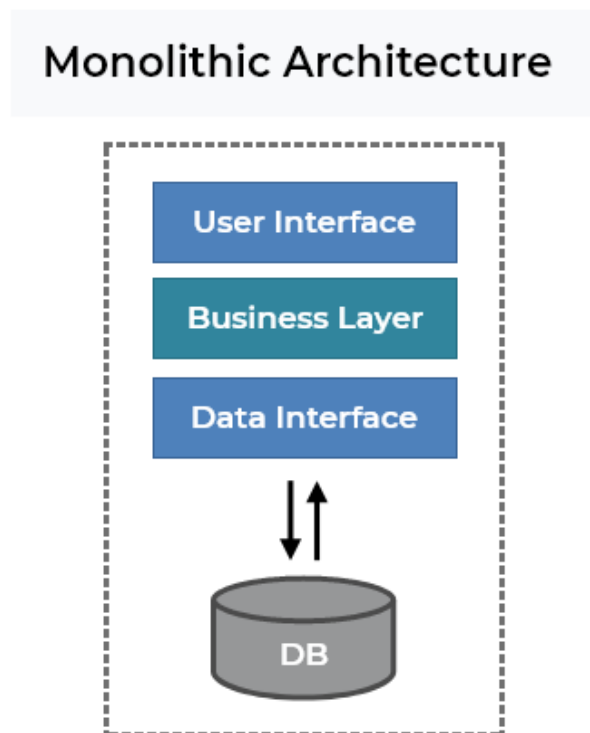
Fonte: ResearchGate. Acesso em 22 de out. de 2022.

A arquitetura de software é a fundação de um sistema de software. Tal qual na engenharia, a fundação tem um profundo efeito na qualidade do produto que será construído a partir de tal base. Devido a isso, uma arquitetura sólida e adaptada para o objetivo final da aplicação possui um enorme impacto no desenvolvimento efetivo e na eventual manutenção do sistema.

2.2 Monólito

Arquitetura monolítica é uma aplicação com uma única base de código que inclui múltiplos serviços. Esses serviços se comunicam com sistemas ou consumidores externos através de diferentes interfaces como serviços *Web*, páginas *HTML* ou *REST API* (AL-DEBAGY et al., 2018).

Figura 3: Arquitetura monolítica.



Fonte: Medium. Acesso em 15 de out. de 2022.

Segundo Chris Richardson em seu estudo "*Pattern: Monolithic Architecture*", algumas vantagens de uma arquitetura monolítica são:

- É uma arquitetura simples de desenvolver - atualmente grande parte das IDEs e das ferramentas de desenvolvimento fornecem suporte para o desenvolvimento de aplicações monolíticas;
- O deploy da aplicação é simples;
- A arquitetura é de fácil escalabilidade - é possível escalar a aplicação ao rodar múltiplas cópias da mesma aplicação atrás de um *load balancer*;

Porém, é importante ressaltar que uma vez que a aplicação e a equipe cresçam substancialmente, algumas complicações surgem, como:

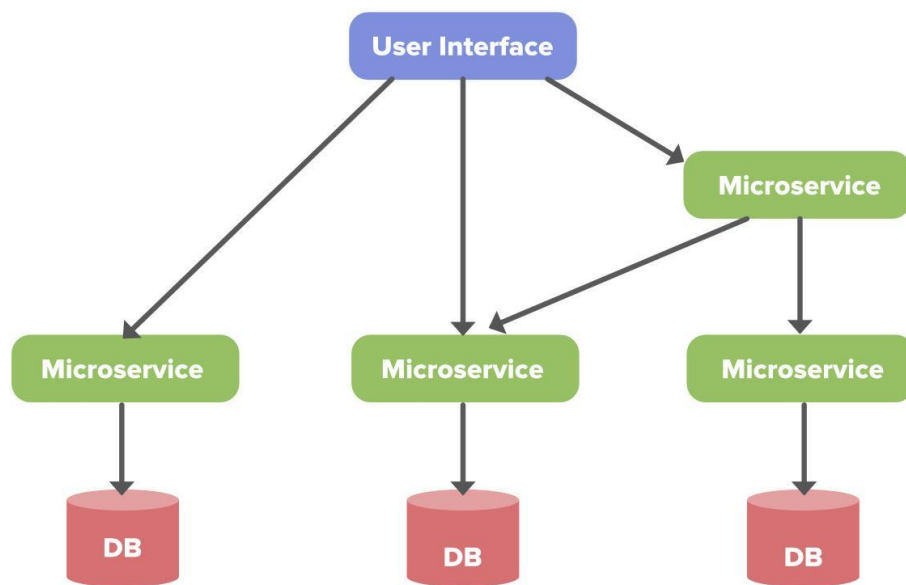
- A grande base de código monolítica acaba por intimidar os desenvolvedores, principalmente novos membros na equipe;
- A aplicação pode ser difícil de compreender e alterar. Como consequência, a velocidade de desenvolvimento pode sofrer com tais características;
- Sobrecarga do container *web* - quanto maior a aplicação, mais demorado é o processo de inicialização do container. O mesmo acontece com o tempo de *deploy* da aplicação;
- O processo de CD (*Continuous Deployment*) é difícil - uma aplicação monolítica de grande porte se torna um obstáculo para *deploys* frequentes;
- Escalar a aplicação pode ser difícil - apesar da escalabilidade do monolito ser um possível ponto positivo, isso se aplica apenas para uma dimensão. Ao mesmo tempo que a aplicação pode escalar para suportar um maior volume de transações através da execução de múltiplas cópias do mesmo código, a arquitetura não consegue acompanhar a progressão do volume de dados. Cada cópia executada em paralelo da aplicação irá acessar o repositório de dados em sua integridade, causando o processo de *cache* menos efetivo e aumentando de forma proporcional o consumo de memória assim como o tráfego de *I/O*;
- Requer um comprometimento de longo prazo com um único *stack* de tecnologias;

2.3 Microsserviços

Microsserviços são um tipo de arquitetura de software onde grandes aplicações são compostas de serviços menores e independentes. De acordo com o artigo publicado por Martin Fowler e James Lewis em 2014, "Em resumo, o estilo arquitetural de microsserviços é uma tática de desenvolvimento de uma única aplicação como um agrupado de pequenos serviços, onde cada um

executa seus próprios processos e gerenciam suas próprias comunicações através de mecanismos leves, normalmente como recursos de API via HTTP" (FOWLER et al., 2014).

Figura 4: Arquitetura de microsserviços.



Fonte: Acervo Lima. Acesso 03 de nov. de 2022.

Este tipo de arquitetura de software, além de ser conhecida como microsserviços, pode ser nomeada de arquitetura orientada a serviços. Existem diversos fatores positivos ao se utilizar de microsserviços, como (RICHARDSON, "Pattern: Microservice Architecture"):

- Um alto grau de flexibilidade e agilidade - serviços específicos podem ser adicionados, editados ou removidos sem que o resto da aplicação sofra alguma perda;
- Alta escalabilidade em duas dimensões - tanto na horizontal quanto na vertical;
- Facilidade de *deploy* assim como de *update* de serviços individuais;

- Um melhor isolamento de falha - no caso da queda de um serviço específico, o restante da aplicação pode continuar funcionando;

No entanto, existem alguns desafios atrelados à implementação de microsserviços também. Alguns deles são:

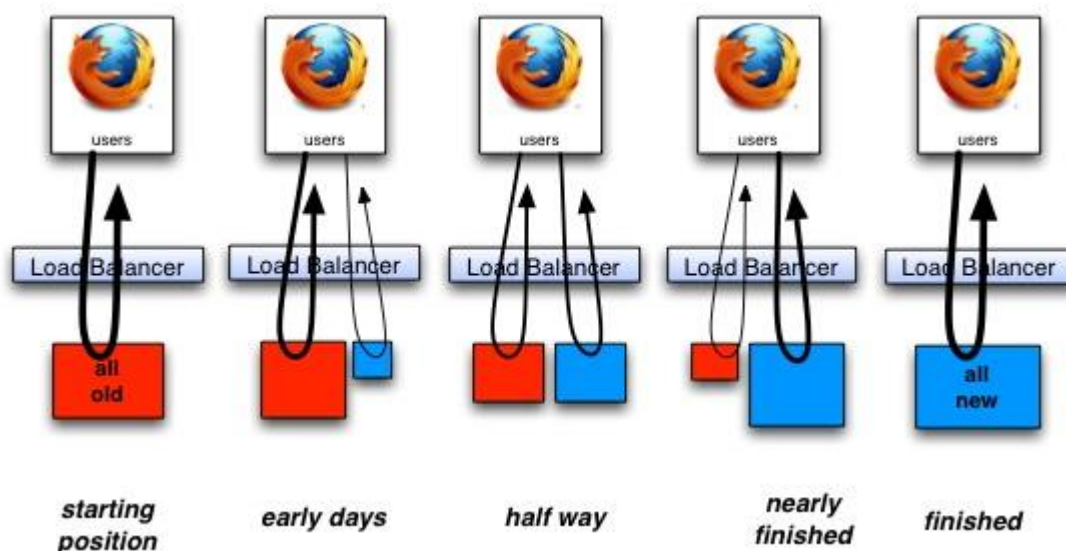
- Pode se tornar difícil a gestão e o monitoramento de um grande número de serviços;
- Pode-se existir *overhead* no processo de comunicação da aplicação - visto que os serviços necessitam se comunicar entre si independentemente;
- Existem cenários onde dificuldades de compatibilidade entre os serviços podem ocorrer;

Apesar dos desafios, os microsserviços vêm se consolidando cada vez mais como uma das arquiteturas mais populares quando o objetivo é a construção de aplicações grandes e complexas.

2.4 Método do Estrangulamento (*Strangler Fig*)

Em 2004, Martin Fowler propôs uma estratégia de modernização de software chamada de método do estrangulamento (*strangler fig application*) (FOWLER, 2004). Segundo ele, a ideia veio a partir de uma planta também chamada de *strangler fig*. O comportamento padrão desta planta é crescer ao redor de uma árvore e escalá-la até o topo no intuito de receber mais luz solar. O resultado de tal ação é a morte gradual da árvore, restando apenas uma vinha com o formato da árvore que acabou de perecer. Este conceito é análogo à forma de reescrita gradual de um sistema (*refactoring*) e expressa a estratégia migratória incremental de alto nível entre uma arquitetura monolítica e uma distribuída em microsserviços (FAN et al., 2017).

Figura 5: Estado da aplicação com a progressão da migração via estrangulamento.



Fonte: RedHat. Acesso 08 de nov. de 2022.

Também de acordo com Martin, a principal razão a se considerar na escolha de uma migração por estrangulamento em detrimento de uma simples, mas demorada reescrita da aplicação inteira é o fator de redução de risco. Ao se usar o método do estrangulamento, o processo de migração de serviços ocorre de forma regular, o que facilita bastante o monitoramento cuidadoso do progresso (FOWLER, 2004). Além da questão do monitoramento facilitado, também é uma característica positiva o tamanho diminuto dos ciclos de *releases* da operação de migração, pois dessa forma, o custo (temporal e financeiro) do desenvolvimento de algumas funcionalidades desnecessárias é reduzido ao mínimo possível.

Segundo Paul Hammant em seu estudo "*Legacy Application Strangulation: Case Studies*", uma das maneiras mais eficazes de progredir com a migração do estado arquitetural da aplicação e em paralelo satisfazer os *stakeholders* e aos membros que custeiam a operação e o seu desenvolvimento, é o ato de mesclar momentos de *releases* focadas unicamente na migração da

arquitetura com etapas de desenvolvimento e implementação de novas funcionalidades (HAMMANT, 2013). Dessa forma, nenhum membro da equipe (envolvido diretamente no processo de desenvolvimento ou não) terá a percepção de que esse momento de migração é extremamente improdutivo e taxativo para a saúde geral do projeto.

3. Metodologia

Este trabalho teve como objetivo a realização de uma revisão bibliográfica, ainda que com atributos de uma revisão sistemática, para analisar pontos críticos e os maiores desafios ou dificuldades percebidos durante uma

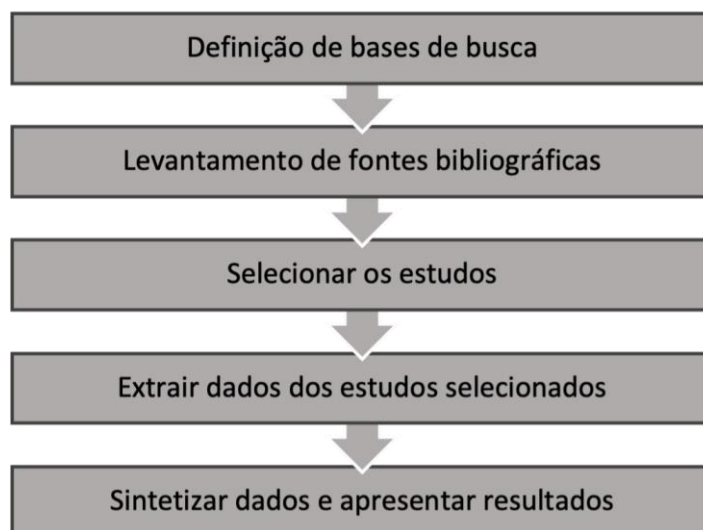
migração partindo de uma arquitetura monolítica para uma distribuída em microsserviços com foco na utilização do método do estrangulamento (*strangler fig*).

A classificação da pesquisa quanto aos seus objetivos se divide em três grandes grupos: qualitativa, descritiva e com o procedimento de uma revisão bibliográfica. O enfoque qualitativo apresenta as seguintes características: o pesquisador é o instrumento-chave, o ambiente é a fonte direta dos dados, não requer o uso de técnicas e métodos estatísticos, têm caráter descritivo, o resultado não é o foco da abordagem, mas sim o processo e seu significado, ou seja, o principal objetivo é a interpretação do fenômeno objeto de estudo [22].

A abordagem qualitativa centra-se na identificação das características de situações, eventos e organizações [23]. Segundo Vergara em [21], a pesquisa descritiva *“expõe características de determinada população ou de determinado fenômeno. Pode também estabelecer correlações entre variáveis e definir sua natureza. Não tem compromisso em explicar os fenômenos que descreve, embora sirva de base para tal explicação”*. A pesquisa descritiva tem o objetivo de descrever as características de um fenômeno, baseado em assuntos teóricos, utilizando livros, artigos e trabalhos acadêmicos.

Tendo em mente a natureza exploratória da pesquisa, as buscas por novas referências bibliográficas (assim como as já obtidas para o pré-projeto) foram realizadas em sites como o "Google Scholar" e o "IEEE Xplore". As buscas da literatura nos sites citados acima gira em torno de um escopo definido por grupos de palavras-chave selecionadas, como: "Strangler Fig Application", "Monolithic Migration", "Migration to Microservices", "Monolithic vs Microservices". Para a execução deste trabalho iremos nos basear nas etapas propostas pela figura 6.

Figura 6: Etapas da metodologia.



Fonte: Elaborado pelo autor.

3.1 Atividades de Pesquisa

3.1.1 - Levantamento de uma base de fontes bibliográficas (artigos/teses)

As fontes de busca escolhidas foram sites comumente usados e/ou conhecidos no meio acadêmico da computação. O quadro 1 contém as bases de dados utilizadas neste trabalho.

Quadro 1: Fontes de busca utilizadas.

Fonte	Endereço
Google Scholar	https://scholar.google.com
IEEE Xplore	https://ieeexplore.ieee.org
ResearchGate	https://www.researchgate.net
ACM Digital Library	https://dl.acm.org

Fonte: Elaborado pelo autor.

Utilizam-se critérios de inclusão e exclusão com o objetivo de minimizar viés e selecionar estudos que possam responder às questões de pesquisa, podendo ser ajustados ao longo da investigação. Os critérios de exclusão (CE) e critérios de inclusão (CI) estão definidos nas tabelas a seguir respectivamente.

Quadro 2: Critérios de exclusão.

Critério	Descrição
CE1	Publicações que não estejam disponíveis para download ou visualização de forma gratuita
CE2	Estudos que foram publicados antes do ano de 2018
CE3	Estudos que não sejam em inglês
CE4	Trabalhos duplicados
CE5	Publicações que não tratam sobre pontos críticos dentro do tema "Migração de monolito para microserviços usando a técnica do estrangulamento"

Fonte: Elaborado pelo autor.

Quadro 3: Critérios de inclusão.

Critério	Descrição
CI1	Publicações disponíveis para visualização ou download
CI2	Estudos que foram publicados no intervalo de 2018 até 2023

CI3	Estudos em inglês
CI4	Trabalhos que tratam sobre pontos críticos dentro do tema "Migração de monolito para microsserviços usando a técnica do estrangulamento"

Fonte: Elaborado pelo autor.

3.1.2 - Procedimento da seleção dos estudos

Os critérios de seleção estabelecidos na seção 3.1.1 foram utilizados para avaliar os estudos obtidos a partir das buscas realizadas, seguindo a estratégia de busca também definida na mesma seção. Esses critérios foram aplicados de forma sequencial e em ordem, visando reduzir a necessidade de retrabalho.

Com o objetivo de garantir que os critérios de inclusão e exclusão fossem avaliados de forma precisa, foi executada a leitura do título, resumo (abstract) e conclusão de todo o material que foram incluídos nos critérios de inclusão, e não foram excluídos durante a aplicação dos critérios de exclusão.

3.1.3 - Estratégia da extração dos dados

A extração dos dados foi realizada em duas etapas distintas. Na primeira etapa, ocorreu a identificação dos estudos, sendo armazenados o título, autor e a base de busca em que o trabalho foi encontrado. Na segunda etapa foram coletadas as informações necessárias para responder à questão de pesquisa.

Em casos onde apenas a leitura dos tópicos propostos inicialmente (título, resumo e conclusão) não foram suficientes para a coleta de todas

as informações necessárias visando a resposta da questão de pesquisa, uma leitura dinâmica dos estudos foi realizada com foco nos tópicos de "resultados do estudo" e "discussão dos resultados".

Para facilitar e agilizar essa análise mais delicada necessária em alguns estudos, palavras-chave e suas respectivas palavras associadas foram definidas para, de forma semelhante ao processo realizado na etapa do procedimento de seleção dos estudos, serem utilizadas como chaves de busca durante esse processo de extração dos dados.

Quadro 4: Palavras-chave e suas respectivas palavras associadas (segunda aplicação).

Palavras-chave	Palavras associadas
Critical point	Crucial point, main difficulty, main issue, key step/challenge
Difficulty	Issue, problem, challenge

Fonte: Elaborado pelo autor.

4. Cronograma

Quadro 5: Cronograma de atividades para o projeto de pesquisa.

Atividade	Fevereiro	Março	Abril	Maio	Junho
Levantamento da base de referências					

Leitura dos materiais levantados					
Aplicação dos filtros e seleção dos estudos					
Extração dos dados					
Preparação da síntese dos resultados					
Escrita do projeto de pesquisa					

Fonte: Elaborado pelo autor.

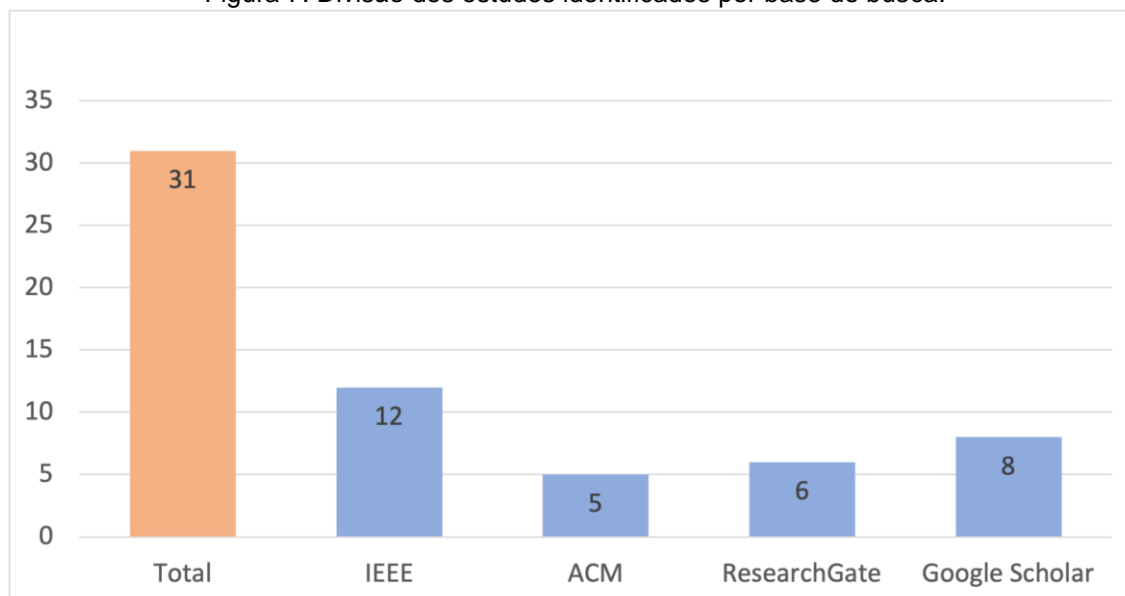
5. Seleção dos Estudos e Apresentação dos Resultados

5.1 Identificação dos Estudos

Para a identificação dos estudos, foram executadas as buscas de acordo com a estratégia definida na metodologia proposta na etapa 3.1. Para cada trabalho resultante das buscas, os seguintes dados foram registrados em uma

planilha do Google: título, autor, ano de publicação e a base de busca em que o trabalho foi encontrado. Ao todo, foram identificados 31 estudos. O resultado detalhado da identificação dos estudos pode ser visualizado na figura 7.

Figura 7: Divisão dos estudos identificados por base de busca.

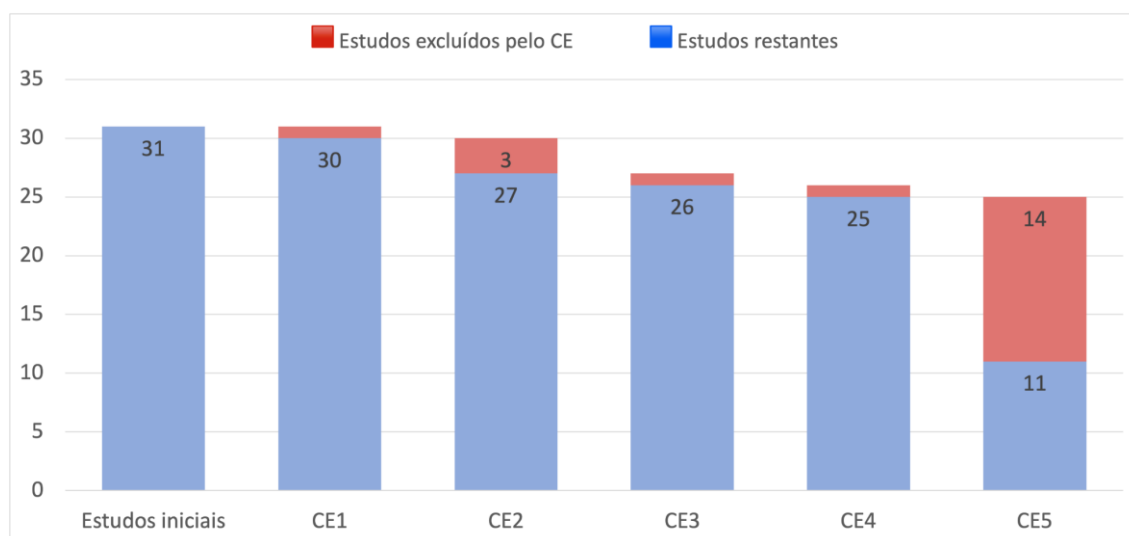


Fonte: Elaborado pelo autor.

5.2 Seleção dos estudos

Nesta etapa, os estudos foram submetidos aos critérios de exclusão estabelecidos em 3.1.1, a fim de selecionar aqueles que seriam utilizados nas etapas subsequentes. Os critérios foram aplicados em conformidade com a ordem definida e de maneira sequencial. Dessa forma, caso um artigo seja excluído na aplicação do critério CE1, não é necessário aplicar os critérios subsequentes para esse estudo em particular. A tabela abaixo apresenta a quantidade de estudos excluídos após a aplicação de cada critério de exclusão.

Figura 8: Estudos selecionados após a aplicação dos critérios de exclusão.



Fonte: Elaborado pelo autor.

Como mostrado na figura, dos 31 estudos em escopo, 1 foi excluído devido a não estar disponível para leitura online, para download ou de livre acesso via o acesso institucional ofertado pela faculdade. Após isso, dos 30 estudos restantes, 3 foram filtrados por se tratarem de trabalhos publicados antes do ano de 2018. Dentre os artigos/teses restantes após a aplicação dos critérios CE1 e CE2, 1 não foi publicado na língua inglesa e também foi excluído. Dos 26 trabalhos restantes, 1 era estudo duplicado. Por fim, 14 outros estudos foram filtrados por não se tratarem exatamente de pontos críticos ou de principais dificuldades percebidas dentro do tema "Migração de monolito para microserviços usando a técnica do estrangulamento".

Ao final da aplicação de todos os critérios de exclusão e inclusão, restaram 11 artigos. O detalhamento dos estudos selecionados (ES) se encontra no quadro 6.

Quadro 6: Detalhamento dos estudos selecionados (ES).

ID	Título do Estudo	Autores	Ano	Base
ES1	Migrating Towards Microservice	Di Francesco, Paolo; Lago,	2018	IEEE

	Architectures: An Industrial Survey	Patricia; Malavolta, Ivano		
ES2	Microservice Migration Using Strangler Fig Pattern: A Case Study on the Green Button System	Li, Chia-Yu; Ma, Shang-Pin; Lu, Tsung-Wen	2020	ResearchGate
ES3	How to Strangle Systematically: An Approach and Case Study for the Continuous Evolution of Monoliths to Microservices	Seifermann, Valentin	2021	Google Scholar
ES4	Microservice Migration Using Strangler Fig Pattern and Domain-Driven Design	Lee, Shin Jie; Ma, Shang Pin; Li, Chia Yu; Lee, Wen Tin	2022	Google Scholar
ES5	Microservices Migration in Industry: Intentions, Strategies and Challenges	Fritzsche, Jonas; Bogner, Justus; Wagner, Stefan; Zimmermann, Alfred	2019	IEEE
ES6	Monoliths to Microservices - Migration Problems and Challenges: A SMS	Velepucha, Victor; Flores, Pamela	2021	IEEE
ES7	Challenges When Moving from Monolith to Microservice Architecture	Kalske, Miika; Makitalo, Niko; Mikkonen, Tommi	2018	ResearchGate
ES8	Migrating from Monolithic Architecture to Microservices: A Rapid Review	Ponce, Francisco; Márquez, Gastón; Astudillo, Hernán	2019	IEEE
ES9	Methods and Process of Service Migration from Monolithic Architecture	Kyryk, Marian; Tymchenko, Oleksandr;	2022	IEEE

	to Microservices	Pleskanka, Nazar; Pleskanka, Mariana		
ES10	Cracking the Monolith: Challenges in Data Transitioning to Cloud Native Architectures	Mishra, Mayank; Kunde, Shruti; Nambiar, Manoj	2018	ACM
ES11	Towards a Roadmap for the Migration of Legacy Software Systems to a Microservice based Architecture	S. da Silva, Hugo; Carneiro, Glauco; Monteiro, Miguel	2023	ACM

Fonte: Elaborado pelo autor.

5.3 Extração dos Dados

Os dados obtidos de cada estudo, a fim de responder à pergunta da pesquisa, foram registrados em uma planilha, juntamente com o trecho que explicitava essa informação. Em estudos mais abrangentes, nos quais os trechos que comprovam os principais pontos de dificuldade não estavam necessariamente na mesma frase ou parágrafo, foram armazenados mais de um trecho para responder a essa pergunta.

5.4 Apresentação e Síntese dos Resultados

Nesta seção, serão apresentados os resultados desta pesquisa, por meio da síntese dos dados coletados na etapa anterior e apresentação da resposta para a pergunta da pesquisa.

Quais são os pontos críticos de uma migração de um sistema monolítico para uma arquitetura de microsserviços utilizando a técnica do estrangulamento?

Na tabela a seguir, é possível ver os pontos de maior dificuldade e criticidade apontados pelos estudos analisados.

Quadro 7: Síntese dos resultados encontrados nos estudos.

Ponto crítico	Estudos
Identificação dos limites entre os serviços para subdivisão	ES1, ES3, ES5, ES6, ES7, ES11
Processo de decomposição do monolito em serviços	ES1, ES2, ES4, ES7
Etapa de segmentação do banco de dados	ES6, ES8, ES10
Alto nível de interdependência entre os serviços do monolito	ES1, ES11
Definição da sequência ótima para decomposição do monolito	ES2, ES3
Limitações relacionadas à requisitos de tecnologia (deprecated tech)	ES3, ES5
Preparo do ambiente de produção para os primeiros microsserviços a serem migrados	ES4
Processo de integração dos serviços migrados com o software legado	ES4
Falta de conhecimento e familiaridade no campo de microsserviços	ES5
Processo de implementação automático dos serviços desacoplados	ES9

Fonte: Elaborado pelo autor.

É perceptível que a etapa de "identificação dos limites entre os serviços para subdivisão" possui uma maior incidência comum entre os estudos analisados (6/11). É válido ressaltar também os pontos do "processo de decomposição do monólito em serviços" (4/11) e a "etapa de segmentação do banco de dados" (3/11) que vêm logo em seguida como pontos sensíveis mais comumente relatados. As citações abaixo foram armazenadas na etapa de extração de dados e comprovam o relato dos pontos críticos em seus respectivos estudos selecionados.

"Regarding architecture transformation, the main challenges are (i) the high level of coupling, (ii) the difficulties in identifying the boundaries of services, and (iii) system decomposition." - **ES1**

"In the migration process, it is crucial to define the order in which services should be extracted. (...) we have to evaluate and compare each identified service's expected benefits and prioritize the extraction of services." - **ES2**

"The way in which service candidates are identified and prioritized for migration is an important factor for migration. In this context, Li et al. [LML20] point out that an extraction order is crucial for the migration to bring maximum benefits. (...) There were limitations and requirements related to technologies. (...) requirements had to be followed, and development had to be done with deprecated technologies." - **ES3**

"The first thorny problem encountered in microservice migration involves splitting the system into candidate microservices. (...) we also conclude two possible issues/difficulties when applying MMSD: (1) preparing the environment of hosting migrated microservices is non-trivial. (...) and (2) designing the interface of a migrated microservice is not simple

and even challenging. We need to carefully ensure the correct integration of the newly-built microservice and the remaining monolith." - **ES4**

"As such, finding the right service cut was a major technical challenge (...) As a second major challenge, we identified the lack of experience with Microservices-related technologies and concepts" - **ES5**

"The challenge in the migration process is to identify and isolate the business functionality and translate this functionality into a microservice. (...) The challenge is to perform a correct division of the database" - **ES6**

"When the starting point is a monolithic application, the organization is then most likely familiar with the domain already and have an idea where the seams of application can be found [1]. The biggest problem in these cases is separating these services. (...) It is important that the decomposition of services is correct." - **ES7**

"And that the main challenge is to perform the database migration. (...) The challenge that is transversal to most of the proposals is the migration of the database." - **ES8**

"The process of monolith application migration is presented. A few key steps, which are required for successful migration, are described. One of the most important things is automated deployment of microservices to the production environment." - **ES9**

"One of the biggest challenges is to maintain data consistency and statefulness across the services and enable a smooth transition of the data. (...) Data is the integral part of an application. Hence any form of transitioning of an application (e.g. monolith to cloud native) implies the transitioning of the underlying application data." - **ES10**

"(...) we highlight four key challenges faced during the migration. The first challenge is the identification of functionalities. (...) The second challenge comes from the need to define optimal boundaries among candidate features for microservices." - **ES11**

5.5 Limitações da Pesquisa e Ameaça À Validade

Para a execução deste trabalho foram utilizadas apenas 4 bases de estudos distintas (Google Scholar, ACM, IEEE, ResearchGate). Além disso, apenas os estudos que estavam disponíveis para acesso gratuito (com o uso do acesso institucional) foram utilizados, o que pode ter limitado o alcance da pesquisa. Apesar das bases escolhidas serem de grande qualidade e reconhecimento no ramo acadêmico, existem outras bases de dados de qualidade que poderiam ser utilizadas para agregar ainda mais valor e quantidade de estudos. Por fim, é válido ressaltar que mesmo que este trabalho considere estudos até o ano de 2023, ele está limitado aos estudos do ano de 2023 publicados apenas nos primeiros 4 meses do ano.

Outro ponto que merece destaque neste tópico foi o processo de decisão das palavras-chave nas etapas 3.1.1 e 3.1.3. Apesar da escolha das palavras-chave terem sido eficazes, ainda sim o processo possui um alto nível de empirismo e subjetividade, sendo assim, passível de aprimoramentos. Outro ponto passível de menção foi o montante restrito de estudos a serem filtrados em consequência da aplicação dos critérios de exclusão escolhidos. Isso acarretou em um número reduzido de artigos passíveis de serem selecionados para a etapa de extração de dados.

6. Conclusão e Trabalhos Futuros

6.1 Conclusão

Este estudo buscou trazer uma visão geral dos pontos críticos e principais dificuldades percebidas durante o processo de migração de um monólito para um microsserviço. Para tal, foi executada uma revisão bibliográfica, com características de revisão sistemática, que foi capaz de reunir informações para responder a problemática proposta na pesquisa. No processo de extração e

coleta dos dados, a seguinte pergunta foi aplicada nos estudos selecionados: "Quais são os pontos críticos de uma migração de um sistema monolítico para uma arquitetura de microsserviços utilizando a técnica do estrangulamento?".

Ao todo, foram analisados 11 artigos e os pontos críticos percebidos com maior número de incidência foram: a identificação dos limites entre os serviços para subdivisão, o processo de decomposição do monólito em serviços e a etapa de segmentação do banco de dados. Essas principais dificuldades foram percebidas em 6, 4 e 3 estudos respectivamente. Consequentemente, pode-se concluir que este estudo possui uma relevância significativa para a comunidade acadêmica, uma vez que oferece uma visão do panorama atual relacionado aos pontos críticos identificados durante a migração de um monólito para uma arquitetura de microsserviços utilizando o método de estrangulamento. Por meio deste trabalho, é possível compreender, no âmbito dos estudos em análise, as dificuldades e os momentos críticos mais comumente observados no processo de migração investigado.

Assim, esse trabalho busca impactar positivamente a experiência de profissionais futuros que estejam envolvidos na migração de arquiteturas de software, fornecendo um compartilhamento prévio de potenciais pontos sensíveis e que requerem atenção durante a tarefa. Isso permitirá uma visão mais abrangente e antecipada dos desafios envolvidos nesse processo de migração, contribuindo para que esta etapa seja menos complexa e mais assertiva.

6.2 Trabalhos Futuros

Com base no estudo realizado, sugerem-se os seguintes direcionamentos de pesquisas para trabalhos futuros:

- Usar esse projeto de pesquisa como base para o desenvolvimento de uma revisão sistemática formal e completa, tratando primariamente atender as limitações apontadas no tópico de ameaça à validade;
- Revisitar e adaptar os critérios de exclusão e inclusão visando abranger o universo de estudos passíveis de serem selecionados e, consequentemente, utilizados para extração de dados;
- Estender a pesquisa para incluir mais bases de dados;
- Estender a pesquisa para que sejam considerados os usos de estudos pagos;
- Desenvolver testes práticos em diferentes cenários visando aprimorar a análise de momentos críticos no processo da migração na área de migração entre arquiteturas monolíticas para microserviços;

Referências

[1] BASS, L., CLEMENTS, P., and KAZMAN, R.: **Software Architecture in Practice**. Addison-Wesley Professional, 2003. Acesso em: 18 de out. de 2022.

[2] DRAGONI, N., GIALLORENZO, S., LAFUENTE, A. L., MAZZARA, M., MONTESI, F., MUSTAFIN, R., SAFINA, L.: **Microservices: yesterday, today, and tomorrow. In Pre- sent and Ulterior Software Engineering**. Springer, 2017. Acesso em: 18 de out. de 2022.

[3] NEWMAN, S.: **Building Microservices**. O'Reilly, 2015. Acesso em: 15 de out. de 2022.

[4] NEWMAN, S.: **Monolith to Microservices: Evolutionary Patterns to Transform Your Monolith**. O'Reilly, 2020. Acesso em: 15 de out. de 2022.

- [5] TAIBI, D., LENARDUZZI, V., PAHL, C.: **Architectural Patterns for Microservices: A Systematic Mapping Study**. Disponível em: <https://bia.unibz.it/esploro/outputs/conferenceProceeding/Architectural-Patterns-for-Microservices-A-Systematic-Mapping-Study/991005773017601241> . Acesso em: 25 de out. de 2022.
- [6] FOWLER, M., LEWIS, J.: **Microservices**. 2014. Disponível em: <https://martinfowler.com/articles/microservices.html> . Acesso em: 25 de out. de 2022.
- [7] FOWLER, M.: **Strangler Fig Application**. 2004. Disponível em: <https://martinfowler.com/bliki/StranglerFigApplication.html> . Acesso em: 25 de out. de 2022.
- [8] MISHRA, M., KUNDE, S., NAMBIAR, M.: **Cracking the monolith: challenges in data transitioning to cloud native architectures**. 2018. Disponível em: <https://dl.acm.org/doi/abs/10.1145/3241403.3241440> . Acesso em: 02 de nov. de 2022.
- [9] S. da SILVA, H., CARNEIRO, G., MONTEIRO, M.: **Towards a Roadmap for the Migration of Legacy Software Systems to a Microservice based Architecture**. Acesso em: 02 de nov. de 2022.
- [10] IHDE, S.: **From a Monolith to Microservices + REST: the Evolution of LinkedIn's Service Architecture**. 2014. Acesso em: 02 de nov. de 2022.
- [11] INGENO, J.: **Software Architect's Handbook**. O'Reilly, 2018. Acesso em: 02 de nov. de 2022.
- [12] GARLAN, D.: **Software Architecture: A Travelogue**. FOSE, 2014. Disponível em: <https://dl.acm.org/doi/10.1145/2593882.2593886> . Acesso em 09 de nov. de 2022.
- [13] AL-DEBAGY, O., MARTINEK, P.: **A Comparative Review of Microservices and Monolithic Architectures**. IEEE International Symposium on Computational Intelligence and Informatics, 2018. Disponível em: <https://ieeexplore.ieee.org/abstract/document/8928192> . Acesso em 09 de nov. de 2022.
- [14] RICHARDSON, C.: **Pattern: Monolithic Architecture**. Disponível em: <https://microservices.io/patterns/monolithic.html> . Acesso em 09 de nov. de 2022.
- [15] RICHARDSON, C.: **Pattern: Microservice Architecture**. Disponível em: <https://microservices.io/patterns/microservices.html> . Acesso em 09 de nov. de 2022.

- [16] LI, C. Y., MA, S. P., LU, T. W.: **Microservice Migration Using Strangler Fig Pattern: A Case Study on the Green Button System**. IEEE International Computer Symposium (ICS), 2020. Disponível em: <https://ieeexplore.ieee.org/abstract/document/9359092/authors> . Acesso em 09 de nov. de 2022.
- [17] FAN, C. Y., MA, S. P.: **Migrating Monolithic Mobile Application to Microservice Architecture: An Experiment Report**. IEEE International Conference on AI & Mobile Services (AIMS), 2017. Disponível em: <https://ieeexplore.ieee.org/abstract/document/8027278/authors> . Acesso em 09 de nov. de 2022.
- [18] HAMMANT, P.: **Legacy Application Strangulation: Case Studies**, 2013. Disponível em: <https://paulhammant.com/2013/07/14/legacy-application-strangulation-case-studies/> . Acesso em 09 de nov. de 2022.
- [19] TAIBI, D., LENARDUZZI, V., PAHL, C.: **Processes, Motivations, and Issues for Migrating to Microservices Architectures: An Empirical Investigation**. IEEE Cloud Computing, 2017. Acesso em: 18 de out. de 2022.
- [20] GIL, Antônio Carlos. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2002. Acesso em: 08 de abril de 2023.
- [21] VERGARA, S. C. **Projetos e relatórios de pesquisa em administração**. 5. ed. São Paulo: Atlas, 2004. Acesso em: 08 de abril de 2023.
- [22] GODOY, A. S. **Introdução a pesquisa qualitativa e suas possibilidades**. Revista de Administração de Empresas. São Paulo, v. 35, n. 2, p. 57-63, Mar./Abr. 1995B. Acesso em: 08 de abril de 2023.
- [23] LLEWELLYN, S.; NORTHCOTT, D. The “singular view” in management case studies qualitative research in organizations and management. **An International Journal**, v. 2, n. 3, p. 194-207, 2007.