

Modularidade e Escalabilidade entre Frontends e Micro-frontends para Sistemas Web Modernos: uma Revisão Rápida da Literatura

Fernanda Fernandes Barrigosse¹

Orientador: Ricardo Theis Geraldi¹

¹ Faculdade de Computação (FACOM) – Universidade Federal de Mato Grosso do Sul (UFMS), Campo Grande (MS) - Brazil

fernanda.barrigosse@ufms.br, ricardo.geraldi@ufms.br

Abstract. Context: Currently, society's demand for modern web systems is exponential for solving or managing problems across various domains such as agriculture, healthcare, smart cities, and transportation. This demand has led to the emergence of increasingly complex and larger systems. **Problem:** In this context, there are several challenges and problems related to modularity and scalability that affect the front-ends of these systems, making it difficult to ensure their evolution. **Gap:** This work explores the micro-frontend architecture as a path to evolve the non-functional requirements of modularity and scalability in the development of conventional web front-ends. **Objective:** To investigate the main motivations, benefits, and problems associated with adopting micro-frontends from the perspective of their modularity and scalability. **Method:** Procedures from the Rapid Literature Review (RRL) research method were adapted to analyze and select studies published between 2020 and 2024, based on updating a multi-vocal secondary study published in 2021. **Results:** Significant evidence suggests that micro-frontends promote team autonomy, scalability, and component reuse, but present challenges and problems in module orchestration, dependency management, and maintaining user interface and experience (UI/UX) consistency. **Conclusion:** This work primarily contributes to the theoretical-practical understanding of the modularity and scalability of the micro-frontend architecture and offers insights for professionals and researchers in companies to aid in the evolution of their modern web systems.

Resumo. Contexto: Atualmente a demanda da sociedade por sistemas web modernos é exponencial para resolver ou gerenciar problemas em diversos domínios da sociedade, como na agricultura, saúde, cidades, transporte dentre outros. Essa demanda tem provocado o surgimento de sistemas complexos e cada vez maiores. **Problema:** Neste contexto, há vários desafios e problemas de modularidade e escalabilidade que afetam o front-end desses sistemas, dificultando a evolução dos mesmos. **Lacuna/Gap:** Este trabalho explora a arquitetura de micro-frontends como um caminho para evolução dos requisitos não-funcionais de modularidade e escalabilidade no desenvolvimento de front-ends web convencionais. **Objetivo:** Investigar as principais motivações, os benefícios e problemas associados à adoção de micro-frontends do ponto de vista da sua modularidade e escalabilidade. **Método:** Foram adaptados procedimentos do método de pesquisa Revisão Rápida de Literatura (RRL) para analisar e selecionar estudos teóricos-práticos publicados entre 2020 e 2024 com base na atualização de um estudo secundário multi-vocal publicado em 2021. **Resultados:** Apresentam indícios significativos que os

micro-frontends promovem autonomia das equipes, escalabilidade e reutilização de componentes, mas apresentam desafios e problemas na orquestração de módulos, gestão de dependências e manutenção da consistência da interface do usuário (UI/UX). Conclusão: Este trabalho contribui, principalmente, para a compreensão teórica-prática da modularidade e escalabilidade da arquitetura de micro-frontends e oferece insights para profissionais e pesquisadores a fim de ajudar na evolução de seus sistemas web modernos.

1. Introdução

Nas últimas décadas, o desenvolvimento de software tem sido atingido por transformações significativas, impulsionado pelo aumento da complexidade de sistemas de software web modernos e pelas demandas crescentes por escalabilidade, modularidade e eficiência de software [1]. As arquiteturas monolíticas, amplamente utilizadas no início da era do desenvolvimento web, começaram a enfrentar desafios substanciais à medida que os sistemas se tornaram maiores e mais complexos. Em resposta, surgiram novas abordagens arquiteturais, como os microsserviços, que trouxeram uma revolução no desenvolvimento backend ao introduzirem modularidade e independência entre componentes [2].

No entanto, enquanto o backend avançava em direção à modularização, o frontend permaneceu tradicionalmente monolítico, carregando as mesmas limitações que as arquiteturas antigas enfrentavam no backend. O crescimento exponencial de sistemas de software web modernos e a necessidade de interfaces cada vez mais dinâmicas e responsivas impulsionaram o surgimento de novas estratégias para o desenvolvimento frontend, como a componentização, *Single Page Applications* (SPAs) e *Progressive Web Applications* (PWAs) [3].

Nesse cenário, a arquitetura de *micro-frontends* emergiu como uma solução inovadora, permitindo que a modularidade (característica dos microsserviços) fosse adaptada ao desenvolvimento de interfaces de usuário. Essa arquitetura possibilita que diferentes times trabalhem em partes isoladas do sistema, promovendo maior flexibilidade, escalabilidade e independência no desenvolvimento de sistemas web modernos [4].

Este trabalho tem como objetivo principal realizar uma Revisão Rápida da Literatura (RRL) para elucidar os principais conceitos e investigar as motivações, os benefícios e problemas relacionados à adoção de *micro-frontends* perante a sua modularidade e escalabilidade. Em adição, este trabalho busca atualizar e estender a revisão de literatura multivocal realizada por Peltonen *et al.* (2021) [8], identificando novos avanços, tendências e práticas associadas à utilização dessa arquitetura em diferentes contextos de desenvolvimento.

Espera-se contribuir para um entendimento mais abrangente das aplicações teórico-práticas dos *micro-frontends*. A análise dos resultados revelou que, embora os *micro-frontends* ofereçam vantagens significativas em termos de modularidade e escalabilidade, sua implementação exige um planejamento cuidadoso para lidar com desafios técnicos e organizacionais no meio acadêmico ou na indústria de software.

Este trabalho está estruturado da seguinte forma: seção 2, apresenta o background com os principais conceitos sobre arquitetura monolítica, frontend, escalabilidade,

modularidade, microsserviços e *micro-frontends*; seção 3, apresenta a metodologia utilizada para a realização dessa RRL; seção 4, apresenta os resultados e discussão divididos pelas questões de pesquisa, além de uma síntese dos resultados. Por fim, a seção 5 apresenta a conclusão.

2. Background

2.1. Arquitetura monolítica

A arquitetura monolítica é consolidada no desenvolvimento de software, sendo prática a integração de todos os componentes e funcionalidades em um único programa [5]. Entretanto, à medida que os sistemas de software web modernos cresceram em tamanho e complexidade, os desafios em projetar arquiteturas ou tomar decisões arquiteturais começaram a surgir [1].

Um dos principais obstáculos das arquiteturas monolíticas é o forte acoplamento entre os elementos. Isso significa que as diferentes partes de sistemas de software web modernos estão intimamente interligadas, tornando as mudanças difíceis e arriscadas. Por exemplo, a atualização de um único componente pode exigir a recompilação de todo o sistema [6]. Isso não só aumenta o tempo e os custos de desenvolvimento, como também o risco de introduzir novos bugs e falhas.

À medida que a base de código aumenta, associado a decisões arquiteturais, a complexidade cresce exponencialmente, dificultando a manutenção e implementação de novos recursos. Em adição, a escalabilidade torna-se um problema, já que a adição de recursos exige escalar todo o sistema [7].

Outro desafio significativo é a coordenação entre os times de desenvolvimento. Com todos os serviços e componentes integrados em um único programa, é comum que diferentes equipes tenham que trabalhar na mesma base de código, o que pode levar a conflitos e atrasos. A integração contínua e os testes automatizados tornam-se mais difíceis de gerenciar, aumentando ainda mais a complexidade do desenvolvimento [8].

Para mitigar esses problemas, a arquitetura baseada em microsserviços emergiu como uma alternativa promissora para a evolução no desenvolvimento de software na camada de backend. Essa arquitetura divide o sistema em pequenos serviços independentes, cada um responsável por uma funcionalidade específica [1]. Essa modularidade permite que cada microsserviço seja desenvolvido, implantado e escalado de forma autônoma, promovendo maior coesão interna e reduzindo o acoplamento entre os serviços. O backend, que é a camada responsável pelo processamento de dados, regras de negócio e comunicação com bancos de dados, se beneficia significativamente dessa independência para atender demandas complexas e escaláveis [9].

No frontend, que é a camada visível ao usuário e responsável pela interação e apresentação de dados, essa mesma filosofia modular foi adaptada na forma de *micro-frontends*. Assim, como os microsserviços no backend, os *micro-frontends* promovem a separação e a independência dos componentes de interface, permitindo que cada um seja desenvolvido e mantido de forma isolada, promovendo forte coesão e baixo acoplamento [10]. Essa abordagem modular, que será explorada nas próximas

seções, facilita a escalabilidade e a manutenção das interfaces de usuário, atendendo às demandas de sistemas de software web modernos e dinâmicos [9].

2.2. Frontend

O frontend de um sistema web moderno é responsável pela parte visual dos sites e pela experiência do usuário. Nos últimos anos, o desenvolvimento de sistemas têm evoluído em tamanho e em inovações, gerando uma necessidade de interfaces mais dinâmicas, responsivas e intuitivas [10].

Tradicionalmente, o desenvolvimento frontend era feito de maneira monolítica. Entretanto, essa prática convergiu para limitações, que afetam requisitos não-funcionais nas arquiteturas como:

- **Escalabilidade:** Em um sistema monolítico, todas as partes do frontend estão conectadas. Escalar alguma parte do frontend seria um desafio porque poderia impactar no desenvolvimento do sistema como um todo [11].
- **Modularidade:** É a capacidade de estruturar um sistema por meio de unidades separadas e independentes que, quando combinadas, resultam em um conjunto coerente [14].
- **Manutenção:** Partindo do mesmo princípio do requisito anterior, por ter um código base grande, qualquer mudança deve ser analisada antes de ser implementada para não retornar efeitos indesejáveis [1].
- **Flexibilidade:** Na eventual necessidade de uma mudança significativa, será preciso reescrever boa parte do código [1].

Para superar essas limitações, surgiram novas abordagens no desenvolvimento frontend:

- **Componentização:** Uma das abordagens mais comuns é a componentização, na qual a interface do usuário é dividida em componentes reutilizáveis e independentes. Frameworks como React, Vue.js e Angular promovem a componentização, permitindo que desenvolvedores criem interfaces complexas a partir de partes menores e mais gerenciáveis [12].
- **Single Page Applications (SPAs):** SPAs carregam uma única página HTML e atualizam dinamicamente o conteúdo conforme o usuário interage com o sistema. Isso proporciona uma experiência de usuário mais rápida e fluida, similar a um aplicativo desktop [8].
- **Progressive Web Apps (PWAs):** São sistemas web que utilizam tecnologias modernas para oferecer uma experiência semelhante à de um aplicativo nativo, incluindo a capacidade de funcionar offline e enviar notificações push [11].

O avanço contínuo dessas abordagens tem tornado as interfaces mais dinâmicas, intuitivas e escaláveis, contribuindo para o atendimento das demandas de usabilidade e desempenho em sistemas modernos. Não apenas otimizaram o desenvolvimento frontend, mas também enfatizaram a importância de organizar os sistemas de maneira modular, permitindo maior independência entre as partes e simplificando a adaptação a novas demandas [13].

2.3. Escalabilidade

De acordo com a norma ISO/IEC 25000, referente a atributos de qualidade de sistemas e software, “*a escalabilidade é o grau em que um produto pode lidar com cargas de trabalho crescentes ou decrescentes, ou adaptar sua capacidade para lidar com a variabilidade*”. Em outras palavras, é a habilidade de um sistema de se manter eficiente à medida que a complexidade do sistema e a carga de processamento aumentam.

Outra maneira de definir a escalabilidade nessa perspectiva é em relação à organização das equipes de desenvolvimento. Com o crescimento do ciclo de vida de sistemas, torna-se cada vez mais necessário escalar times responsáveis pela sua manutenção e atualizações. Em sistemas monolíticos, esse crescimento dificulta a coordenação entre múltiplos times, impactando negativamente a agilidade nos processos de desenvolvimento e liberação de novas funcionalidades [10].

Em suma, a escalabilidade tornou-se um requisito fundamental para o desenvolvimento de sistemas web devido à crescente complexidade dos sistemas, ao aumento do número de usuários e à necessidade de entregas ágeis e eficientes [1]. Arquiteturas que suportam a escalabilidade, tanto em termos de desempenho do sistema quanto em termos da organização das equipes de desenvolvimento, são essenciais para garantir o sucesso de projetos de software modernos.

2.4. Modularidade

A modularidade é uma característica/atributo de qualidade ou requisito não-funcional fundamental no desenvolvimento de sistemas web. A modularidade é a capacidade de criar componentes independentes que, em conjunto, formam um projeto coeso. Esses componentes podem ser chamados de módulos [14].

De acordo com a norma ISO/IEC 25000, referente a atributos de qualidade de sistemas e software, “*a modularidade é a capacidade de um sistema ser decomposto em módulos distintos e reutilizáveis*”. A modificação ou substituição de um módulo deve impactar minimamente sobre os outros módulos. Além disso, a modularidade promove a reutilização de código, facilita a manutenção e melhora a colaboração entre equipes de desenvolvimento [10].

A modularidade é um princípio que transcende o frontend e backend, sendo um dos pilares fundamentais na evolução da arquitetura de sistemas de software modernos, como os microsserviços. Essa relação evidencia como a decomposição modular aplicada no desenvolvimento de sistemas é a base para arquiteturas mais independentes e escaláveis [13].

2.5. Microsserviços

Os microsserviços surgiram como uma solução para os problemas das arquiteturas monolíticas. São responsáveis por dividir o sistema em pequenos serviços independentes, sendo cada um deles responsável por uma função específica. Cada microsserviço é desenvolvido, implantado e escalado de forma independente, o que promove a modularidade [10].

Outra vantagem significativa é a escalabilidade. Como cada microsserviço pode ser escalado individualmente, as organizações conseguem otimizar recursos e lidar melhor com demandas variáveis em diferentes partes do sistema. Além disso, a manutenção do sistema é facilitada, uma vez que falhas em um serviço não impactam diretamente os demais, aumentando a resiliência geral do sistema [1].

A arquitetura de microsserviços é amplamente adotada por empresas que lidam com grandes volumes de dados e operações complexas, devido à sua capacidade de suportar sistemas dinâmicos e de alta disponibilidade [1]. A similaridade conceitual com os *micro-frontends*, que abrangem a modularidade para o frontend, reforça o alinhamento entre essas duas abordagens modernas, criando um ecossistema integrado e eficiente no desenvolvimento de sistemas.

2.6. *Micro-frontends*

Os *micro-frontends* foram introduzidos em 2016 [12] e buscam resolver desafios associados à complexidade, escalabilidade e falta de modularidade no desenvolvimento frontend. Essa arquitetura aplica os princípios dos microsserviços ao desenvolvimento frontend [11].

Os *micro-frontends* dividem a interface gráfica do usuário em diferentes módulos independentes, como um quebra-cabeças. Cada módulo é responsabilidade de um time específico e pode ser desenvolvido de maneira independente [8]. Essa separação permite que equipes diferentes trabalhem em paralelo, utilizando tecnologias e processos de desenvolvimento que melhor atendem às suas necessidades.

Vantagens dos *micro-frontends* [8]:

- **Desenvolvimento mais rápido:** Equipes podem trabalhar simultaneamente em diferentes partes do sistema;
- **Menor impacto de falhas:** Devido à modularização das partes, se houver uma falha o erro estará isolado apenas ao segmento que pertence;
- **Equipes com maior autonomia:** Cada equipe pode escolher as tecnologias e ferramentas que acham mais adequadas às suas necessidades; e
- **Código modular e reutilizável:** Facilita a manutenção, resultando em um código mais limpo e gerenciável.

Desafios dos *micro-frontends* [8]:

- **Dependências duplicadas:** Devido a divisão dos *micro-frontends*, se não houver um bom gerenciamento de dependências pode ser que diferentes equipes tenham dependências duplicadas;
- **Consistência da experiência de usuário:** Manter uma interface e experiência de usuário padrão entre diferentes equipes pode ser desafiador; e
- **Aumento da complexidade:** A configuração inicial e integração dos *micro-frontends* deve ser planejada de maneira diligente e pode se tornar muito complexa.

2.7. Microsserviços vs. *micro-frontends*

Microsserviços e *micro-frontends* são duas abordagens arquitetônicas que têm recebido crescente atenção na comunidade acadêmica e na indústria nos últimos anos. Enquanto os microsserviços são amplamente reconhecidos como uma maneira eficaz de lidar com complexidade e escalabilidade no backend de sistemas de software, os *micro-frontends* surgiram como uma abordagem complementar para modularizar e desacoplar o frontend dos sistemas [11].

Os microsserviços têm como propósito dividir sistemas monolíticos em componentes menores e mais gerenciáveis. “*Os principais ideais de microsserviços são: (i) serem pequenos, (ii) focados em uma única tarefa; e (iii) autônomos*” [15]. Ao aproximar essa abordagem para o frontend é possível ter maior modularidade e independência no desenvolvimento [11].

Ao dividir o frontend em *micro-frontends*, cada módulo pode ser elaborado por equipes independentes [16], seguindo os princípios de autonomia encontrados nos microsserviços. Essa separação permite que cada parte da interface do usuário seja desenvolvida, testada, implantada e escalada de forma independente, facilitando a colaboração entre equipes de desenvolvimento e promovendo a reutilização de código [13].

Neste contexto, os *micro-frontends* oferecem flexibilidade na escolha de tecnologias, permitindo que cada *micro-frontend* seja desenvolvido utilizando a tecnologia mais adequada para a tarefa específica que realiza [11]. Assim, tanto os microsserviços quanto os *micro-frontends* representam estratégias valiosas para lidar com a complexidade e os desafios do desenvolvimento de sistemas de software web modernos.

A crescente popularidade de *micro-frontends*, discutida na seção anterior, motivou a realização desta pesquisa. Com o objetivo de melhor compreender os benefícios, problemas e tendências associados a essa arquitetura, foi conduzida uma revisão rápida da literatura. A seção seguinte detalha a metodologia empregada para identificar, avaliar e analisar os estudos.

3. Metodologia

Este estudo tem como objetivo principal estender o trabalho de Peltonen *et al.* (2021) [8] até Julho/2024, apoiado por procedimentos do método de Revisão de Rápida da Literatura (RRL) [17]. A RRL é uma abordagem que visa identificar, avaliar e interpretar estudos teóricos-práticos disponíveis sobre um tópico específico (*micro-frontends*), com o objetivo de responder a questões de pesquisa. Adota-se uma perspectiva sistemática de uma RRL para enriquecer a análise dos dados, permitindo uma interpretação mais profunda e abrangente dos resultados. Para esta revisão, não foram considerados textos de literatura cinza e livros sobre o assunto [8].

A seleção dos estudos foi conduzida por meio de 3 filtragens seriadas, conforme detalhado no diagrama representado na Figura 1. O diagrama ilustra que no filtro 1 foi realizada uma busca inicial nas bases de dados com a seguinte *string* de pesquisa: “*micro-frontend*” OR “*micro frontend*”. Todos os resultados gerados nesta etapa fazem parte do filtro 1.

Em seguida, no filtro 2, foram aplicados os critérios de exclusão e realizada a leitura dos resumos, introduções e conclusões dos textos para seleção. No filtro 3, os estudos que atenderam aos critérios da etapa anterior foram analisados em profundidade. Nessa etapa, foi realizada a leitura completa dos estudos do filtro 3, com o objetivo de avaliar a relevância e detalhar os resultados para esta pesquisa.

A Figura 2 complementa essa análise ao apresentar a evolução temporal das publicações sobre *micro-frontends* e mostra um pico em 2023. Acredita-se que essa concentração possa indicar um retorno das pesquisas e um aumento no interesse acadêmico e profissional por *micro-frontends*, possivelmente após um período de menor atividade em 2020, que pode ter sido influenciado pela pandemia, e impulsionado pelo avanço das tecnologias web.

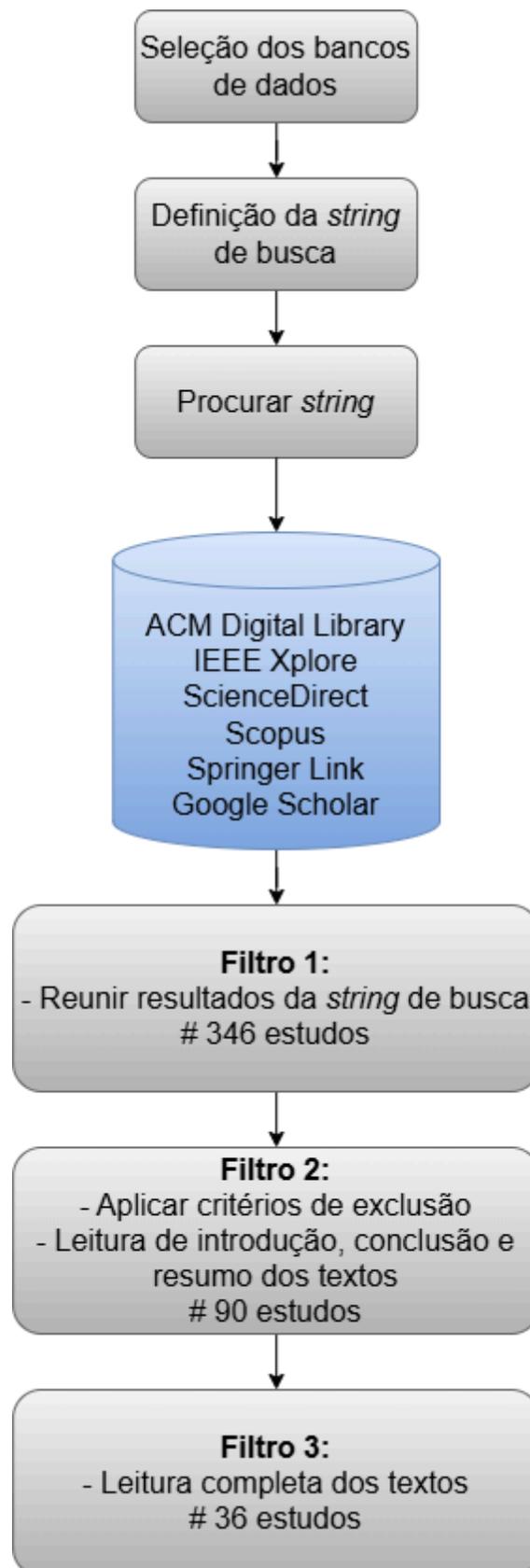


Figura 1. Filtros da pesquisa para recuperar e selecionar os estudos da literatura.

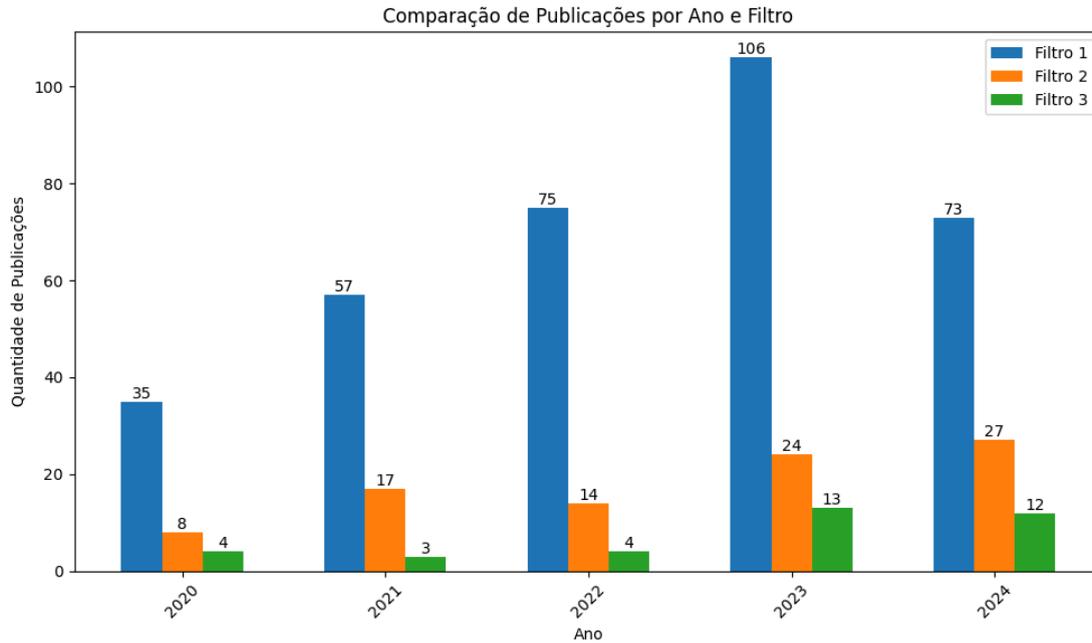


Figura 2. Quantidade de Publicações por Ano x Filtro.

3.1. Fontes de Pesquisa

As bases digitais deste estudo foram previamente selecionadas considerando a relevância e credibilidade científica de cada uma. As bases de dados digitais científicas adotadas foram:

Tabela 1. Bases utilizadas durante o estudo

Plataformas	Links
ACM digital Library	https://dl.acm.org/
IEEE Xplore	https://ieeexplore.ieee.org
ScienceDirect	https://www.sciencedirect.com/
Scopus	https://www.scopus.com/
Springer Link	https://link.springer.com/
Google Scholar	https://scholar.google.com/

As plataformas Citeseer Library e Inspec, mencionadas no estudo de Peltonen *et al.* (2021) [8], não foram utilizadas aqui devido à ausência completa de resultados ao aplicar a *string* de pesquisa "*micro-frontend*" OR "*micro frontend*". No estudo de referência de Peltonen *et al.* (2021), a pesquisa foi realizada utilizando *strings* similares ("*Micro-Frontend**" e "*Micro Frontend**"). Nesta pesquisa, a recuperação dos estudos foi realizada entre janeiro/2020 e julho/2024.

3.2. Questões de Pesquisa

Os estudos que atenderam aos critérios de inclusão foram analisados de forma a responder às seguintes questões (questões traduzidas na íntegra de Peltonen *et al.*, (2021) [8]):

Q1 Quais são as Motivações para adoção de *micro-frontend*?

Essa questão busca entender porque as organizações optam por adotar a arquitetura de *micro-frontend*.

Q2 Quais são os Benefícios alcançados por utilizar *micro-frontend*?

Essa questão se concentra em examinar as vantagens práticas proporcionadas pela adoção de *micro-frontends*.

Q3 Quais Problemas são introduzidos pelo uso de *micro-frontend*?

A investigação é voltada para os desafios e problemas que surgem com a implementação de *micro-frontends*. Nenhuma solução tecnológica é isenta de dificuldades. Essa pergunta visa mapear as barreiras ou complicações que as organizações enfrentam ao adotar essa arquitetura.

As questões de pesquisa definem o direcionamento do estudo, guiando a análise dos dados coletados.

3.3. Critérios de Inclusão

Os seguintes critérios foram estabelecidos para a inclusão dos estudos:

- Publicações realizadas entre janeiro de 2020 e julho de 2024;
- Estudos que abordam o tema de *micro-frontends*;
- Textos escritos em inglês [8] ou português;

3.4. Critérios de Exclusão

Os critérios para exclusão de textos foram:

- “Adoção do termo "*micro-frontend*" para diferentes propósitos ou domínios” [8];
- Publicações no formato de livros;
- Teses ou dissertações;
- Literatura cinza (blogs, sites oficiais, artigos de senso comum);
- “Resultados duplicados” dentre as bases de pesquisa. [8]

A seção 4 organiza e apresenta os resultados em torno dessas questões, permitindo uma discussão detalhada sobre os aspectos motivadores, os benefícios alcançados e os problemas enfrentados com a adoção de *micro-frontends*. Ao responder a essas questões, este estudo contribui para o avanço do conhecimento sobre o tema e oferece insights valiosos para profissionais e pesquisadores da área.

4. Resultados e Discussão

A discussão dos resultados encontrados está organizada em torno das três principais questões de pesquisa: motivações, benefícios e problemas associados à adoção de

micro-frontends com comparações diretas ao estudo de Peltonen *et al.* (2021) [8], que utilizou 43 fontes, das quais 40 eram de literatura cinza. Agora, com o aumento da utilização de *micro-frontends* foi possível encontrar 345 fontes iniciais, dentre elas, estudos, livros e dissertações. Após análise dos critérios de exclusão, que incluíram também a remoção de textos de literatura cinza, 35 estudos foram selecionados para análise final.

A combinação de procedimentos da RRL mostrou-se eficaz para identificar e analisar a crescente literatura teórico-prática e atual sobre *micro-frontends*. A partir da análise dos 35 estudos selecionados, foi possível construir uma visão mais completa e atualizada sobre o tema. A seguir, serão apresentados e discutidos os resultados obtidos para cada uma das questões de pesquisa.

4.1. Motivações para a Adoção de *Micro-Frontends* (Q1)

Este tópico aborda diretamente a Q1, que investiga os fatores práticos que levam as organizações a adotarem *micro-frontends*. A necessidade de **escalabilidade e a delegação de responsabilidades são aspectos centrais que impulsionam essa transição**. O uso de *micro-frontends* não apenas facilita a manutenção de sistemas em expansão, mas também permite a **atualização independente de componentes, reduzindo o risco de falhas sistêmicas** [10].

A principal motivação para a adoção de *micro-frontends* está relacionada à complexidade crescente dos sistemas monolíticos. À medida que o número de funcionalidades nas aplicações frontend aumenta, o gerenciamento feito por uma única equipe se torna desafiador. Nesse contexto, grandes empresas, como Ikea e Spotify, optaram por implementar *micro-frontends* para permitir que **equipes autônomas gerenciem e desenvolvam partes específicas do sistema** [18]. Acredita-se que essa autonomia proporciona um modelo de trabalho mais eficiente, com menos dependências e maior capacidade de resposta às mudanças do mercado.

As motivações para a adoção de *micro-frontends* podem ser agrupadas em diferentes categorias. A primeira delas refere-se ao **crescimento da complexidade, que torna a manutenção e a evolução do software cada vez mais difíceis** [12]. À medida que os sistemas crescem em escopo e funcionalidades, **as bases de código se tornam volumosas e difíceis de gerenciar**, resultando em tempos de compilação mais longos, aumento de erros de integração e dificuldades para implementar novas funcionalidades de forma ágil. Além disso, em sistemas monolíticos, **a coordenação entre equipes pode ser prejudicada**, já que múltiplos desenvolvedores precisam trabalhar em uma mesma base de código, gerando conflitos de versões e dependências intrincadas. Assim, a adoção de uma arquitetura de *micro-frontends* se apresenta como uma solução viável para mitigar essas dificuldades [8].

Outra categoria envolve **a necessidade de escalabilidade das equipes de desenvolvimento** [12]. Com a expansão contínua das funcionalidades, a capacidade de implementar mudanças rápidas e eficazes torna-se essencial. A arquitetura de *micro-frontends* permite que diferentes equipes trabalhem em partes do sistema de forma independente, resultando em **maior agilidade no desenvolvimento e na entrega de novas funcionalidades**. Assim, as motivações para a adoção de *micro-frontends*

refletem as **necessidades de modernização e eficiência no desenvolvimento de software** [10]. A Figura 3 apresenta um diagrama que sintetiza os principais tópicos discutidos nesta seção sobre as motivações para a adoção de *micro-frontends*.

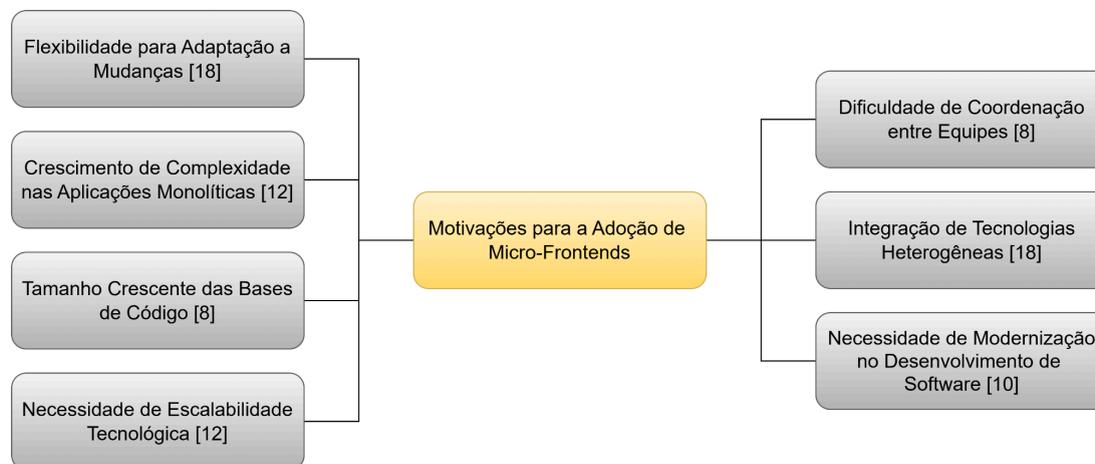


Figura 3. Principais categorias da Q1 com as principais motivações para a adoção de *micro-frontends*.

O diagrama na Figura 3 destaca como a constante evolução e o crescimento dos sistemas de software web modernos aumentaram a complexidade dos sistemas monolíticos, gerando a necessidade de novas abordagens arquiteturais. Ademais, evidencia fatores como escalabilidade, autonomia das equipes e facilidade de manutenção, que impulsionam a adoção dessa arquitetura.

4.2. Benefícios da Adoção de *Micro-Frontends* (Q2)

A adoção da arquitetura de *micro-frontends* apresenta uma série de vantagens significativas no desenvolvimento de sistemas de software web modernos. Essa arquitetura surge como uma solução eficaz para enfrentar a **complexidade crescente dos frontends monolíticos**, ao possibilitar a **decomposição de sistemas** amplos em unidades menores e melhor gerenciáveis. O que não apenas **simplifica o processo de desenvolvimento**, mas também **facilita a manutenção contínua**, conferindo às equipes uma agilidade maior na implementação de melhorias [10].

Um dos seus principais benefícios é a capacidade de promover um desenvolvimento eficiente, permitindo que **equipes independentes trabalhem simultaneamente em diversas partes do sistema** [18]. Essa **autonomia** não apenas melhora a colaboração entre as equipes, mas contribui significativamente para a escalabilidade do sistema. Neste sentido, as equipes têm a liberdade de escolher diferentes frameworks e bibliotecas que atendam melhor às suas necessidades específicas, promovendo uma adaptação mais eficaz às demandas do projeto [6].

Outro benefício crucial é a **escalabilidade**. A arquitetura de *micro-frontends* viabiliza a expansão dos sistemas de maneira controlada, já que cada módulo pode ser escalado individualmente sem impactar o restante do sistema. Isso é particularmente relevante em projetos de longo prazo, nos quais a demanda por recursos pode variar significativamente. Paralelamente, a **estrutura modular** promove a reutilização de

componentes, economizando tempo e esforço na criação de funcionalidades adicionais, ao mesmo tempo que mantém a consistência do sistema [12].

A **manutenibilidade** ou capacidade de manutenção é outro ponto de destaque. A independência dos módulos permite que atualizações, correções de bugs e aprimoramentos sejam realizados em partes específicas do sistema, sem a necessidade de interromper todo o sistema. Isso resulta em menos tempo de inatividade e melhora a experiência do usuário final.

A arquitetura de *micro-frontends* também **aumenta a resiliência do sistema**. Falhas em um módulo não comprometem o funcionamento geral do sistema, o que garante **maior tolerância a erros** e contribui para a confiabilidade do sistema [10].

Outro ponto a ser destacado é a **flexibilidade em suportar diferentes tecnologias** [11]. As equipes têm a liberdade de adotar as ferramentas mais adequadas para suas áreas específicas, o que não apenas melhora a eficiência, mas reduz a complexidade geral do sistema. A **descentralização na tomada de decisões** permite que as equipes gerenciem seus serviços de backend e frontend de maneira ágil, aumentando ainda mais a capacidade de resposta às necessidades do mercado.

Por fim, é fundamental ressaltar que, embora a implementação de *micro-frontends* ofereça benefícios evidentes, também traz desafios. Contudo, as vantagens em termos de escalabilidade, flexibilidade e autonomia das equipes reforçam o potencial dessa arquitetura no desenvolvimento de sistemas web modernos, respondendo de forma clara às indagações levantadas pela QP2 sobre os benefícios obtidos ao utilizar *micro-frontends*. A Figura 4 apresenta um diagrama que sintetiza os principais benefícios da adoção de *micro-frontends* discutidos nesta seção.

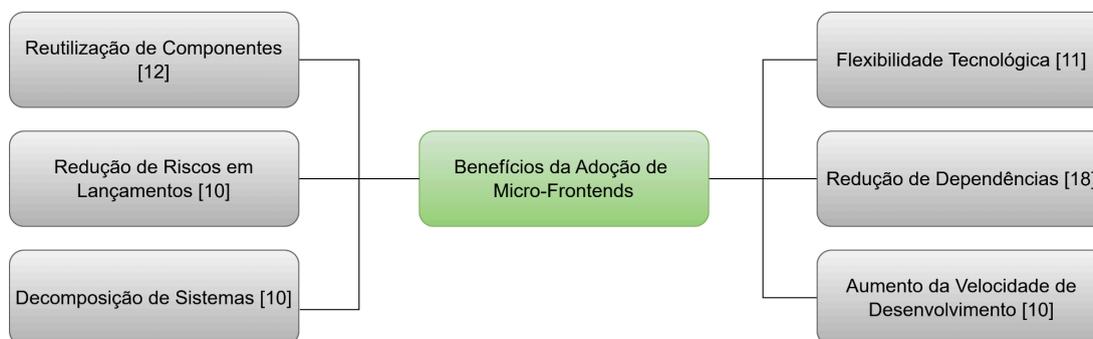


Figura 4. Principais categorias da Q2 sobre benefícios da adoção de *micro-frontends* (Q2).

A Figura 4 destaca como essa abordagem arquitetural contribui para a escalabilidade do sistema, a flexibilidade no uso de tecnologias, a autonomia das equipes e a melhoria na manutenibilidade dos sistemas. Em adição, evidencia a maior resiliência e a redução da complexidade associada aos sistemas monolíticos, reforçando as vantagens obtidas ao adotar essa estratégia.

4.3. Problemas Associados aos *Micro-Frontends* (Q3)

Apesar dos diversos benefícios, a adoção de *micro-frontends* também apresenta alguns desafios e problemas que devem ser considerados pelas empresas. Este capítulo explora

essas adversidades ao responder à QP3, apresentando os principais obstáculos técnicos e organizacionais que podem surgir.

Um dos maiores desafios na implementação de *micro-frontends* é a **orquestração e o gerenciamento adequado dos diferentes módulos** em um único sistema [6]. A complexidade de carregar múltiplos *micro-frontends* de maneira eficiente, garantindo que suas dependências sejam adequadamente geridas e evitando sobrecarga desnecessária, é uma questão crítica. O **roteamento entre esses serviços** é outro problema significativo, pois é necessário definir regras claras para garantir uma navegação fluida e sem conflitos, assegurando que o usuário seja corretamente redirecionado ao *micro-frontend* adequado [18].

Outro ponto de atenção é o **isolamento dos *micro-frontends***. Para evitar conflitos de estilos e comportamento entre módulos é preciso considerar com cuidado as dependências entre os diferentes frameworks utilizados pelas equipes. Em adição, a **comunicação eficiente entre *micro-frontends*** é fundamental para garantir a integridade do sistema. Essa comunicação pode ser complexa e deve ser tratada com soluções como eventos personalizados ou o uso de bibliotecas de gerenciamento de estado compartilhado [1].

Manter a consistência da interface e da experiência do usuário (UI/UX) em todo o sistema também é um grande desafio. A ausência de uma gestão adequada pode resultar em inconsistências visuais e de usabilidade entre os diferentes módulos. Para mitigar esses problemas, o uso de diretrizes de estilo e bibliotecas de componentes compartilhadas pode ser eficaz. O gerenciamento de dependências se mostra crucial para evitar duplicações e garantir que as bibliotecas sejam carregadas de maneira eficiente [6].

A arquitetura de *micro-frontends* pode aumentar significativamente o **esforço de desenvolvimento e manutenção** [12]. Sua complexidade pode exigir **mais tempo e recursos para que as equipes** se familiarizem com novas ferramentas e tecnologias, o que eleva a curva de aprendizado [10]. Outro aspecto que não pode ser negligenciado é a **compatibilidade entre navegadores**, que deve ser garantida mesmo com o uso de recursos mais recentes [6].

Do ponto de vista organizacional, **a governança é um ponto crítico na adoção de *micro-frontends***. Com múltiplas equipes trabalhando de forma independente, é fundamental definir responsabilidades claras e processos de tomada de decisão eficientes para evitar conflitos e garantir a integração harmoniosa dos módulos. O **fenômeno das "ilhas de conhecimento"**, em que equipes desenvolvem seus *micro-frontends* isoladamente, também pode gerar fragmentação do conhecimento, dificultando a coesão e a integração do sistema como um todo [10].

Por fim, **questões de desempenho**, como o **aumento do tamanho da carga útil** e a **duplicação de código entre os módulos**, também devem ser levadas em consideração. A **independência dos *micro-frontends* pode resultar em código redundante**, o que afeta o tempo de carregamento do sistema e, conseqüentemente, a experiência do usuário. Além disso, o **gerenciamento eficaz de dependências compartilhadas é crucial** para evitar conflitos e otimizar o desempenho [6].

A Figura 5 ilustra os principais problemas com desafios e limitações associados à adoção da arquitetura de *micro-frontends*, conforme discutido nesta seção. Observando a Figura 5, é possível identificar problemas técnicos, como a inconsistência de UI/UX, e organizacionais, como a dificuldade de coordenação entre equipes.



Figura 5. Principais categorias da Q3 sobre problemas associados aos *micro-frontends*.

4.4. Síntese dos principais resultados

A análise dos estudos (Seção 4.3) sobre *micro-frontends* revelou um cenário complexo e dinâmico, marcado por avanços e desafios na área. A Figura 6 apresenta uma visão integrada desses principais achados. A adoção dessa arquitetura é impulsionada principalmente pela necessidade de modularidade de sistemas de software web modernos, com impactos em escalabilidade, flexibilidade e autonomia das equipes de desenvolvimento. No entanto, a implementação de *micro-frontends* não é isenta de obstáculos, exigindo atenção a diversos aspectos técnicos e organizacionais que variam conforme as decisões do time e a experiência das pessoas envolvidas.

Principais Resultados				
Motivadores	Benefícios	Desafios	Fatores para Sucesso	Tendências Futuras
Modularidade	Reutilização de Componentes	Orquestração de Módulos	Experiência das Equipes	Adoção Crescente
Necessidade de Escalabilidade	Redução de Riscos em Lançamentos	Consistência da UI/UX	Maturidade de Processos	Evolução de Ferramentas
Autonomia de Equipes	Velocidade de Desenvolvimento	Governança Fraca	Governança Eficaz	Consolidação de Boas Práticas
Integração de Tecnologias Heterogêneas	Flexibilidade Tecnológica	Curva de Aprendizado	Boas Práticas de Desenvolvimento	

Figura 6. Síntese dos principais resultados da análise sobre *micro-frontends*.

Um dos principais benefícios identificados é a capacidade de **escalar** os sistemas de software web modernos de forma **modular** para que diferentes times trabalhem de maneira independente em partes específicas do sistema [19]. Essa abordagem é particularmente vantajosa em grandes projetos com equipes distribuídas e requisitos em constante evolução [11].

A **modularidade** promovida pelos *micro-frontends* não apenas facilita a reutilização de componentes, mas também oferece flexibilidade tecnológica, permitindo que cada equipe escolha as ferramentas adequadas para suas necessidades. Entretanto, essa **flexibilidade** traz consigo desafios significativos, como a orquestração de múltiplos módulos e a gestão de dependências, que exigem um planejamento cuidadoso e a adoção de melhores práticas de desenvolvimento de software, como o isolamento de dependências e a adoção de bibliotecas de componentes [15].

A **governança** eficaz é outro aspecto crítico para o sucesso da adoção de *micro-frontends*. Com múltiplas equipes trabalhando de forma independente, é essencial estabelecer processos claros de tomada de decisão e garantir a consistência da interface do usuário. A falta de uma governança adequada pode resultar em inconsistências visuais e de usabilidade, afetando diretamente a experiência do usuário. Deste modo, a fragmentação de sistemas de software web modernos pode levar a problemas de desempenho, como atrasos na renderização de módulos, caso não sejam adotadas técnicas como o carregamento progressivo e a otimização de dependências [8].

A **experiência das equipes** e a maturidade dos processos de desenvolvimento também desempenham um papel crucial na implementação bem-sucedida de *micro-frontends*. Equipes com maior experiência em arquiteturas modulares e ferramentas modernas tendem a enfrentar menos dificuldades na adoção dessa abordagem. Por outro lado, organizações com processos menos maduros ou equipes menos experientes podem encontrar desafios adicionais, como a curva de aprendizado associada a novas tecnologias e a complexidade na integração de diferentes módulos [10].

As **tendências futuras** apontam para uma crescente adoção de *micro-frontends*, impulsionada pela evolução das ferramentas e frameworks disponíveis. Espera-se que a maturidade dessa arquitetura contribua para a resolução dos desafios atuais, como a comunicação entre módulos e a garantia de consistência da interface. Acredita-se que a experiência acumulada pelas equipes e a adoção de práticas de governança eficazes serão fatores determinantes para o sucesso da implementação [15].

Em resumo, a arquitetura de *micro-frontends* apresenta um grande potencial para o desenvolvimento de sistemas de software web modernos, mas sua adoção exige uma avaliação cuidadosa dos *trade-offs* envolvidos. A decisão de adotar essa arquitetura deve ser baseada nas necessidades específicas do projeto e na capacidade da organização de lidar com os desafios técnicos e organizacionais associados. A experiência das equipes, a maturidade dos processos e as decisões tomadas durante a implementação são fatores críticos que influenciam diretamente o sucesso da adoção de *micro-frontends*.

4.5. Ameaças à Validade

Foram identificadas algumas ameaças à validade que podem afetar a generalização e a confiabilidade deste trabalho. Essas ameaças estão relacionadas a aspectos metodológicos, técnicos e organizacionais, e devem ser consideradas ao interpretar as conclusões deste trabalho.

Uma das principais ameaças à **validade interna** é o viés de seleção dos estudos incluídos na revisão. Embora tenham sido utilizados critérios claros de inclusão e exclusão, é possível que alguns trabalhos relevantes tenham sido excluídos inadvertidamente, especialmente aqueles que não utilizam explicitamente o termo "*micro-frontend*" em seus títulos ou resumos. Para mitigar esse risco, foram consultadas múltiplas bases digitais.

A **validade externa** também pode ser comprometida pela restrição temporal aos estudos publicados entre 2020 e 2024. Embora essa restrição tenha sido necessária para garantir a relevância dos dados, ela pode ter excluído trabalhos importantes publicados antes desse período. A maioria dos estudos selecionados concentra-se em grandes organizações com equipes distribuídas, o que pode limitar a aplicabilidade dos resultados em projetos menores ou com equipes menos experientes. Para reduzir esse impacto, futuras pesquisas poderiam incluir uma gama mais ampla de contextos e escalas de projetos.

A **validade de construto** pode ser impactada pela variação nas definições de *micro-frontends* entre os estudos. Apesar do cuidado na seleção dos trabalhos mais relevantes, diferenças na interpretação do conceito podem ter influenciado a análise. Para mitigar esse problema, foi considerada uma busca abrangente em diversas perspectivas sobre o tema.

Por fim, a **validade de conclusão** pode ser afetada pela natureza qualitativa da análise. A interpretação dos resultados envolve subjetividade, especialmente em revisões que dependem da análise dos pesquisadores. Esse aspecto limita a generalização dos achados e reforça a importância de futuras validações por outros estudos.

Em síntese, embora tenham sido tomadas medidas para mitigar as ameaças à validade, é importante reconhecer que algumas limitações podem influenciar a generalização e a aplicabilidade dos resultados. Pesquisas futuras podem abordar essas limitações ao incluir uma gama mais ampla de estudos, contextos e métricas quantitativas para complementar a análise qualitativa.

5. Conclusão

A presente pesquisa, ao explorar a arquitetura de *micro-frontends*, descobriu um cenário promissor para o desenvolvimento de sistemas de software web modernos. A modularidade e a escalabilidade proporcionadas por essa arquitetura permitem que as equipes de desenvolvimento trabalhem de forma mais autônoma e eficiente, adaptando-se rapidamente às mudanças do mercado. No entanto, como destacado na síntese dos resultados, a implementação de *micro-frontends* exige um planejamento

cuidadoso, boa governança, considerando os desafios relacionados à orquestração de módulos, comunicação entre equipes, consistência da interface e desempenho.

A análise dos estudos revelou que a principal motivação para a adoção de *micro-frontends* é a necessidade de escalar sistemas de software web modernos complexos e promover a autonomia das equipes de desenvolvimento. Neste contexto, a flexibilidade e a reutilização de componentes são benefícios significativos que tornam essa arquitetura atraente para projetos de grande escala. Entretanto, como discutido na seção de ameaças à validade, a complexidade técnica e organizacional, associada aos *micro-frontends*, exigem a adoção de ferramentas e frameworks mais robustos, além de uma governança eficaz para garantir o sucesso da implementação.

As ameaças à validade identificadas, como o viés de seleção e a restrição temporal aos estudos publicados entre 2020 e 2024, podem limitar a generalização dos resultados. Todavia, as medidas adotadas para mitigar essas limitações, como a consulta a múltiplas bases de dados, aumentam a confiabilidade das conclusões. Ainda assim, é importante reconhecer que os resultados podem não ser aplicáveis a todos os contextos, especialmente em projetos menores ou com equipes menos experientes.

Trabalhos futuros podem se concentrar em avaliar o impacto dos *micro-frontends* na experiência do usuário, analisar o custo total de propriedade dessa arquitetura e desenvolver métricas para medir o sucesso da implementação [6]. É fundamental investigar as melhores práticas para a governança e a comunicação entre as equipes em projetos que adotam *micro-frontends* [12]. Outro fator que pode ajudar em futuras pesquisas é a criação de repositórios abertos de projetos que usam *micro-frontends*. Esses estudos podem contribuir para a maturidade da arquitetura, ajudando a resolver os desafios atuais e a explorar novas oportunidades.

Por fim, a arquitetura de *micro-frontends* representa uma abordagem inovadora para o desenvolvimento de sistemas de software web modernos, oferecendo diversas vantagens em termos de modularidade, escalabilidade e flexibilidade. A adoção da arquitetura de *micro-frontends* exige atenção ao planejamento e considerar desafios técnicos e organizacionais envolvidos. As pesquisas futuras devem continuar explorando o potencial dessa arquitetura, desenvolvendo soluções para os desafios existentes e ampliando o entendimento sobre *micro-frontends* em contextos diferentes.

Textos selecionados

- [S1] PELTONEN, S.; MEZZALIRA, L.; TAIBI, D. Motivations, benefits, and issues for adopting Micro-Frontends: A Multivocal Literature Review. *Information and software technology*, v. 136, n. 106571, p. 106571, 2021.
- [S2] TAIBI, D.; MEZZALIRA, L. Micro-Frontends: Principles, Implementations, and Pitfalls. *ACM SIGSOFT Software Engineering Notes*, v. 47, n. 4, p. 25–29, 28 set. 2022.
- [S3] KUNŠTNÁR, V.; PODHORSKÝ, P. Micro Frontend Architecture. 2024 Zooming Innovation in Consumer Technologies Conference (ZINC), p. 124–129, 22 maio 2024.

- [S4] JOUNI MÄNNISTÖ; ANTTI-PEKKA TUOVINEN; MIKKO RAATIKAINEN. Experiences on a Frameworkless Micro-Frontend Architecture in a Small Organization. 2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C), 13 mar. 2023.
- [S5] GASHI, E.; HYSENI, D.; SHABANI, I.; BETIM ÇIÇO. The advantages of Micro-Frontend architecture for developing web application. 2024 13th Mediterranean Conference on Embedded Computing (MECO), 11 jun. 2024.
- [S6] ZHANG, C.; ZHANG, D.; ZHOU, H.; WANG, X.; LIANG LV; ZHANG, R.; XIE, Y.; LIU, H.; LI, Y.; JIA, D.; HUANG, Y.; JIANG, T. Application Business Information Interaction Bus Based on Micro Frontend Framework. 2023 7th International Symposium on Computer Science and Intelligent Control (ISCSIC), v. 02, p. 306–310, 27 out. 2023.
- [S7] PERLIN, R.; EBLING, D.; MARAN, V.; DESCOVI, G.; MACHADO, A. An Approach to Follow Microservices Principles in Frontend. 2023 IEEE 17th International Conference on Application of Information and Communication Technologies (AICT), v. 2, p. 1–6, 18 out. 2023.
- [S8] SHAKIL, M.; ZOITL, A. Towards a Modular Architecture for Industrial HMIs. 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 8 set. 2020.
- [S9] SIMÕES, B.; CARRETERO, M.; MARTÍNEZ, J.; MUÑOZ, S.; ALCAIN, N. Implementing Digital Twins via micro-frontends, micro-services, and web 3D. *Computers & Graphics*, v. 121, p. 103946–103946, 24 maio 2024.
- [S10] WANG, D.; YANG, D.; ZHOU, H.; WANG, Y.; HONG, D.; DONG, Q.; SONG, S. A Novel Application of Educational Management Information System based on Micro Frontends. *Procedia Computer Science*, v. 176, p. 1567–1576, 2020.
- [S11] KAUSHIK, N.; KUMAR, H.; RAJ, V. Micro Frontend Based Performance Improvement and Prediction for Microservices Using Machine Learning. *Journal of Grid Computing*, v. 22, n. 2, 24 abr. 2024.
- [S12] MORAES, F.; CAMPOS, G.; ALMEIDA, N.; AFFONSO, F. Micro Frontend-Based Development: Concepts, Motivations, Implementation Principles, and an Experience Report. *Proceedings of the 26th International Conference on Enterprise Information Systems*, v. 2, p. 175–184, 28 abr. 2024.
- [S13] CAPDEPON, Q.; HLAD, N.; SERIAI, A.-D.; DERRAS, M. Migration Process from Monolithic to Micro Frontend Architecture in Mobile Applications. [s.l.] IWST 2023: International Workshop on Smalltalk Technologies, [s.d.]. Disponível em: <<https://ceur-ws.org/Vol-3627/paper05.pdf>>. Acesso em: jul. 2024.
- [S14] ZHOU, J.; YANG, L.; WU, J. Micro-frontend architecture base. Sixth International Conference on Computer Information Science and Application Technology (CISAT 2023), v. 96, p. 37, 11 out. 2023.
- [S15] PETCU, A.; FRUNZETE, M.; STOICHESCU, D. A. Benefits, challenges, and performance analysis of a scalable web architecture based on micro-frontends. Disponível em:

<https://www.scientificbulletin.upb.ro/rev_docs_arhiva/full454_102480.pdf>. Acesso em: 31 oct. 2024.

- [S16] ABDELFATTAH, A.; CERNY, T. Filling the gaps in microservice frontend communication: Case for new frontend patterns. Proceedings of the 13th International Conference on Cloud Computing and Services Science. Anais...SCITEPRESS - Science and Technology Publications, 2023.
- [S17] NISHIZU, Y.; KAMINA, T. Implementing micro frontends using signal-based web components. Journal of information processing, v. 30, n. 0, p. 505–512, 2022.
- [S18] PÖLÖSKEI, I.; BUB, U. Enterprise-level migration to micro frontends in a multi-vendor environment. Acta polytechnica Hungarica, 2021.
- [S19] TILAK, P. Y.; YADAV, V.; DHARMENDRA, S. D.; BOLLOJU, N. A platform for enhancing application developer productivity using microservices and micro-frontends. 2020 IEEE-HYDCON. Anais...IEEE, 2020.
- [S20] PAVLENKO, A.; ASKARBEEKULY, N.; MEGHA, S.; MAZZARA, M. Micro-frontends: application of microservices to web front-ends. Journal of internet services and information security, v. 10, n. 2, p. 49–66, 2020.
- [S21] MARCO, V.; FARIAS, K. Exploring the technologies and architectures used to develop micro-frontend applications: A systematic mapping and emerging perspectives. SSRN Electronic Journal, 2024.
- [S22] STEPANOV, O.; KLYM, H. Features of the implementation of micro-interfaces in information systems. Advances in Cyber-Physical Systems, 2024.
- [S23] SAVANI, N. The Future of Web Development: An In-depth Analysis of Micro-Frontend Approaches. International Journal of Computer Trends and Technology, v. 71, n. 11, p. 65-69, 2023.
- [S24] TABATADZE, B.; ASANIDZE, G. Synthesis of contemporary approaches used in the development of the client-side in technological projects. Globalization and Business, v. 15, n. 15, p. 49–53, 2023.
- [S25] PRAJWAL, Y.; PAREKH, J.; RAJASHREE SHETTAR A Brief Review of Micro-frontends. Disponível em: <<https://uijrt.com/articles/v2/i8/UIJRTV2I80017.pdf>>. Acesso em: 02 nov. 2024.
- [S26] GEREMIA, K.; WIEST, R.; LUIS, J.; JUNIOR; FONTANA, F.; LOPES, M.; LAZZARETTI, A.; KOLLING, R. Proposta de arquitetura baseada em micro-frontends para multi aplicações. OBSERVATÓRIO DE LA ECONOMÍA LATINOAMERICANA, v. 22, n. 4, p. e4257, 2024.
- [S27] ASKO, A.; URBANIEC, B.; GALATAS, E.; KUSTRA, A.; RODERICK, C.; DEGHAJE, S. Micro frontends - a new migration process for monolithic web applications. Proceedings of ICALEPCS 2023 JACoW Publishing, Geneva, Switzerland, 28 fev. 2024.
- [S28] SHUKLA, A. An Innovative Solution to Manage Projects using Different Javascript Frameworks. Journal of Artificial Intelligence & Cloud Computing, v. 2, n. 2, p. 1-3, 2023.

- [S29] BATISTA, M. V.; PINTO, G. S. ARQUITETURA DE MICROFRONTEND. Revista Interface Tecnológica, v. 20, n. 2, p. 208–218, 2024.
- [S30] DHARMA, B.; ANGGRAINI, D. A Development of Modern Web Application Frontend Structures Using Micro Frontends. International Research Journal of Advanced Engineering and Science, v. 7, n. 1, p. 149-155, 2022.
- [S31] TOSIC, M.; PETROVIC, N.; TOSIC, O. Semantic micro-front-end approach to enterprise knowledge graph applications development. Proceedings of the 19th International Conference on Web Information Systems and Technologies. Anais...SCITEPRESS - Science and Technology Publications, 2023.
- [S32] ANTUNES, F.; JULIA, M.; ANTÔNIO, M.; TAIBI, D.; KALINOWSKI, M. Investigating benefits and limitations of migrating to a micro-frontends architecture. Anais do XXXVIII Simpósio Brasileiro de Engenharia de Software (SBES 2024).
- [S33] NIKULINA, O.; KHATSKO, K. Method of converting the monolithic architecture of a front-end application to microfrontends. Bulletin of National Technical University “KhPI”. Series: System Analysis, Control and Information Technologies, n. 2 (10), p. 79–84, 2023.
- [S34] STERL, S.; BILLIG, A.; TAFFO, F.W.; SVOBODA, D.; GERHOLD, L. Perceiving and behaving in a crisis: Developing a multi-Functional Crisis Information Platform for Psychosocial Situations (CIP-PS). Proceedings of the International ISCRAM Conference, 2024.
- [S35] CHERIPURATHU, K. G.; KULKARNI, S. Integrating microservices and microfrontends: A comprehensive literature review on architecture, design patterns, and implementation challenges. Journal of Scientific Research and Technology, p. 1–12, 2024.

Referências

- [1] PETCU, A.; FRUNZETE, M.; STOICHESCU, D. A. Benefits, challenges, and performance analysis of a scalable web architecture based on micro-frontends. Disponível em: <https://www.scientificbulletin.upb.ro/rev_docs_arhiva/full454_102480.pdf>. Acesso em: 31 oct. 2024.
- [2] KUNŠTNÁR, V.; PODHORSKÝ, P. Micro Frontend Architecture. 2024 Zooming Innovation in Consumer Technologies Conference (ZINC), p. 124–129, 22 maio 2024.
- [3] PÖLÖSKEI, I.; BUB, U. Enterprise-level migration to micro frontends in a multi-vendor environment. Acta polytechnica Hungarica, 2021.
- [4] MORAES, F.; CAMPOS, G.; ALMEIDA, N.; AFFONSO, F. Micro Frontend-Based Development: Concepts, Motivations, Implementation Principles, and an Experience Report. Proceedings of the 26th International Conference on Enterprise Information Systems, v. 2, p. 175–184, 28 abr. 2024.
- [5] PERLIN, R.; EBLING, D.; MARAN, V.; DESCOVI, G.; MACHADO, A. An Approach to Follow Microservices Principles in Frontend. 2023 IEEE 17th

- International Conference on Application of Information and Communication Technologies (AICT), v. 2, p. 1–6, 18 out. 2023.
- [6] ANTUNES, F.; JULIA, M.; ANTÔNIO, M.; TAIBI, D.; KALINOWSKI, M. Investigating benefits and limitations of migrating to a micro-frontends architecture. *Anais do XXXVIII Simpósio Brasileiro de Engenharia de Software (SBES 2024)*.
- [7] SIMÕES, B.; CARRETERO, M.; MARTÍNEZ, J.; MUÑOZ, S.; ALCAIN, N. Implementing Digital Twins via micro-frontends, micro-services, and web 3D. *Computers & Graphics*, v. 121, p. 103946–103946, 24 maio 2024.
- [8] PELTONEN, S.; MEZZALIRA, L.; TAIBI, D. Motivations, benefits, and issues for adopting Micro-Frontends: A Multivocal Literature Review. *Information and software technology*, v. 136, n. 106571, p. 106571, 2021.
- [9] CHERIPURATHU, K. G.; KULKARNI, S. Integrating microservices and microfrontends: A comprehensive literature review on architecture, design patterns, and implementation challenges. *Journal of Scientific Research and Technology*, p. 1–12, 2024.
- [10] MARCO, V.; FARIAS, K. Exploring the technologies and architectures used to develop micro-frontend applications: A systematic mapping and emerging perspectives. *SSRN Electronic Journal*, 2024.
- [11] PRAJWAL, Y.; PAREKH, J.; RAJASHREE SHETTAR A Brief Review of Micro-frontends. Disponível em: <<https://uijrt.com/articles/v2/i8/UIJRTV2I80017.pdf>>. Acesso em: 02 nov. 2024.
- [12] BATISTA, M. V.; PINTO, G. S. ARQUITETURA DE MICROFRONTEND. *Revista Interface Tecnológica*, v. 20, n. 2, p. 208–218, 2024.
- [13] WANG, D.; YANG, D.; ZHOU, H.; WANG, Y.; HONG, D.; DONG, Q.; SONG, S. A Novel Application of Educational Management Information System based on Micro Frontends. *Procedia Computer Science*, v. 176, p. 1567–1576, 2020.
- [14] RICHARDS, M.; FORD, N. *Fundamentals of Software Architecture: An Engineering Approach*. 2020.
- [15] PAVLENKO, A.; ASKARBEEKULY, N.; MEGHA, S.; MAZZARA, M. Micro-frontends: application of microservices to web front-ends. *Journal of internet services and information security*, v. 10, n. 2, p. 49–66, 2020.
- [16] TAIBI, D.; MEZZALIRA, L. Micro-Frontends: Principles, Implementations, and Pitfalls. *ACM SIGSOFT Software Engineering Notes*, v. 47, n. 4, p. 25–29, 28 set. 2022.
- [17] CARTAXO, B.; PINTO, G.; SOARES, S. *Rapid Reviews in Software Engineering*. Springer eBooks, p. 357–384, 1 jan. 2020.
- [18] STEPANOV, O.; KLYM, H. Features of the implementation of micro-interfaces in information systems. *Advances in Cyber-Physical Systems*, 2024.

[19] GASHI, E.; HYSENI, D.; SHABANI, I.; BETIM ÇIÇO. The advantages of Micro-Frontend architecture for developing web application. 2024 13th Mediterranean Conference on Embedded Computing (MECO), 11 jun. 2024.