

Exploring the technologies and architectures used to develop micro-frontend applications: A systematic mapping and emerging perspectives

Vítor Adriano Marco
vitoradrianomarco@gmail.com
Universidade do Vale do Rio dos Sinos
São Leopoldo, Rio Grande do Sul, Brazil

Kleinner Farias
kleinnerfarias@unisinos.br
Universidade do Vale do Rio dos Sinos
São Leopoldo, Rio Grande do Sul, Brazil

ABSTRACT

Context: Micro-frontend architecture in software development seeks to establish a robust framework for implementing user interfaces. This modular approach enables independent creation, deployment and scalability of each project. Despite the relevance of this field for large-scale projects in industry, there are still few studies that classify and examine this architectural technique. **Objective:** This work, therefore, offers a comprehensive overview of this architecture, highlighting gaps, challenges and research trends. **Method:** A systematic mapping of the literature was developed and executed, following already established practical guidelines. In total, 25 primary studies were selected, analyzed and categorized, after applying a rigorous filtering process to a sample of 5,308 candidate studies, with the aim of answering nine research questions. **Results:** More than 75% of the selected studies adopted Javascript as a technology; more than 30% adopted micro-frontend as their architecture; More than 35% adopted ReactJS as a framework/library; 25% of studies use the web components pattern; More than 20% of studies perceive modularization and isolation as reasons for micro-frontend adoption. **Conclusions:** The study benefits professionals and researchers, providing a set of knowledge about current literature, being a starting point for future studies. The article reports challenges that deserve investigation, exploring the topic of how to measure micro-frontends and how they are perceived after implementation.

KEYWORDS

Micro-frontend; Sistematic Mapping Study; Enterprise Applications; Literature Review

1 INTRODUÇÃO

No cenário atual, o desenvolvimento de sistemas de software ocorre em ambientes de negócios cada vez mais instáveis na indústria, caracterizados por requisitos complexos e voláteis, pressões por ciclos de desenvolvimento mais curtos e evolução rápida de tecnologias e modelos arquiteturais [40, 51]. A arquitetura de software desempenha um papel crucial no desenvolvimento e manutenção desses sistemas, sendo que a evolução de uma aplicação pode resultar em degradação arquitetural, afetando a qualidade do sistema e encurtando sua vida útil [9, 18, 19]. Para lidar com esses desafios, algumas abordagens de arquitetura, como microservices e micro-frontend, surgiram para mitigar a complexidade de grandes monólitos [60].

A arquitetura de micro-frontend assume atualmente um papel fundamental na indústria devido à crescente complexidade das aplicações web e à demanda por evolução contínua dos sistemas. Esta

abordagem de design modifica a concepção de aplicativos front-end ao modularizá-los em microaplicativos independentes, promovendo assim a colaboração harmoniosa entre eles. A estrutura arquitetural em questão favorece maior modularidade, reutilização de componentes e facilita a manutenção, uma vez que cada microaplicativo pode ser construído, implantado e escalado de forma independente. Adicionalmente, proporciona agilidade no desenvolvimento, aprimorando a experiência do usuário por meio de uma navegação mais ágil e responsiva. Destaca-se, ainda, como uma abordagem essencial para a construção de sistemas front-end robustos, flexíveis e escaláveis

A literatura atual, no entanto, apresenta uma lacuna notável na ausência de um mapeamento sistemático abrangente sobre arquitetura de micro-frontends. Apesar do interesse crescente e da rápida evolução nesse campo, a falta de um levantamento detalhado e estruturado das publicações existentes sobre essa temática é evidente. Consequentemente, os desenvolvedores acabam adotando arquiteturas sem um conhecimento amplo e detalhado. Profissionais e pesquisadores enfrentam o desafio de escolher entre as abordagens disponíveis aquela que melhor atende às suas necessidades. A realização de um mapeamento sistemático da literatura em arquitetura de micro-frontends se mostra necessária para consolidar o conhecimento existente, identificar lacunas de pesquisa, destacar tendências emergentes e fornecer uma base conceitual para futuras investigações e desenvolvimentos nesse campo em constante evolução. Além disso, é importante destacar que revisões recentes da literatura abordam aspectos específicos da arquitetura de micro-frontend, mas não oferecem um mapeamento sistemático ou uma visão abrangente das tendências e direções de pesquisa. [26, 42, 43, 46, 58].

Este artigo preenche essa lacuna ao apresentar uma visão geral abrangente da literatura (Seção 6), identificando lacunas, desafios e tendências (Seção 7). Para alcançar esse objetivo, foi realizado um mapeamento sistemático da literatura com base em diretrizes estabelecidas [4, 28, 38, 50]. Foram analisados 25 estudos, selecionados após a aplicação de um rigoroso processo de filtragem em uma amostra inicial de 5308 estudos, para responder a nove perguntas de pesquisa (QPs). O mapeamento sistemático da literatura foi escolhido como método de pesquisa devido à sua capacidade de proporcionar uma visão ampla da área, resumir a literatura existente e destacar desafios e direções futuras [5, 30, 47]. As principais contribuições deste artigo incluem: (i) um corpo de conhecimento através de um mapa sistemático sobre a literatura atual, que serve como ponto de partida para futuras pesquisas; (ii) um protocolo de revisão reutilizável em pesquisas futuras; e (iii) uma avaliação

abrangente da qualidade dos estudos atuais, reduzindo a curva de aprendizado e viés em futuras revisões de literatura.

O restante do artigo está organizado da seguinte forma: a Seção 3 compara este estudo com a literatura atual, a Seção 4 apresenta o processo de pesquisa, a Seção 5 detalha o processo de filtragem, a Seção 6 apresenta os resultados e a Seção 7 introduz discussões e direções futuras.

2 FUNDAMENTAÇÃO TEÓRICA

Esta Seção apresenta os conceitos fundamentais que serão empregados ao longo do trabalho, alinhando-se com a literatura contemporânea existente.

2.1 Arquitetura de micro-frontend

A arquitetura de micro-frontends incorpora os princípios dos microservices na camada de interface, fragmentando uma aplicação monolítica em aplicações independentes. Essa abordagem visa promover a independência de tecnologia, permitindo que cada equipe desenvolva e mantenha suas próprias soluções de forma isolada. Além disso, a escalabilidade é aprimorada, uma vez que cada componente pode ser dimensionado independentemente, otimizando o desempenho em resposta a demandas específicas. A resiliência é fortalecida, uma vez que falhas ou atualizações afetam apenas os micro-frontends pertinentes, minimizando o impacto global e proporcionando uma experiência de usuário mais estável e confiável. [21, 60]

2.2 Mapeamento Sistemático da literatura

O mapeamento sistemático da literatura é um sistema estruturado para identificar, avaliar e interpretar todas as pesquisas relevantes em torno de uma questão de pesquisa específica, área de tópico ou fenômeno de interesse. Sendo assim, fornece uma visão abrangente sobre determinado tema, além de identificar lacunas na pesquisa atual, orientando novas investigações. Tem um processo rigoroso de elaboração da string de busca, critérios de exclusão e inclusão de estudos pertinentes. [30]

3 TRABALHOS RELACIONADOS

A pesquisa pelos trabalhos relacionados foi realizada no repositório digital, *Google Scholar*. Grande parte dos trabalhos selecionados foram resultados de pesquisas pelo termo “front-end architecture AND micro-frontend”. A metodologia adotada para analisar os trabalhos relacionados tem sido previamente validada em estudo publicados na literatura [10, 11, 22, 23, 33, 36]

3.1 Análise dos trabalhos relacionados

[43]. Apresenta um survey com o objetivo de identificar as motivações, benefícios e questões relacionadas à adoção de *micro-frontends*. A implementação de micro-frontend em uma arquitetura flexível e escalável para atender às necessidades de sistemas modernos. Os benefícios da adoção de micro-frontends incluem independência de equipe, melhoria da modularidade e reutilização de componentes, enquanto as questões apontadas incluem complexidade de gerenciamento e sobrecarga de comunicação.

[42]. Apresenta uma abordagem de micro-frontends como uma aplicação de microservices para o front-end da web. Ele discute

como essa arquitetura pode ajudar a superar desafios relacionados à complexidade e ao escalonamento dos sistemas front-end, além de melhorar a colaboração entre as equipes de desenvolvimento. São apresentados exemplos de implementações bem-sucedidas e discutidas as vantagens e desvantagens dessa abordagem.

[46]. Apresenta uma breve revisão da literatura sobre micro-frontends, que aborda a implementação de microservices no front-end de aplicações web. O autor discute os conceitos básicos e as vantagens, tais como escalabilidade, desacoplamento, reutilização de componentes e independência de equipe, além de discutir alguns desafios como a complexidade de gerenciamento, segurança e compatibilidade com navegadores. Por fim, são mencionados alguns desafios relacionados à adoção de micro-frontends, como a necessidade de gerenciar múltiplos ambientes de execução.

[58]. Apresenta uma study mapping sobre as diferentes abordagens de implementação de micro-frontends, incluindo Web Components e Single Page Application (SPA), e discute as vantagens e desvantagens de cada uma delas. Também são discutidos os desafios enfrentados na adoção de micro-frontends, como a complexidade de gerenciamento, segurança e compatibilidade com navegadores.

[26]. Apresenta um conjunto de diretrizes para adoção de arquiteturas e padrões de front-end com base em microservices. São abordados os desafios enfrentados na adoção de micro-frontends no front-end, como a complexidade de gerenciamento e a necessidade de garantir a coesão entre as equipes de desenvolvimento. O estudo fornece exemplos de padrões de arquitetura de front-end, como a arquitetura baseada em componentes e a arquitetura em camadas.

3.2 Análise comparativa dos trabalhos relacionados

A análise de pesquisas anteriores, apresentada na Tabela 1, pode ajudar a identificar lacunas no conhecimento, identificar tendências e abordagens promissoras e fornecer uma base sólida para a pesquisa atual.

4 METODOLOGIA

Este artigo adota uma metodologia robusta baseada em um estudo de mapeamento sistemático, que vai além da mera apresentação dos resultados e busca abranger uma variedade de aspectos relacionados a eles. Essa abordagem proporciona uma compreensão mais abrangente do campo de estudo, incluindo informações sobre a localização da publicação, a área de estudo, a distribuição dos artigos entre diferentes bancos de dados e a evolução do número de estudos ao longo do tempo [14]. A metodologia adotada foi baseada em estudos anteriores publicados [13, 56, 61].

O processo de pesquisa segue as diretrizes estabelecidas por [44] e atualizado por [45]. Essas diretrizes envolvem as seguintes etapas:

- **Definição das Questões de Pesquisa:** A primeira etapa envolve a formulação das questões de pesquisa que direcionarão o estudo. Isso permite estabelecer os principais tópicos a serem abordados e fornece uma estrutura para a análise subsequente.
- **Definição do Processo de Busca:** Uma estratégia de busca é elaborada para coletar os estudos relevantes. Isso inclui a identificação de palavras-chave, fontes de pesquisa e

Table 1: Análise comparativa dos Trabalhos Relacionados

Referência	Foco	Método de Pesquisa	Tipo de Arquitetura	Tecnologias
[42]	micro-frontends em single-page applications	Revisão da literatura	micro-frontend em single-page applications	React, Angular, Web Components
[43]	Entender as motivações e benefícios para as companhias ao adotar micro-frontend	Survey	micro-frontend	-
[46]	Visão geral sobre a arquitetura de micro-frontend	Revisão da literatura	micro-frontend	React, Angular e Vue
[58]	Visão geral sobre micro-frontends, forma de implementar e como selecionar a melhor abordagem no desenvolvimento	Estudo de mapeamento	micro-frontend em Application Shell, iFrames, Edge Side Includes, Web Components e Server-Side	Angular, SAP Luigi, Stencil, Tailor, Open Components, Ara e Piro
[26]	Guia para adoção de arquiteturas micro-frontend	Estudo de mapeamento	micro-frontend	-

critérios de seleção. A busca é conduzida de maneira sistemática e abrangente.

- **Definição dos Critérios de Filtragem:** São estabelecidos critérios de inclusão e exclusão para avaliar a relevância dos estudos identificados. Esses critérios garantem que apenas os estudos pertinentes sejam incluídos na análise.
- **Extração de Dados:** As informações relevantes são extraídas dos estudos selecionados, permitindo a compilação de dados que serão posteriormente analisados.
- **Análise e Classificação:** Os dados coletados são analisados e classificados de acordo com os objetivos do estudo. Isso envolve a categorização dos estudos com base em diferentes parâmetros, como ano de publicação, local de publicação e área de estudo.

4.1 Questões de pesquisa

No decorrer deste artigo, são abordadas diversas questões de pesquisa que servem como diretrizes fundamentais para o estudo. As perguntas da pesquisa podem ser agrupadas em três categorias distintas: Questão Geral (QG), Questões Focalizadas (QF) e Questões Estatísticas (QE). Cada uma dessas categorias desempenha um papel fundamental na estruturação e na busca por respostas relevantes neste estudo.

Questão Geral (QG) Inicia-se com a Questão Geral (QG), que tem como objetivo principal a exploração das tecnologias e metodologias contemporâneas no contexto do desenvolvimento de aplicações front-end. Esta questão fornece um panorama amplo e essencial para entender o cenário atual no desenvolvimento de interfaces de usuário, abrangendo aspectos tecnológicos, ferramentas e práticas de desenvolvimento.

Questões Focalizadas (QF) O conjunto de seis perguntas denominadas QFs direciona a investigação para aspectos específicos do desenvolvimento de aplicações front-end. Cada uma dessas questões concentra-se em áreas cruciais, como tipos de arquitetura,

frameworks/bibliotecas, padrões de design, avaliação e mensuração de micro-frontend, percepção dos benefícios e motivações para a adoção dessas abordagens. Essas questões fornecem insights detalhados sobre as práticas e desafios enfrentados no desenvolvimento de front-end, além de destacar os benefícios e as razões subjacentes à escolha de determinadas abordagens.

Questões Estatísticas (QE) Por fim, as Questões Estatísticas (QE) concentram-se em aspectos quantitativos do estudo. Composta por duas perguntas, esta categoria busca entender a distribuição geográfica dos estudos, isto é, onde eles estão sendo conduzidos (QE1), e também examina o número de publicações por ano (QE2). Essas questões estatísticas enriquecem a análise ao fornecer informações importantes sobre as tendências regionais e a evolução temporal das pesquisas relacionadas ao desenvolvimento front-end.

A Tabela 2 apresenta uma visão geral das perguntas de pesquisa abordadas neste artigo, destacando a diversidade e a amplitude do estudo, além de demonstrar o compromisso em obter respostas abrangentes e bem fundamentadas na análise.

4.2 Processo de pesquisa

O processo de pesquisa adotado segue a metodologia desenvolvida por [44] para mapeamento de estudos. Esse processo é composto por três etapas essenciais: especificação da string de busca, seleção dos bancos de dados e coleta dos resultados.

A primeira etapa envolve a elaboração da string de busca, que é fundamental para identificar os estudos relevantes na área de estudo. Nesse contexto, foram selecionados três termos principais que abrangem os principais aspectos do tema em análise: "front-end", "architecture" e "Enterprise Applications". A Tabela 3 apresenta esses termos principais e seus sinônimos mais relevantes. A estratégia de pesquisa é construída por meio da combinação destes termos e sinônimos, utilizando operadores lógicos como disjunção e conjunção para garantir uma abordagem abrangente.

Table 2: Questões de pesquisa

Referência	Pergunta
Questão Geral	Quais são as tecnologias e metodologias atuais para o desenvolvimento de aplicações front-end?
Pergunta focalizada	
FQ1	Quais os tipos de arquitetura estão sendo usadas para desenvolver aplicações front-end?
FQ2	Quais os frameworks/bibliotecas estão sendo utilizados?
FQ3	Quais são patterns para implementar os micro-frontend?
FQ4	Como os micro-frontend são avaliados e mensurados?
FQ5	Como os benefícios de micro-frontend são percebidos?
FQ6	Quais são as motivações para adoção de micro-frontend?
Perguntas estatísticas	
QE1	Onde os estudos estão sendo conduzidos?
QE2	Qual o número de publicações por ano?

Table 3: Termo de string de pesquisa

Termo principal	Termo de pesquisa
front-end	("front end" OR interface OR "graphical user interface" OR UI)
Architecture	("software design" OR pattern OR "software architecture")
Enterprise Applications	("corporate systems" OR "enterprise platforms" OR "enterprise resource planning" OR "enterprise asset management" OR "human capital management")

A segunda etapa consiste na escolha criteriosa dos bancos de dados que serão utilizados na busca dos estudos. A seleção de bancos de dados apropriados é essencial para garantir a cobertura das fontes de informação mais relevantes no contexto da pesquisa. Neste estudo, foram selecionados bancos de dados reconhecidos por sua relevância na área de ciência da computação e tecnologia. A Tabela 4 sintetiza os bancos de dados escolhidos e suas respectivas URLs, fornecendo uma visão clara das fontes de pesquisa utilizadas neste estudo.

A terceira etapa envolve a execução da busca nos bancos de dados selecionados utilizando a string de busca previamente definida. Os resultados da busca são coletados e organizados de forma a possibilitar a análise e seleção dos estudos relevantes para a revisão sistemática. Este processo de coleta é conduzido de maneira sistemática e documentado, assegurando a transparência e reprodutibilidade da pesquisa. A Tabela 5 sintetiza essa execução e os parâmetros iniciais dessa execução em cada base de dados.

4.3 Critérios de Inclusão e Exclusão

O processo de filtragem desempenha um papel fundamental na seleção dos artigos mais relevantes para este estudo, permitindo que sejam retirados do conjunto de estudos aqueles que não se encaixam no escopo da pesquisa. Para orientar esse processo, foram estabelecidos critérios de inclusão e exclusão, fornecendo uma estrutura sólida para a seleção criteriosa dos artigos.

Os Critérios de Inclusão (CI) representam as premissas que orientam a análise dos artigos, incluindo aspectos como o local de publicação e o conteúdo abordado. Por outro lado, os critérios de exclusão (CE) definem as condições que determinam a exclusão de artigos do estudo. A Tabela 6 apresenta de forma detalhada os

critérios de inclusão e exclusão aplicados em cada etapa do processo de filtragem, garantindo a consistência e a objetividade na seleção dos artigos.

Os estudos foram selecionados em duas etapas distintas. Na primeira etapa, foram filtrados com base nos títulos e resumos, usando metadados para direcionar a seleção. Posteriormente, os artigos completos das pesquisas foram incluídos no processo de seleção.

A Figura 1 ilustra o processo de filtragem, destacando os critérios de inclusão (CI) e exclusão (CE) que foram aplicados em cada etapa. A análise contemplou um período que compreende de janeiro de 2000 a maio de 2023, o que inicialmente resultou em um conjunto de 5.308 artigos. Em seguida, os critérios CI1, CI3, CE2 e CE5 foram aplicados, reduzindo o número de artigos para 2.614.

Na segunda fase do processo, os critérios de exclusão CE1 foram empregados, priorizando a análise dos títulos dos artigos. Esse passo reduziu ainda mais a seleção, resultando em 94 estudos conduzidos. Na terceira etapa, os critérios foram estendidos para incluir a análise dos resumos, o que levou à remoção de 48 estudos adicionais. Na quarta etapa, os 46 artigos remanescentes passaram por uma etapa de revisão minuciosa, na qual duplicatas foram identificadas e removidas, resultando em um conjunto final de 44 artigos.

É relevante notar que este estudo aplicou as duas primeiras etapas da abordagem de três etapas introduzida por [29]. O primeiro passo consistiu em uma leitura superficial, envolvendo a análise do título, resumo e prefácio dos artigos. Foi feita uma leitura dos títulos das seções, com o restante do conteúdo sendo ignorado nessa fase inicial, culminando na leitura das conclusões dos trabalhos. O segundo passo incluiu a análise minuciosa das figuras presentes nos textos, com atenção especial dada aos gráficos. Após essa análise preliminar, realizou-se a leitura completa dos trabalhos, o que resultou na

Table 4: Base de dados utilizadas para pesquisa

Base de dados	URL
ACM Digital Library	https://dl.acm.org/
IEEE Xplore Digital	https://ieeexplore.ieee.org/Xplore/home.jsp
ScienceDirect	https://www.sciencedirect.com/
Scopus	https://www.scopus.com/
Google Scholar	https://scholar.google.com/

Table 5: Customização para cada database

Database	Descrição do campo de busca
ACM Digital Library	Pesquisa selecionando o filtro avançado "Custom range - "From jan 2000" e "to may 2023" e filtrando por "Content Type – Research Article" and "Publications – Media Formats – PDF."
IEEE Xplore Digital	Pesquisa selecionando o filtro avançado "Publication Year - "Specify Year Range 2000 - 2023"
Science Direct	Pesquisa selecionando o filtro avançado "year(s) - "2000 - 2023"
Scopus	Pesquisa regular
Google Scholar	Pesquisa regular

Table 6: Critério de Inclusão e Exclusão

Referência	Descrição
Critério de inclusão	
CI1	Trabalhos acadêmicos (ex., artigos, pesquisas, teses de doutorado e mestrado) com foco em arquiteturas de software, implementação enterprise para interfaces gráficas.
CI2	Trabalhos escritos, publicados ou disseminados em inglês.
CI3	Trabalhos completos ou artigos completos
CI4	Estudos publicados de Janeiro de 2000 até Maio de 2023.
Critério de exclusão	
CE1	Exclui artigos cujo título, resumo ou conteúdo não apresentam uma relação semântica significativa com a string de busca, evitando a inclusão de estudos não pertinentes.
CE2	Artigos que não estejam no formato .pdf
CE3	Artigos que não foram publicados em Inglês, ou que possam ser patentes
CE4	Artigos duplicados
CE5	Artigos publicados antes de Janeiro de 2000

seleção dos 25 estudos finais. Por fim, uma leitura aprofundada, focada nas questões abordadas nos trabalhos, confirmou a seleção dos mesmos, totalizando 25 artigos na análise final.

5 FILTRAGEM DOS TRABALHOS

Esta seção descreve em detalhes o o processo de filtragem dos trabalhos, fornecendo uma visão abrangente do método de seleção dos estudos candidatos. A filtragem é composta por sete etapas distintas, nas quais critérios de inclusão e exclusão são aplicados. A Figura 1 visualiza os resultados obtidos em cada uma dessas etapas. Cada passo do processo de filtragem dos estudos é apresentado abaixo:

- **Passo 1: Busca inicial.** A pesquisa teve início com a coleta dos resultados iniciais após submeter a string de busca às bases de dados, conforme especificado na Tabela 5. Esse

estágio inicial resultou em um total de 5.308 estudos candidatos.

- **Passo 2: Remoção de impurezas (CI1, CI3, CE2 e CE3).** Nesta etapa, foram aplicados critérios de inclusão (CI1, CI3) e exclusão (CE2, CE3) para eliminar impurezas na pesquisa. Foram descartados estudos que não possuíam correlação semântica entre títulos, resumos ou não estavam formatados de acordo com o escopo do artigo. Além disso, artigos publicados em idiomas diferentes do inglês foram excluídos. Isso resultou em 2.614 estudos (49,25%) mantidos para a próxima etapa, enquanto 2.694 estudos (50,75%) foram removidos.
- **Passo 3: Filtragem por título.** Nesta etapa, foi realizado uma leitura aprofundada dos títulos dos estudos para remover aqueles que não se enquadravam no escopo da pesquisa. Foram selecionados 94 estudos, enquanto muitos foram

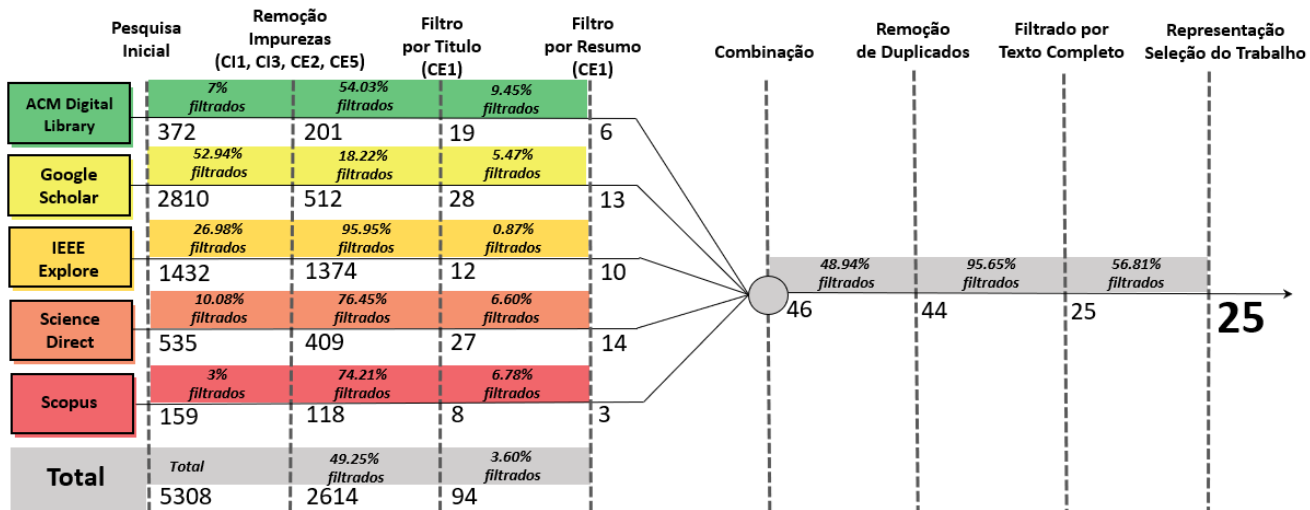


Figure 1: Passo a passo da filtragem de trabalhos

descartados devido a termos irrelevantes, como "front-end em ondas de frequência", que estavam completamente fora do escopo do trabalho.

- **Passo 4: Filtragem por resumo.** Na terceira etapa, foi realizada uma análise inicial dos resumos dos estudos. Foram selecionados estudos com resumos relevantes para uma análise mais detalhada. No total, 46 trabalhos foram mantidos, representando 68,09% dos estudos após a primeira filtragem.
- **Passo 5: Combinação dos trabalhos selecionados.** Nesta etapa, foi agrupado os estudos selecionados de diversas bases de dados, resultando na combinação dos artigos. Não houve alteração na quantidade total de estudos nesta fase.
- **Passo 6: Remoção de duplicados.** Durante essa etapa, foi identificado e removido possíveis estudos duplicados, resultando em um conjunto final de 44 artigos.
- **Passo 7: Filtragem por texto completo.** Na última etapa, foi realizado uma varredura inicial pelos subtítulos, buscando palavras-chave relacionadas ao escopo do projeto. Foram removidos trabalhos que apresentavam foco predominante em arquiteturas de backend em detrimento do front-end. Posteriormente, realizamos uma leitura minuciosa do texto completo dos trabalhos, resultando na seleção de 25 estudos que estavam diretamente relacionados ao escopo do projeto, como indicado na Tabela 7.

Essas etapas de filtragem, estruturadas com cuidado, garantiram que apenas estudos alinhados com os objetivos e o escopo da pesquisa fossem incluídos na análise. A Figura 1 oferece uma representação visual de cada etapa do processo de filtragem, demonstrando a seleção progressiva e a exclusão dos estudos candidatos.

A Tabela 7 apresenta os 25 estudos mantidos após a última etapa de filtragem, fornecendo detalhes sobre os autores, bases de dados de origem, anos de publicação e foco dos trabalhos.

6 RESULTADOS DO ESTUDO

Esta seção apresenta os resultados obtidos, analisa e categoriza os dados obtidos do processo de filtragem. Essa abordagem visa uma organização mais estruturada dos resultados, facilitando a compreensão das relações entre as variáveis, as tendências emergentes e as conclusões no contexto das questões de pesquisa previamente formuladas na Tabela 2.

6.1 QG: Quais são as tecnologias e metodologias atuais para o desenvolvimento de aplicações front-end?

A pesquisa QG busca responder à questão de quais são as tecnologias fundamentais no desenvolvimento de front-end moderno. A Tabela 8 apresenta os resultados relacionados à QG. Uma característica proeminente é que a maioria dos estudos (76%, 19/25) utiliza JavaScript. É importante ressaltar que nessa categoria estão incluídas diversas bibliotecas e frameworks JavaScript, tais como ReactJS (47.37%, 9/19), Angular (26.31%, 5/19), VueJS (10.53%, 2/19), e o restante opta pelo JavaScript "vanilla" (15.79%, 3/19).

Após o JavaScript, tecnologia mais comum nos estudos foi o Webpack, com 48% de adoção (12/25). Esse dado sugere que o amplo uso do Webpack pode ser influenciado pela popularidade das bibliotecas e frameworks modernos de desenvolvimento web, como React, Vue e Angular. A terceira maior ocorrência é destinado para HTML e CSS que seguem empatados, com 44% de adoção cada (11/25). Isso destaca que, mesmo com o surgimento de novas ferramentas e tecnologias, o desenvolvimento web ainda mantém suas bases fundamentais.

Uma observação curiosa, que tecnologias como redux, styled-components, typescript e docker não tiveram um percentual relevante de destaque, fica para um estudo futuro entender o motivo, se foi devido falta de material sobre ou se estão em desuso ou se até não são relevantes para o desenvolvimento front-end.

Table 7: Trabalhos selecionados

ID	Autor	Base de dados	Ano	Foco
A1	[39]	IEEE	2023	micro-frontend
A2	[59]	IEEE	2020	arquitetura
A3	[2]	IEEE	2014	entreprise
A4	[35]	IEEE	2019	arquitetura
A5	[37]	IEEE	2023	aplicação web
A6	[7]	IEEE	2021	aplicação web
A7	[48]	IEEE	2021	aplicação web
A8	[6]	IEEE	2020	aplicação web
A9	[17]	IEEE	2022	reactjs
A10	[63]	Science Direct	2012	arquitetura
A11	[52]	Science Direct	2019	micro-frontend
A12	[54]	Science Direct	2023	erp
A13	[55]	Science Direct	2019	erp
A14	[57]	Science Direct	2014	arquitetura
A15	[49]	Science Direct	2023	arquitetura
A16	[32]	Scholar	2021	micro-frontend
A17	[3]	Scholar	2008	erp
A18	[62]	Scholar	2014	erp
A19	[20]	Scholar	2017	arquitetura
A20	[34]	Scholar	2021	arquitetura
A21	[31]	Scholar	2018	erp
A22	[25]	Scholar	2020	erp
A23	[15]	Scholar	2021	micro-frontend
A24	[12]	Scholar	2022	micro-frontend
A25	[27]	Scholar	2018	arquitetura

Conclusão da QG: A QG identificou de forma sistemática 25 tecnologias na prototipação de front-end moderno. Os resultados destacam que os estudos primários alinham-se com as práticas predominantes no mercado para a criação de interfaces ricas e dinâmicas, com 76% dos casos voltados para JavaScript. Essa observação oferece uma direção valiosa aos profissionais que procuram orientação na seleção de tecnologias para seus projetos, utilizando estudos publicados e validados por trabalhos anteriores. Essa abordagem ajuda a mitigar os riscos relacionados ao emprego de tecnologias ainda em estágio inicial de desenvolvimento.

6.2 FQ1: Quais os tipos de arquitetura estão sendo usadas para desenvolver aplicações front-end?

A pesquisa FQ1 fornece uma visão abrangente das diferentes arquiteturas utilizadas no desenvolvimento de aplicações front-end, com base nos resultados obtidos na busca de estudos primários. Os dados revelam uma variedade de abordagens arquiteturais adotadas por pesquisadores e desenvolvedores.

A maioria dos estudos primários, representando 32% das publicações, está focada na arquitetura de micro-frontends. Isso sugere um interesse crescente nessa abordagem arquitetural, que visa dividir as aplicações front-end em componentes menores e independentes. O destaque para os micro-frontends pode indicar uma

tendência em direção a arquiteturas mais modulares e escaláveis para o desenvolvimento front-end.

Além disso, alguns estudos selecionados optaram por não detalhar a arquitetura utilizada em suas pesquisas, o que representa outros 32% dos casos. Isso pode indicar uma adoção de abordagens mais tradicionais, como o modelo monolítico, ou pode refletir a falta de clareza na documentação dos métodos arquiteturais empregados.

Outras abordagens arquiteturais também são evidentes nos estudos primários, com 12% dos trabalhos mencionando explicitamente uma abordagem de três camadas, como [A7, A14, A15]. Além disso, 12% dos estudos destacaram o uso de Single Page Applications (SPA), uma abordagem que concentra a lógica do lado do cliente em uma única página da web interativa. A arquitetura de Progressive Web App (PWA) também foi mencionada em 8% dos estudos, demonstrando o interesse em desenvolver aplicativos que sejam confiáveis e rápidos, independentemente da conectividade.

Table 8: Tecnologias e metodologias atuais para o desenvolvimento de aplicações front-end (QG).

Resposta	Quantidade	Percentual	Lista de Estudos Primários
Javascript	19/25	76%	[A1], [A2], [A3], [A4], [A5], [A6], [A7], [A8], [A9], [A12], [A14], [A15], [A16], [A19], [A20], [A21], [A23], [A24], [A25]
Webpack	12/25	48%	[A2], [A4], [A6], [A7], [A8], [A9], [A16], [A20], [A21], [A23], [A24], [A25]
HTML	11/25	44%	[A1], [A3], [A5], [A7], [A9], [A12], [A14], [A15], [A19], [A20], [A24]
CSS	11/25	44%	[A1], [A3], [A5], [A7], [A9], [A12], [A14], [A15], [A19], [A20], [A24]
Bootstrap	3/25	12%	[A5], [A7], [A21]
Redux	2/25	8%	[A9], [A21]
Jquery	2/25	8%	[A5], [A7]
LitElement	1/25	4%	[A1]
Date picker	1/25	4%	[A9]
React MUI	1/25	4%	[A9]
WebSocket	1/25	4%	[A11]
Docker	1/25	4%	[A19]
SASS	1/25	4%	[A19]
styled-components	1/25	4%	[A20]
Typescript	1/25	4%	[A20]
Babel	1/25	4%	[A20]
Design System	1/25	4%	[A20]
Axios	1/25	4%	[A21]
Visual Studio Code	1/25	4%	[A24]
Postman	1/25	4%	[A24]
Git	1/25	4%	[A24]
React developer tools extension	1/25	4%	[A24]
Redux devtools extensions	1/25	4%	[A24]
Não especificado	5/25	20%	[A10], [A13], [A17], [A18], [A22]

Table 9: Arquiteturas para desenvolver aplicações front-end (FQ1).

Resposta	Quantidade	Percentual	Lista de Estudos Primários
micro-frontend	8/25	32%	[A1], [A2], [A4], [A11], [A16], [A20], [A23], [A24]
MVC	3/25	12%	[A7], [A14], [A15]
SPA	3/25	12%	[A8], [A16], [A20]
PWA	2/25	8%	[A4], [A25]
Component-Based Architecture	2/25	8%	[A20], [A21]
REST API	1/25	4%	[A5]
React DOM	1/25	4%	[A9]
Module Federation	1/25	4%	[A16]
Multi-tenancy	1/25	4%	[A18]
Modular front-end Architecture	1/25	4%	[A20]
Não especificado	8/25	32%	[A3], [A6], [A10], [A12], [A13], [A17], [A19], [A22]

Conclusão da FQ1: A FQ1 explorou 10 arquiteturas voltadas para o desenvolvimento front-end, destacando o micro-frontend como a principal, com 32%. Notavelmente, 32% dos casos não mencionaram qualquer tipo de arquitetura. Isso sugere uma carência na inclusão de tópicos relacionados à arquitetura de software no processo de formação de profissionais nesta área. Além disso, a FQ1 orienta gestores ao evidenciar a necessidade de contar com profissionais especializados em arquitetura de software para enfrentar protipação de cada aplicação.

6.3 FQ2: Quais os frameworks ou bibliotecas estão sendo utilizados?

A questão de pesquisa FQ2 desempenha um papel fundamental na compreensão do cenário atual das tecnologias de front-end, buscando identificar os frameworks e bibliotecas mais utilizados no

desenvolvimento de aplicações modernas. A Tabela 10 apresenta um panorama abrangente dos resultados coletados a partir dos estudos primários. Surpreendentemente, a maioria dos estudos (48%, ou seja, 12 de 25) não faz menção direta aos frameworks ou bibliotecas específicas utilizados.

No entanto, entre os estudos que mencionaram frameworks específicos, emergem três líderes no cenário do front-end: ReactJS, Angular e VueJS. ReactJS, com 36% de adoção, é o framework mais citado, evidenciando sua popularidade entre os desenvolvedores e pesquisadores. Angular (20%) e VueJS (8%) também desempenham papéis significativos, mostrando a diversidade de opções disponíveis.

Além disso, no contexto de micro-frontends, destaca-se a presença da biblioteca "single-spa", projetada para conectar múltiplos frameworks em uma única aplicação, atuando como um "gateway" para aplicações front-end. Embora os estudos relacionados tenham citado essa abordagem em apenas 8% dos casos, essa área parece promissora e poderia beneficiar-se de pesquisas adicionais para avaliar plenamente seus benefícios e desafios.

Conclusão da FQ2: A FQ2 explorou 8 frameworks destinados ao desenvolvimento front-end, evidenciando o ReactJS como o mais proeminente, presente em 36% dos estudos. É notável que em 48% dos casos, nenhum framework específico foi mencionado. Isso reflete a constante evolução das tecnologias web, gerando uma incerteza ao adotar um framework específico que pode impactar a vida útil do projeto. Esses resultados fornecem orientação valiosa para gestores e desenvolvedores na escolha de um framework seguro e validado, considerando a dinâmica do cenário tecnológico.

6.4 FQ3: Quais são patterns para implementar os micro-frontend?

A questão de pesquisa FQ3 direciona nossa atenção para a identificação de padrões comuns na implementação de micro-frontends. A Tabela 11, que apresenta uma visão geral dos resultados, destaca que a maioria dos trabalhos analisados (56%, ou seja, 14 de 25) não suportam explicitamente nenhum padrão específico de micro-frontend. Isso pode ser um reflexo da relativa novidade desse campo acadêmico, uma vez que a elaboração acadêmica de micro-frontends começou a ganhar destaque em torno de 2016.

No entanto, entre os padrões que foram identificados, "Web Component" se destaca como o mais prevalente, com 28% de adoção, encontrado em 7 dos 25 trabalhos. Além disso, "Component-Based," "Composition Architecture," e "Modular micro-frontend Architecture" são padrões identificados com 8% de adoção cada, presentes em 2 dos 25 trabalhos.

Essa análise revela uma oportunidade significativa para pesquisas futuras. A falta de um consenso claro sobre os padrões de micro-frontend sugere a necessidade de um estudo mais aprofundado e do preenchimento da lacuna entre a literatura cinza (ou prática) e a academia. Explorar o motivo pelo qual a maioria dos trabalhos não adota explicitamente padrões e investigar como abordar esse

desafio pode ser um caminho promissor para futuras investigações nessa área.

Conclusão FQ3: A FQ3 derivou 9 padrões para a implementação de micro-frontends, destacando o Web Component como o mais utilizado em 28% dos estudos primários. É notável também que em 56% dos estudos, nenhum padrão específico foi mencionado. Este resultado sugere uma lacuna na inclusão de tópicos relacionados a padrões de projetos na formação de profissionais para a criação de interfaces gráficas. Além disso, destaca a importância para gestores de investir no aprimoramento das equipes, reconhecendo a relevância desses padrões na prática de desenvolvimento.

6.5 FQ4: Como os micro-frontend são avaliados e mensurados?

A questão de pesquisa FQ4 desempenha um papel crucial na busca por uma compreensão mais profunda de como os micro-frontends são avaliados e mensurados. A Tabela 12 nos fornece um panorama dos resultados obtidos a partir dos estudos primários. No entanto, um desafio notável se destaca: a falta de consenso na abordagem de avaliação e mensuração dos micro-frontends.

É interessante observar que um grande percentual de trabalhos (76%, ou seja, 19 de 25) não aborda de maneira explícita como os micro-frontends são avaliados, o que cria uma lacuna significativa na pesquisa e na compreensão desta área. A ausência de diretrizes claras para a avaliação de micro-frontends pode tornar difícil a comparação e a definição de boas práticas. Portanto, existe uma oportunidade empolgante para pesquisas futuras focarem nesse ponto, explorando como os micro-frontends podem ser avaliados de maneira eficaz, identificando métricas relevantes e desenvolvendo diretrizes práticas para a mensuração.

Conclusão FQ4: A FQ4 identificou 9 métodos para avaliar e mensurar micro-frontends. Os resultados revelam que essa área não é amplamente explorada, uma vez que 76% dos estudos não abordam nenhuma forma específica de avaliação. Isso indica uma oportunidade para pesquisadores desenvolverem métricas destinadas a avaliar arquiteturas de front-end. Essas métricas podem fornecer uma abordagem padronizada para gestores e desenvolvedores, ajudando a mitigar o impacto de implementações incorretas de micro-frontends.

6.6 FQ5: Como os benefícios de micro-frontend são percebidos?

A questão de pesquisa FQ5 aborda um ponto essencial: qual é a percepção dos benefícios da adoção de micro-frontends? A Tabela 13 revela uma visão geral dos resultados obtidos em relação a essa questão. Surpreendentemente, um número substancial de estudos (72%, ou seja, 18 de 25) não aborda explicitamente os benefícios percebidos pela adoção de micro-frontends. Isso pode ser atribuído a várias razões, incluindo a falta de foco em avaliar os benefícios ou,

Table 10: Frameworks/bibliotecas utilizados (FQ2).

Resposta	Quantidade	Percentual	Lista de Estudos Primários
ReactJS	9/25	36%	[A2], [A4], [A6], [A7], [A9], [A20], [A21], [A24], [A25]
Angular	5/25	20%	[A4], [A7], [A8], [A16]
Vue	2/25	8%	[A4], [A7]
single-spa	2/25	8%	[A20], [A23]
Redux	2/25	8%	[A21], [A24]
Apache Cordova	1/25	4%	[A19]
Ionic	1/25	4%	[A19]
NextJS	1/25	4%	[A24]
Não especificado	12/25	48%	[A1], [A3], [A5], [A10], [A11], [A12], [A13], [A14], [A15], [A17], [A18], [A22]

Table 11: Patterns para implementar os micro-frontends (FQ3).

Resposta	Quantidade	Percentual	Lista de Estudos Primários
Web Component	7/25	28%	[A1], [A3], [A5], [A7], [A9], [A12], [A16]
Component-Based	2/25	8%	[A4], [A20]
Composition Architecture	2/25	8%	[A11], [A23]
Modular micro-frontend Architecture	2/25	8%	[A20], [A23]
Fragment Architecture	1/25	4%	[A11]
Without-visual-representation Architecture	1/25	4%	[A11]
Unified SPA	1/25	4%	[A16]
Separate Runtime	1/25	4%	[A23]
Iframe Composition	1/25	4%	[A23]
Não Suporta	14/25	56%	[A2], [A6], [A8], [A10], [A13], [A14], [A15], [A17], [A18], [A19], [A21], [A22], [A24], [A25]

Table 12: Avaliação e mensuração dos micro-frontend(FQ4).

Resposta	Quantidade	Percentual	Lista de Estudos Primários
ATAM-based	1/25	4%	[A1]
Data Visualization	1/25	4%	[A2]
Quality of Experience	1/25	4%	[A16]
Time-Instant	1/25	4%	[A16]
Testing in Test Environment	1/25	4%	[A20]
Cycle of Testing and Issue Fixing	1/25	4%	[A20]
Static Code Analysis	1/25	4%	[A20]
ISO-20510	1/25	4%	[A23]
Load Test	1/25	4%	[A24]
Não Suporta	19/25	76%	[A3], [A4], [A5], [A6], [A7], [A8], [A9], [A10], [A11], [A12], [A13], [A14], [A15], [A17], [A18], [A19], [A21], [A22], [A25]

em alguns casos, os autores podem não ter explorado ou destacado esses aspectos em seus trabalhos.

No entanto, entre os estudos que abordam os benefícios, três aspectos se destacam: Manutenção, Escalabilidade e Flexibilidade. Cada um desses aspectos é percebido como benéfico por aproximadamente 20% dos estudos. Isso sugere que a implementação de

micro-frontends pode oferecer vantagens significativas na manutenção mais eficaz de sistemas, na escalabilidade para atender às demandas crescentes e na flexibilidade para incorporar diversos frameworks dentro de um projeto.

Table 13: Percepção dos benefícios do micro-frontend (FQ5).

Resposta	Quantidade	Percentual	Lista de Estudos Primários
Manutenção	5/25	20%	[A1], [A11], [A16], [A20], [A24]
Escalabilidade	5/25	20%	[A1], [A11], [A16], [A20], [A23]
Flexibilidade	4/25	16%	[A1], [A11], [A20], [A23]
ConFigurabilidade	2/25	8%	[A1], [A11]
Redução de Custo	1/25	4%	[A1]
Implementação	1/25	4%	[A11]
Versionamento Independente	1/25	4%	[A20]
Não Suporta	18/25	72%	[A2], [A3], [A4], [A5], [A6], [A7], [A8], [A9], [A10], [A12], [A14], [A15], [A17], [A18], [A19], [A21], [A22], [A25]

Conclusão FQ5: A FQ5 identificou 7 percepções das implementações de micro-frontends. A maioria dos resultados apresenta uma correlação com seu conceito próximo, os micro serviços, sugerindo uma interseção significativa entre ambos. No entanto, permanece uma oportunidade para os pesquisadores conduzirem uma pesquisa abrangente, como um levantamento (survey), junto às empresas que implementaram micro-frontends, a fim de validar e aprofundar os resultados obtidos até o momento. Isso pode proporcionar uma compreensão mais holística e prática das percepções das organizações em relação a essas implementações.

Conclusão FQ6: A FQ6 gerou 16 resultados relacionados à motivação da adoção de micro-frontends, com Modularidade e Isolamento se destacando como o resultado mais expressivo, presente em 24% dos estudos primários. Esse achado auxilia desenvolvedores e gestores ao fornecer uma base sólida para sustentar a escolha dessa arquitetura. Para pesquisadores, abre uma oportunidade de realizar levantamentos adicionais para validar se essas percepções permanecem consistentes após a implementação efetiva dessa arquitetura. Isso contribuiria para uma compreensão mais aprofundada e prática dos motivos subjacentes à adoção de micro-frontends.

6.7 FQ6: Quais são as motivações para adoção de micro-frontend?

A questão de pesquisa FQ6 busca compreender as motivações por trás da adoção de micro-frontends, fornecendo insights valiosos sobre os problemas inerentes às arquiteturas monolíticas no front-end. A Tabela 14 apresenta os resultados relacionados a esta questão, e revela uma diversidade de perspectivas em relação às motivações para adotar essa abordagem inovadora.

É interessante notar que a maioria dos trabalhos (72%, ou seja, 18 de 25) não menciona explicitamente nenhuma motivação específica para a adoção de micro-frontends. Isso pode ser interpretado de várias maneiras, indicando que alguns projetos podem não ter necessidades ou complexidades que justifiquem a transição para essa arquitetura. No entanto, essa falta de clareza também aponta para uma oportunidade de pesquisa futura, que pode incluir a realização de pesquisas ou estudos de caso mais detalhados para entender melhor as razões subjacentes à não adoção de micro-frontends em alguns cenários.

Por outro lado, entre os trabalhos que mencionam razões para a adoção de micro-frontends, destacam-se a Modularidade e Isolamento, que representam 24% dos casos, e a Flexibilidade, citada em 16% dos estudos. Essas motivações ressaltam a importância da modularidade e da capacidade de isolar componentes e funcionalidades em projetos de front-end complexos. A adoção de micro-frontends permite a flexibilidade de incorporar frameworks e bibliotecas específicas para abordar desafios específicos dentro do projeto.

6.8 QE1: Onde os estudos estão sendo conduzidos?

A pesquisa QE1 desempenha um papel fundamental ao analisar os países onde os estudos sobre micro-frontends estão sendo conduzidos. A Tabela 15 fornece uma visão geral da distribuição geográfica desses estudos. Com base na afiliação institucional dos autores participantes dos artigos, é possível observar padrões interessantes.

Os resultados apontam para uma concentração significativa de pesquisas na Europa, representando 68% dos estudos selecionados. Isso sugere que a Europa é uma região líder em explorar e investigar a arquitetura de micro-frontends, com países como Finlândia, Portugal, Alemanha e Espanha contribuindo ativamente para o corpo de conhecimento nesse campo.

Esses resultados destacam não apenas a importância de explorar a pesquisa em micro-frontends em diferentes partes do mundo, mas também a oportunidade de promover a colaboração internacional e a disseminação de conhecimento para abordar desafios globais no desenvolvimento de front-end. Além disso, essa análise geográfica pode inspirar pesquisadores e desenvolvedores em regiões menos representadas a se envolverem mais ativamente neste campo de pesquisa em crescimento.

Table 14: Motivações para adoção de micro-frontend (FQ6).

Resposta	Quantidade	Percentual	Lista de Estudos Primários
Modularidade e isolamento	6/25	24%	[A16], [A1], [A2], [A11], [A20], [A23]
Flexibilidade	4/25	16%	[A1], [A11], [A20], [A23]
Desenvolvimento independente	3/25	12%	[A1], [A20], [A23]
Escalabilidade	3/25	12%	[A16], [A20], [A23]
Manutenção	3/25	12%	[A16], [A1], [A20]
Problemas induzidos pela estrutura	2/25	8%	[A1], [A4]
Custos reduzidos	1/25	4%	[A1]
Ciclos de lançamento Curto	1/25	4%	[A1]
Diminuição de gargalo	1/25	4%	[A2]
Performance	1/25	4%	[A16]
Risk mitigation	1/25	4%	[A16]
Release independente	1/25	4%	[A20]
Reutilização de código	1/25	4%	[A20]
Crescimento de usuários em web Applications devido à COVID-19	1/25	4%	[A24]
Não Suporta	18/25	72%	[A3], [A5], [A6], [A7], [A8], [A9], [A10], [A12], [A13], [A14], [A15], [A17], [A18], [A19], [A21], [A22], [A25], [A2]

Table 15: Local dos estudos (QE1).

Resposta	Quantidade	Percentual	ID Trabalho
Finlândia	4	16%	[A1], [A20], [A21], [A25]
Portugal	4	16%	[A6], [A8], [A15], [A23]
Indonésia	3	12%	[A7], [A22], [A24]
Alemanha	3	12%	[A11], [A13], [A17]
Índia	2	8%	[A2], [A9]
Espanha	2	8%	[A4], [A19]
Bahrein	1	4%	[A3]
Eslováquia	1	4%	[A5]
Suíça	1	4%	[A10]
China	1	4%	[A12]
Taiwan	1	4%	[A14]
Áustria	1	4%	[A16]
Arábia Saudita	1	4%	[A18]

Conclusão da QE1: A QE1 identificou, por meio de uma pesquisa quantitativa, 13 países com foco na exploração de arquiteturas front-end. Contudo, observou-se que os continentes Americano, Africano e a Oceania estão sub-representados. Essa constatação sinaliza uma oportunidade para pesquisadores desses países conduzirem estudos mais aprofundados sobre o tema. Essa iniciativa forneceria uma direção valiosa para a academia, desenvolvedores e gestores dessas regiões na implementação de arquiteturas front-end, contribuindo assim para um entendimento mais abrangente e diversificado nesse campo.

6.9 QE2: Qual o número de publicações por ano?

A pesquisa QE2 oferece uma análise detalhada das publicações anuais sobre front-ends, revelando padrões na evolução desse campo.

Em torno de 2008, o tema representava 4%, indicando um interesse inicial. Até 2012, manteve uma presença constante, permanecendo nos 4%. Em 2014, a pesquisa sobre front-ends começou a ganhar notoriedade, representando 12% dos estudos. Após uma leve queda em 2017 (4%), a pesquisa se recuperou, apresentando crescimento em 2018 (8%) e estabilidade em 2019 (12%). Essa tendência foi mantida em 2020, também alcançando 12%. O ano de 2021 se destacou, representando 24% das publicações de estudos primários. Isso sugere um aumento significativo no interesse e nas pesquisas relacionadas a front-ends nesse período. Até 2023, 16% dos estudos primários já foram publicados, indicando um interesse contínuo.

Essa ascensão nos últimos anos reflete a crescente complexidade das aplicações front-end. Empresas e pesquisadores buscam estudos que auxiliem na definição de arquiteturas visuais mais eficientes. A Figura 2 destaca visualmente esses padrões, ressaltando a notável evolução do campo.

Tipo de publicação	Quantidade	Percentual
Periódico	15	60%
Conferência	10	40%

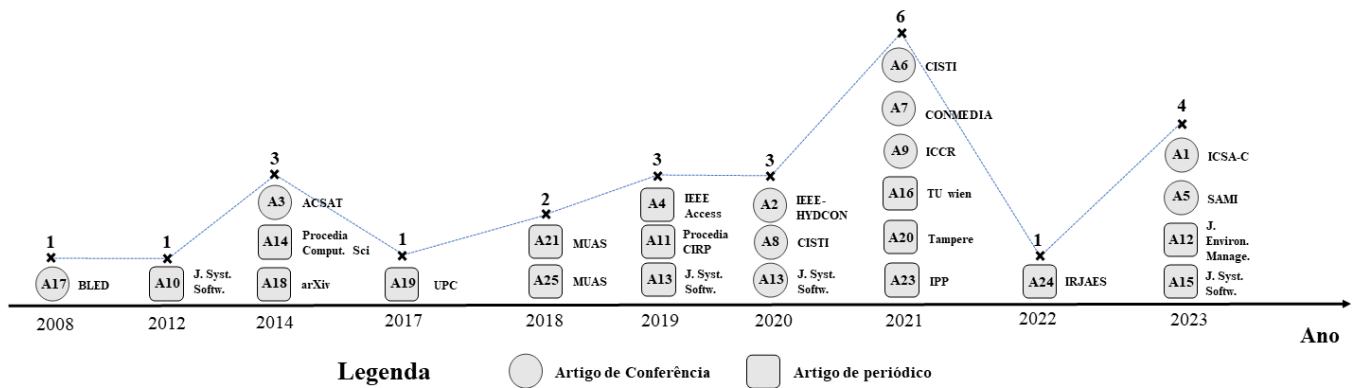


Figure 2: Distribuição dos estudos primários ao longo dos anos (baseado em [18])

Conclusão QE2: A QE2 definiu de forma clara a evolução das pesquisas no decorrer de 15 anos. Essa análise temporal destaca a importância crescente das arquiteturas no campo de desenvolvimento de front-end e aponta para a necessidade contínua de pesquisa e inovação para lidar com os desafios crescentes nesse domínio.

7 DISCUSSÃO E DIREÇÕES FUTURAS

Esta seção tem como objetivo aprofundar a análise dos dados obtidos no presente estudo e fornecer insights sobre as possíveis direções futuras para pesquisadores e profissionais da área. Com base nos dados coletados, a arquitetura de micro-frontend emerge como um tópico subexplorado, embora se torne evidente que os projetos que optaram por sua implementação colheram benefícios significativos em termos de manutenção, escalabilidade e isolamento [39].

Falta de Arquitetura de Software: Busca-se compreender como as arquiteturas de software para o front-end têm sido abordadas ao longo dos anos. Para alcançar esse objetivo, organiza-se os estudos primários utilizando um gráfico de bolhas, conforme apresentado na Figura 3, e considera-se três dimensões de dados. Cada bolha representa uma tripla (d1, d2, d3), onde d1 indica o tipo de arquitetura abordada, d2 é o ano de publicação, e d3 representa a quantidade de estudos explorando um determinado tópico de

pesquisa. Este gráfico possibilita compreender, com base em evidências quantitativas, em quais tópicos de pesquisa a comunidade tem dedicado mais atenção.

Notavelmente, a arquitetura de micro-frontend tem sido a mais abordada nos últimos 5 anos, representando 32% (8/25) dos estudos. Isso evidencia uma crescente preocupação em relação ao tipo de arquitetura front-end, indicando que os softwares têm demandado uma atenção cada vez maior para esse tema. A ausência de uma arquitetura específica pode resultar em uma vida útil reduzida do sistema. No entanto, é alarmante observar que um número significativo de estudos selecionados, embora tratem da elaboração de aplicações específicas, não abordam nenhum tipo específico de arquitetura. Isso contribui para a degradação arquitetural nos sistemas desenvolvidos [8, 41].

Ampla opção de tecnologias. A análise dos dados revelou uma notável diversidade de tecnologias e frameworks usados no desenvolvimento front-end. Isso reflete a natureza em constante evolução desse campo, onde as preferências e as ferramentas mudam rapidamente. É importante ressaltar que a escolha das tecnologias deve ser guiada pelas necessidades específicas de cada projeto [1]. A diversidade de opções também sugere a necessidade de um maior foco na adaptação às mudanças constantes e na seleção criteriosa de tecnologias para otimizar a eficiência e a eficácia do desenvolvimento.

A variedade de tecnologias e frameworks disponíveis para o desenvolvimento front-end é uma característica marcante deste

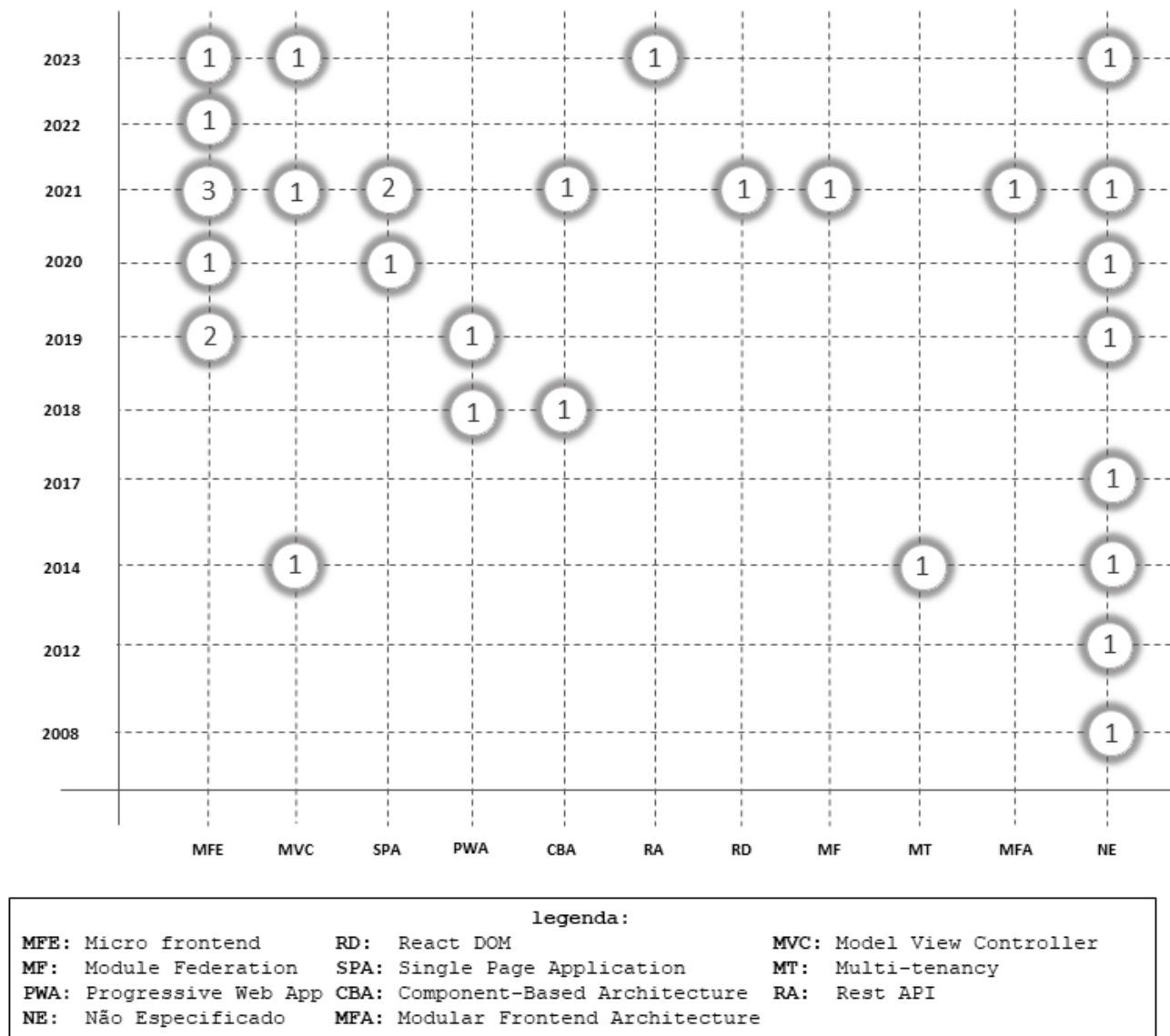


Figure 3: Gráfico de bolhas que mostra a relação entre três variáveis. O eixo X exibe a arquitetura utilizada. O eixo Y representa o ano. O tamanho dos discos é determinado pelo número de estudos primários classificados de acordo com os critérios ao longo dos eixos X e Y. (baseado em [24])

campo. No entanto, essa diversidade também pode apresentar desafios, especialmente quando se trata de escolher as tecnologias certas para um projeto específico. Desenvolvedores e equipes de projeto devem considerar cuidadosamente as necessidades e requisitos de seus aplicativos ao selecionar as ferramentas apropriadas. Além disso, é essencial acompanhar de perto a evolução do campo e estar disposto a se adaptar a novas tecnologias à medida que surgirem.

Lacunas educacionais. As arquiteturas de micro-frontend, como seus equivalentes em micro serviços, demonstraram ser desafiadoras de implementar. Elas exigem um alto grau de conhecimento por parte das equipes de desenvolvimento e incorrem em custos significativos para manutenção [32, 43]. Ainda assim, os benefícios em termos de escalabilidade e isolamento tornam essas arquiteturas atrativas para projetos de grande porte. Portanto, um foco contínuo na gestão de custos e no aprimoramento das habilidades da equipe é essencial.

A implementação de arquiteturas de micro-frontend não é uma tarefa simples. Requer um profundo conhecimento técnico e a capacidade de lidar com desafios complexos, como a orquestração dinâmica de páginas da web, testes e depuração. Portanto, é fundamental que as equipes de desenvolvimento estejam preparadas para enfrentar esses desafios e investir na automação de processos para melhorar a eficiência.

8 CONCLUSÃO E TRABALHOS FUTUROS

O estudo adotou uma abordagem de mapeamento sistemático voltada para micro-frontends. Foi desenvolvido um protocolo de pesquisa metódico, detalhando de maneira clara os objetivos, as questões de pesquisa, a estratégia de busca, os critérios de exclusão e inclusão, além do procedimento de extração de dados. No total, foram selecionados cuidadosamente 25 estudos primários após a análise de uma lista inicial de 5308, proveniente de cinco renomados bancos de dados eletrônicos. Esse processo permitiu ampliar as fronteiras do conhecimento no desenvolvimento de front-ends modernos, proporcionando diretrizes mais claras para profissionais e acadêmicos interessados nesse campo dinâmico.

Os resultados indicam que: (i) diversas tecnologias são amplamente empregadas no desenvolvimento front-end, em vez de uma específica; (ii) as arquiteturas de software não são detalhadas de maneira abrangente; (iii) a maioria dos trabalhos não aborda ou detalha um framework específico; (iv) os estudos não oferecem informações detalhadas sobre padrões de desenvolvimento para arquiteturas de micro-frontends; (v) a avaliação dos micro-frontends não é uma preocupação central, visto que 76% dos estudos não fazem menção a essa etapa; (vi) a maioria dos trabalhos não explora os benefícios da adoção de micro-frontends ou de arquiteturas front-end em geral; (vii) há falta de clareza sobre os motivos para a adoção de micro-frontends, já que 72% dos estudos não detalham esses motivos; (viii) os continentes Americano, Oceania e África, raramente abordam esse tema específico, resultando na exclusão de um conjunto de dados mais diversificado.

Algumas recomendações para pesquisas futuras seriam: aumentar amplitude da análise dos estudos primários; refinar a pesquisa sobre arquiteturas para front-end; conduzir uma revisão sistemática da literatura para examinar rigorosamente os padrões, tecnologias ou questões de qualidades no desenvolvimento de front-end; ampliar pesquisas nos continentes Americano, Oceania e África, com foco em identificar o que profissionais e pesquisadores tem utilizado.

Além disso, pesquisadores e profissionais da indústria poderiam explorar dois temas interessantes. Primeiro, a avaliação de carga cognitiva [16, 53] em tarefas de desenvolvimento de software tem crescido nos últimos anos. Neste sentido, com a crescente adoção da arquitetura micro-frontend, é imperativo explorar a carga cognitiva imposta aos desenvolvedores durante a fase de implementação, por exemplo, de micro-frontends usando tecnologias emergentes. Em outras palavras, este estudo permitiria investigar e avaliar a carga cognitiva vivenciada por desenvolvedores de software ao trabalhar com arquiteturas micro-frontend, focando principalmente nos aspectos de modularidade e reuso. Segundo, relacionar experiência do usuário e carga cognitiva em aplicações micro-frontend ainda é um tema inexplorado. Investigar como os usuários finais

percebem e vivenciam a carga cognitiva ao interagir com aplicações baseadas em micro-frontend, considerando fatores como navegação, capacidade de resposta e experiência geral do usuário, emerge como um tema estratégico. Finalmente, esperamos que os resultados discutidos ao longo deste estudo possam incentivar pesquisadores e profissionais a fechar as lacunas citadas. Sendo esse trabalho, um primeiro passo em como avançar a literatura atual sobre arquiteturas front-end.

REFERENCES

- [1] 2011. IEEE Guide—Adoption of ISO/IEC TR 24748-1:2010 Systems and Software Engineering—Life Cycle Management—Part 1: Guide for Life Cycle Management. *IEEE Std 24748-1-2011* (2011), 1–96. <https://doi.org/10.1109/IEEESTD.2011.5871657>
- [2] Hanin M. Abdullah and Ahmed M. Zeki. 2014. Frontend and Backend Web Technologies in Social Networking Sites: Facebook as an Example. In *2014 3rd International Conference on Advanced Computer Science Applications and Technologies*. 85–89. <https://doi.org/10.1109/ACSAT.2014.22>
- [3] Christoph Adolphs and Petra Schubert. 2008. PersoBOX: A Personalization Engine between ERP System and Web Frontend. *BLED 2008 Proceedings* (2008), 7.
- [4] Aakash Ahmad and Muhammad Ali Babar. 2016. Software architectures for robotic systems: A systematic mapping study. *Journal of Systems and Software* 122 (2016), 16–39. <https://doi.org/10.1016/j.jss.2016.08.039>
- [5] Shatha Al-Qudah, Kenza Meridji, and Khalid T. Al-Sarayreh. 2015. A Comprehensive Survey of Software Development Cost Estimation Studies. In *Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication* (Batna, Algeria) (IPAC '15). Association for Computing Machinery, New York, NY, USA, Article 53, 5 pages. <https://doi.org/10.1145/2816839.2816913>
- [6] Cassiano Andrade, Paulo Alves, José Eduardo Fernandes, and Flávio Coutinho. 2020. A web platform to support mentoring programs in higher education. In *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*. 1–6. <https://doi.org/10.23919/CISTI49556.2020.9140982>
- [7] Hiuram Antunes and Inácio de Sousa Adelino da Fonseca. 2021. Advanced web methodology for flexible web development. In *2021 16th Iberian Conference on Information Systems and Technologies (CISTI)*. 1–4. <https://doi.org/10.23919/CISTI52073.2021.9476295>
- [8] Ahmed Baabad, Hazura Binti Zulzalil, Salmi Binti Baharom, et al. 2020. Software architecture degradation in open source software: A systematic literature review. *IEEE Access* 8 (2020), 173681–173709.
- [9] Ahmed Baabad, Hazura Binti Zulzalil, Sa'adah Hassan, and Salmi Binti Baharom. 2020. Software Architecture Degradation in Open Source Software: A Systematic Literature Review. *IEEE Access* 8 (2020), 173681–173709. <https://doi.org/10.1109/ACCESS.2020.3024671>
- [10] Vinicius Bischoff, Kleinner Farias, Lucian José Gonçalves, and Jorge Luis Vitória Barbosa. 2019. Integration of feature models: A systematic mapping study. *Information and Software Technology* 105 (2019), 209–225. <https://doi.org/10.1016/j.infsof.2018.08.016>
- [11] Vinicius Bischoff, Kleinner Farias, Juliano Paulo Menzen, and Gustavo Pessin. 2021. Technological support for detection and prediction of plant diseases: A systematic mapping study. *Computers and Electronics in Agriculture* 181 (2021), 105922. <https://doi.org/10.1016/j.compag.2020.105922>
- [12] I Wayan Krishna Dharma BP and Dina Anggraini. 2022. a Development of Modern Web Application Frontend Structures Using Micro Frontends. (2022).
- [13] Carlos Carbonera, Kleinner Farias, CéSar Graeff, and Robson Silva. 2023. Software Merge: A Two-Decade Systematic Mapping Study. In *Proceedings of the XXXVII Brazilian Symposium on Software Engineering* (Campo Grande, Brazil) (SBES '23). Association for Computing Machinery, New York, NY, USA, 99–108. <https://doi.org/10.1145/3613372.3613391>
- [14] Idette D. Cooper. 2016. What is a "mapping study?". *Journal of Medical Library Association* 104, 1 (January 2016), 76–78. <https://doi.org/10.3163/1536-5050.104.1.013>
- [15] Ricardo Alexandre Pinto da Silva. 2021. *A Micro Frontends Solution—Analyzing quality attributes*. Ph.D. Dissertation. Instituto Politécnico do Porto (Portugal).
- [16] Guilherme Dalcin, Willian Bolzan, Luan Lazzari, and Kleinner Farias. 2023. Recommendation of UML Model Conflicts: Unveiling the Biometric Lens for Conflict Resolution. In *Proceedings of the XXXVII Brazilian Symposium on Software Engineering*. 83–88.
- [17] Palak Dwivedi, Kshamta, and Abhishek Joshi. 2022. ReactJS For Trading Applications. In *2022 International Conference on Cyber Resilience (ICCR)*. 01–07. <https://doi.org/10.1109/ICCR56254.2022.9995932>
- [18] Carlos Eduardo Carbonera, Kleinner Farias, and Vinicius Bischoff. 2020. Software development effort estimation: a systematic mapping study.

- IET Software* 14, 4 (2020), 328–344. <https://doi.org/10.1049/iet-sen.2018.5334> arXiv:<https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-sen.2018.5334>
- [19] Kleinner Farias, Alessandro Garcia, and Carlos Lucena. 2014. Effects of stability on model composition effort: an exploratory study. *Software & Systems Modeling* 13, 4 (01 Oct 2014), 1473–1494. <https://doi.org/10.1007/s10270-012-0308-2>
 - [20] Jaume Font Escrivano and Vilma Karina Reynoso Vasquez. 2017. *ERP implementation for an administrative agency as a corporate frontend and an e-commerce smartphone app*. Master's thesis. Universitat Politècnica de Catalunya.
 - [21] Michael Geers. 2020. *Micro frontends in action*. Simon and Schuster.
 - [22] Lucian Gonçalves, Kleinner Farias, Bruno da Silva, and Jonathan Fessler. 2019. Measuring the Cognitive Load of Software Developers: A Systematic Mapping Study. In *2019 IEEE/ACM 27th International Conference on Program Comprehension (ICPC)*. 42–52. <https://doi.org/10.1109/ICPC.2019.00018>
 - [23] Lucian José Gonçalves, Kleinner Farias, and Bruno C. da Silva. 2021. Measuring the cognitive load of software developers: An extended Systematic Mapping Study. *Information and Software Technology* 136 (2021), 106563. <https://doi.org/10.1016/j.infsof.2021.106563>
 - [24] Lucian José Gonçalves, Kleinner Farias, Toacy Cavalcante De Oliveira, and Murilo Scholl. 2019. Comparison of Software Design Models: An Extended Systematic Mapping Study. *ACM Comput. Surv.* 52, 3, Article 48 (jul 2019), 41 pages. <https://doi.org/10.1145/3313801>
 - [25] J Gunawan and RR Kosala. 2020. Genie enterprise resource planning for small medium enterprises implementing single page web application. In *IOP Conference Series: Earth and Environmental Science*, Vol. 426. IOP Publishing, 012170.
 - [26] Holger Harms, Collin Rogowski, and Luigi Lo Iacono. 2017. Guidelines for Adopting Frontend Architectures and Patterns in Microservices-Based Systems. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering (Paderborn, Germany) (ESEC/FSE 2017)*. Association for Computing Machinery, New York, NY, USA, 902–907. <https://doi.org/10.1145/3106237.3117775>
 - [27] Mira Hiltunen. 2018. Creating multiplatform experiences with progressive web apps. (2018).
 - [28] Magne Jorgensen and Martin Shepperd. 2007. A Systematic Review of Software Development Cost Estimation Studies. *IEEE Transactions on Software Engineering* 33, 1 (2007), 33–53. <https://doi.org/10.1109/TSE.2007.256943>
 - [29] S. Keshav. 2007. How to Read a Paper. *SIGCOMM Comput. Commun. Rev.* 37, 3 (jul 2007), 83–84. <https://doi.org/10.1145/1273445.1273458>
 - [30] B Kitchenham and S Charters. 2007. *Guidelines for performing systematic literature reviews in software engineering*. Technical Report. Technical report, EBSE Technical Report EBSE-2007-01. <https://www.cs.auckland.ac.nz/~norsaremah/2007%20Guidelines%20for%20performing%20SLR%20in%20SE%20v2.3.pdf>
 - [31] Jarno Koppala. 2018. *Erp solution with reactjs*. (2018).
 - [32] Manuel Kroiß. 2021. *From Backend to Frontend-Case study on adopting Micro Frontends from a Single Page ERP Application monolith*. Ph.D. Dissertation. Wien.
 - [33] Rosemary Francisco Leonardo dos Santos Paula, Lucas Pfeiffer Salomão Dias and Jorge Luis Victória Barbosa. 2022. Analysing IoT Data for Anxiety and Stress Monitoring: A Systematic Mapping Study and Taxonomy. *International Journal of Human-Computer Interaction* 0, 0 (2022), 1–21. <https://doi.org/10.1080/10447318.2022.2132361> arXiv:<https://doi.org/10.1080/10447318.2022.2132361>
 - [34] Lasse Linkola. 2021. *Design and Implementation of Modular Frontend Architecture on Existing Application*. Master's thesis. Tampere University.
 - [35] Manel Mena, Antonio Corral, Luis Iribarne, and Javier Criado. 2019. A Progressive Web Application Based on Microservices Combining Geospatial Data and the Internet of Things. *IEEE Access* 7 (2019), 104577–104590. <https://doi.org/10.1109/ACCESS.2019.2932196>
 - [36] Juliano Paulo Menzen, Kleinner Farias, and Vinicius Bischoff. 2021. Using biometric data in software engineering: a systematic mapping study. *Behaviour & Information Technology* 40, 9 (2021), 880–902.
 - [37] Kristian Micko, Michal Beca, Peter Papcun, and Iveta Zolotova. 2023. Responsive Web Applications Enables Automatization of Karaoke Song Playlist. In *2023 IEEE 21st World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. 000133–000138. <https://doi.org/10.1109/SAMI58000.2023.10044533>
 - [38] Leticia Montalvilho and Oscar Diaz. 2016. Requirement-driven evolution in software product lines: A systematic mapping study. *Journal of Systems and Software* 122 (2016), 110–143. <https://doi.org/10.1016/j.jss.2016.08.053>
 - [39] Jouni Männistö, Antti-Pekka Tuovinen, and Mikko Raatikainen. 2023. Experiences on a Frameworkless Micro-Frontend Architecture in a Small Organization. In *2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C)*. 61–67. <https://doi.org/10.1109/ICSA-C57050.2023.00025>
 - [40] Anderson Oliveira, Vinicius Bischoff, Lucian José Gonçalves, Kleinner Farias, and Matheus Segalotto. 2018. BRCode: An interpretive model-driven engineering approach for enterprise applications. *Computers in Industry* 96 (2018), 86–97. <https://doi.org/10.1016/j.compind.2018.01.002>
 - [41] Anderson Oliveira, Willian Oizumi, Leonardo Sousa, Wesley KG Assunção, Alessandro Garcia, Carlos Lucena, and Diego Cedrim. 2022. Smell Patterns as Indicators of Design Degradation: Do Developers Agree?. In *Proceedings of the XXXVI Brazilian Symposium on Software Engineering*. 311–320.
 - [42] Andrey Pavlenko, Nursultan Askarbekuly, Swati Megha, and Manuel Mazzara. 2020. Micro-frontends: application of microservices to web front-ends. *Journal of Internet Services and Information Security*. 10, 2 (2020).
 - [43] Severi Peltonen, Luca Mezzalana, and Davide Taibi. 2021. Motivations, benefits, and issues for adopting Micro-Frontends: A Multivocal Literature Review. *Information and Software Technology* 136, 106571 (2021).
 - [44] Kai Petersen, Robert Feldt, Shahid Mujtaba, and et al. 2008. Systematic Mapping Studies in Software Engineering. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE)*. 1–10. <https://doi.org/10.14236/ewic/EASE2008.8>
 - [45] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology* 64 (2015), 1–18. <https://doi.org/10.1016/j.infsof.2015.03.007>
 - [46] Y.R. Prajwal, Jainil Viren Parekh, and Rajashree Shettar. 2021. A Brief Review of Micro-frontends. *United International Journal for Research and Technology*. 2, 8 (2021).
 - [47] Dong Qiu, Bixin Li, Shunhui Ji, and Hareton Leung. 2014. Regression Testing of Web Service: A Systematic Mapping Study. *ACM Comput. Surv.* 47, 2, Article 21 (aug 2014), 46 pages. <https://doi.org/10.1145/2631685>
 - [48] Christopher Rafael, Philipus Wijaya Adikusumo, Ngurah Agus Bangkit Sanjaya, and Maria Susan Anggreainy. 2021. Application of Software Engineering in Intermediate and Higher Education through Web Apps Development. In *2021 6th International Conference on New Media Studies (CONMEDIA)*. 135–139. <https://doi.org/10.1109/CONMEDIA53104.2021.9617201>
 - [49] Américo Rio and Fernando Brito e Abreu. 2023. PHP code smells in web apps: Evolution, survival and anomalies. *Journal of Systems and Software* 200 (2023), 111644. <https://doi.org/10.1016/j.jss.2023.111644>
 - [50] Pilar Rodríguez, Alireza Haghighatkah, Lucy Ellen Lwakatare, Susanna Teppola, Tanja Suomalainen, Juho Eskeli, Teemu Karvonen, Pasi Kuvaja, June M. Verner, and Markku Oivo. 2017. Continuous deployment of software intensive products and services: A systematic mapping study. *Journal of Systems and Software* 123 (2017), 263–291. <https://doi.org/10.1016/j.jss.2015.12.015>
 - [51] Maluane Rubert and Kleinner Farias. 2022. On the effects of continuous delivery on code quality: A case study in industry. *Computer Standards & Interfaces* 81 (2022), 103588.
 - [52] Eike Schaffer, Andreas Mayr, Jonathan Fuchs, Martin Sjarov, Johannes Vorndran, and Jörg Franke. 2019. Microservice-based architecture for engineering tools enabling a collaborative multi-user configuration of robot-based automation solutions. *Procedia CIRP* 86 (2019), 86–91. <https://doi.org/10.1016/j.procir.2020.01.017> 7th CIRP Global Web Conference – Towards shifted production value stream patterns through inference of data, models, and technology (CIRPe 2019).
 - [53] Matheus Segalotto, William Bolzan, and Kleinner Farias. 2023. Effects of Modularization on Developers' Cognitive Effort in Code Comprehension Tasks: A Controlled Experiment. In *XXXVII Brazilian Symposium on Software Engineering*. 206–215.
 - [54] Shen Shen, Cheng-Zhi Qin, Liang-Jun Zhu, and A-Xing Zhu. 2023. From scenario to roadmap: Design and evaluation of a web-based participatory watershed planning system for optimizing multistage implementation plans of management practices under stepwise investment. *Journal of Environmental Management* 342 (2023), 118280. <https://doi.org/10.1016/j.jenvman.2023.118280>
 - [55] Harry Sneed and Chris Verhoef. 2019. Re-implementing a legacy system. *Journal of Systems and Software* 155 (2019), 162–184. <https://doi.org/10.1016/j.jss.2019.05.012>
 - [56] Thiago Vieira de Souza, Kleinner Farias, and Vinicius Bischoff. 2020. Big Data Analytics Applied in Supply Chain Management: A Systematic Mapping Study. In *Proceedings of the XVI Brazilian Symposium on Information Systems (São Bernardo do Campo, Brazil) (SBSI '20)*. Association for Computing Machinery, New York, NY, USA, Article 14, 8 pages. <https://doi.org/10.1145/3411564.3411612>
 - [57] Xing-Yu Su, Fong-Ci Lin, Lichin Chen, Kuo-Chin Huang, Chia-Wen Lu, Chia-Yi Chen, Te-Wei Ho, and Feipei Lai. 2014. A Service Oriented Tele-health Promotion Information System with Mobile Application. *Procedia Computer Science* 37 (2014), 274–281. <https://doi.org/10.1016/j.procs.2014.08.041> The 5th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2014)/ The 4th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH 2014)/ Affiliated Workshops.
 - [58] Davide Taibi and Luca Mezzalana. 2022. Micro-Frontends: Principles, Implementations, and Pitfalls. *SIGSOFT Softw. Eng. Notes* 47, 4 (sep 2022), 25–29. <https://doi.org/10.1145/3561846.3561853>
 - [59] P Yedhu Tilak, Vaibhav Yadav, Shah Dhruv Dharmendra, and Narasimha Bolloju. 2020. A platform for enhancing application developer productivity using microservices and micro-frontends. In *2020 IEEE-HYDICON*. 1–4. <https://doi.org/10.1109/HYDICON48903.2020.9242913>
 - [60] Victor Velepucha and Pamela Flores. 2023. A Survey on Microservices Architecture: Principles, Patterns and Migration Challenges. *IEEE Access* 11 (2023), 88339–88358. <https://doi.org/10.1109/ACCESS.2023.3305687>

- [61] Roger Denis Vieira and Kleinner Farias. 2020. Usage of Psychophysiological Data as an Improvement in the Context of Software Engineering: A Systematic Mapping Study. In *Proceedings of the XVI Brazilian Symposium on Information Systems* (São Bernardo do Campo, Brazil) (SBSI '20). Association for Computing Machinery, New York, NY, USA, Article 4, 8 pages. <https://doi.org/10.1145/3411564.3411580>
- [62] Djamal Ziani. 2014. Configuration in ERP SaaS Multi-Tenancy. *arXiv preprint arXiv:1405.0650* (2014).
- [63] Olaf Zimmermann, Christoph Miksovich, and Jochen M. Küster. 2012. Reference architecture, metamodel, and modeling principles for architectural knowledge management in information technology services. *Journal of Systems and Software* 85, 9 (2012), 2014–2033. <https://doi.org/10.1016/j.jss.2012.05.003> Selected papers from the 2011 Joint Working IEEE/IFIP Conference on Software Architecture (WICSA 2011).