

UMA ANÁLISE DO ENTENDIMENTO DE CONCEITOS SOBRE MICRO-FRONTEND NA INDÚSTRIA: UM ESTUDO EXPLORATÓRIO

Felipe Pedroso Xavier¹

Kleinner Farias²

Resumo: Após o surgimento dos microsserviços, o desenvolvimento de software amadureceu muito a nível de desacoplamento, passamos a entender as motivações para utilizar esse padrão de arquitetura. Logo, nasceu o conceito de micro-frontend, para resolver os mesmos problemas que a abordagem de microsserviços, porém no contexto de desenvolvimento frontend. Com o surgimento do micro-frontend nasceram frameworks com a proposta de aplicar essa abordagem e facilitar sua implementação, criando uma camada adicional de estudo para os desenvolvedores. Neste contexto, este trabalho tem como objetivo realizar um estudo empírico sobre o entendimento dos desenvolvedores frontend sobre conceitos essenciais do micro-frontend, explorando benefícios, implantação, roteamento e comunicação desta abordagem, por meio de uma pesquisa utilizando questionários. Os principais resultados obtidos revelam que os desenvolvedores reconhecem os benefícios da implantação independente. No entanto, abordagens como DDD e a composição de micro-frontends via CDN são pouco compreendidas. Além disso, lidar com o roteamento e entender quais tecnologias têm essa responsabilidade demonstrou baixa compreensão. Ao contribuir com um entendimento mais sólido sobre micro-frontend, este estudo oferece oportunidades de melhoria tanto para indústria quanto para o setor acadêmico, destacando conceitos e melhores práticas no desenvolvimento de software a serem exploradas.

Palavras-chave: Desenvolvimento de Software, Microsserviços, Micro-frontend.

Abstract: After the emergence of microservices, software development has matured a lot in terms of decoupling, we have come to understand the motivations for using this architectural pattern. Therefore, the concept of micro-frontend was born, to solve the same problems as the microservices approach, but in the context of frontend development. With the emergence of micro-frontend, frameworks were born with the proposal to apply this approach and facilitate its implementation, creating an additional layer of study for developers. In this context, this work aims to carry out an empirical study on frontend developers' understanding of essential micro-frontend concepts, exploring benefits, implementation, routing and communication of this approach, through a survey using questionnaires. The main results obtained reveal that developers recognize the benefits of independent deployment. However, approaches like DDD and composing micro-frontends via CDN are poorly understood. Furthermore, dealing with routing and understanding which technologies have this responsibility demonstrated low understanding. By contributing to a more solid understanding of micro-frontend, this study offers opportunities for improvement for both industry and academia, highlighting concepts and best practices in software development to be explored.

¹Graduando em Sistemas de Informação pela Unisinos. Email: felipe.sadler@gmail.com

²Possui doutorado em Informática pela Pontifícia Universidade Católica do Rio de Janeiro (2012), mestrado em Ciência da Computação pela Pontifícia Universidade Católica do Rio Grande do Sul (2008), graduação em Ciência da Computação pela Universidade Federal de Alagoas (2006) e em Tecnologia da Informação pelo Instituto Federal de Alagoas. Email: kleinnerfarias@unisinos.br

Analyze the collected data, filter only professionals with adequate knowledge to make the research result more assertive and finish generating the results of the study.

Key-words: Software Development, Microservices, Micro-frontend.

1 INTRODUÇÃO

Micro-frontends foram introduzidos em 2016 para permitir a decomposição do front-end em front-ends individuais e semi-independentes, separando a lógica de negócios do frontend e criando serviços independentes que interagem entre si (Peltonen; Mezzalira; Taibi, 2021). Também, de acordo com (Hideki, 2021), o projeto é decomposto em pedaços menores e mais simples, que podem ser desenvolvidos, testados e entregues de forma independente. Logo, Micro-frontends são uma arquitetura emergente inspirada na arquitetura de microserviços (Mezzalira, 2021), que por sua vez, compartilham de alguns princípios de arquitetura, podendo citar alguns mencionados por (Barzotto, T. R.; Farias, K), como: Modelar em torno de conceitos de negócios, Descentralizar todas as coisas, Tornar os serviços implantáveis de forma independente e Isolar falha.

Com arquitetura de micro-frontend a base de código é menor e coesa e, portanto, conveniente de manter (Ferracaku, 2021). Portanto, há uma redução concreta da carga cognitiva da equipe em comparação com o manuseio de uma base de código monolítica (Mezzalira, 2021). De acordo com a pesquisa de (Montelius, 2021, conforme citado por M. Geers, 2020), existem várias razões pelas quais as empresas adotam a arquitetura de micro front-end. Uma delas é que otimiza o desenvolvimento de recursos, pois todas as habilidades necessárias para desenvolver um recurso estão incluídas em uma única equipe.

Além disso, a demanda por uma melhor experiência de usuário ao navegar por plataformas web aumentou nestes últimos anos, consequência do aumento de novos serviços e novas concorrências no mercado. Logo, as empresas necessitam otimizar seu tempo de entrega de software, não deixando de lado a qualidade. Portanto o micro-frontend possibilita adicionarmos novas funcionalidades rapidamente, evoluir o negócio, entregando um novo recurso da aplicação de forma autônoma e mitigando possíveis problemas neste fluxo. Logo, existem diversas tecnologias utilizadas no mercado que possibilitam o desenvolvimento de aplicações em micro-frontend. Porém, cada tecnologia leva consigo suas peculiaridades, como forma de implementação e diferentes métodos para resolver o mesmo problema. Portanto, muitos desenvolvedores centralizam seus estudos em determinadas tecnologias, e por muitas

vezes deixam de entender as motivações, conceitos essenciais e decisões que devem ser tomadas antes de iniciar um projeto utilizando tal abordagem.

Após absorção de conceitos essenciais sobre o tema, obtivemos uma visão macro sobre o processo de implementação desta abordagem e definimos alguns pontos como essenciais para o entendimento do desenvolvedor, como: compartilhamento de dados, tipos de divisões disponíveis, abordagem de design de software, comunicação entre micro-frontends, mitigação de bugs, equipes tecnologicamente agnósticas e isolamento de falhas. Portanto, a proposta deste estudo visa avaliar o nível de entendimento dos participantes em relação ao desenvolvimento de aplicações utilizando a abordagem de micro-frontend. Desta forma, estes participantes correspondem aos profissionais de tecnologia que trabalham no Brasil, atuantes na área de desenvolvimento frontend ou com experiência equivalente.

Logo, este artigo apresenta um estudo empírico exploratório sobre o entendimento dos desenvolvedores atuantes em relação à abordagem de micro-frontend. Dessa forma, através do método survey criamos uma pesquisa por questionário com os pontos mencionados anteriormente, aplicamos a escala Likert nas questões para medir o grau de concordância ou discordância em relação às afirmações. E por fim, recolhemos os dados qualitativos para análise e resultados.

O estudo está dividido conforme a seguinte estrutura: a Seção 2 conterá a fundamentação teórica, com os principais conceitos essenciais para o entendimento do estudo proposto; a Seção 3 abordará os trabalhos relacionados, evidenciando o processo de seleção dos artigos selecionados e analisando-os para realizarmos um comparativo destes com o presente proposto. a Seção 4 abordará a descrição da metodologia aplicada neste estudo.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo apresentamos conceitos considerados essenciais para o entendimento sobre micro-frontend, bem como para o método de pesquisa survey.

2.1 A Evolução da arquitetura frontend

Nos últimos anos, o desenvolvimento e a busca por novas melhores práticas cresceram bastante. As aplicações eram desenvolvidas seguindo a arquitetura monolítica, porém esse padrão nos trazia algumas dificuldades técnicas, estimulando o surgimento da arquitetura orientada a microsserviços, derrubando o backend monolítico.

Com a adoção dessa nova arquitetura desenvolvida no backend e absorvendo os benefícios que ela trazia, trouxeram essa mesma abordagem para o ecossistema de desenvolvimento frontend. Esta seção apresenta essas três arquiteturas, de modo a entender sua escala evolutiva até chegarmos à arquitetura de micro-frontend no desenvolvimento web.

Arquitetura Monolítica. Uma arquitetura monolítica trata-se de uma aplicação onde, todos os serviços estão altamente acoplados, ou seja, estão dependentes, compartilham de uma mesma máquina, bem como recursos de processamento, memória, banco de dados e arquivos, como demonstrado na Figura 1, onde podemos visualizar o backend da aplicação e o banco de dados implementados no mesmo contexto. Devido ao fato de todo o desenvolvimento estar em um único executável, uma única alteração de código pode afetar todos os recursos que a aplicação provê, gerando necessariamente um novo *build* e um *redesploy* de toda a aplicação (Barzotto, T. R.; Farias, K, 2022).

Pelo alto acoplamento o código com o crescimento torna-se confuso, de difícil manutenção e com pouca escalabilidade. Mas podemos destacar alguns benefícios também, segundo (HIDEKI, R, 2021):

dentre as principais motivações para a adoção da arquitetura monolítica, estão:

- **Velocidade de desenvolvimento:** a velocidade de entrega de novas funcionalidade é essencial, principalmente para novos projetos e startups;
- **Simplicidade arquitetural:** o projeto deve ser simples de se entender e fácil de ser alterado, de forma que novos membros sejam capazes de se integrar e contribuir rapidamente.

Vale notar que os pontos acima são proeminentes no início do desenvolvimento, e vão diminuindo conforme o projeto aumenta.

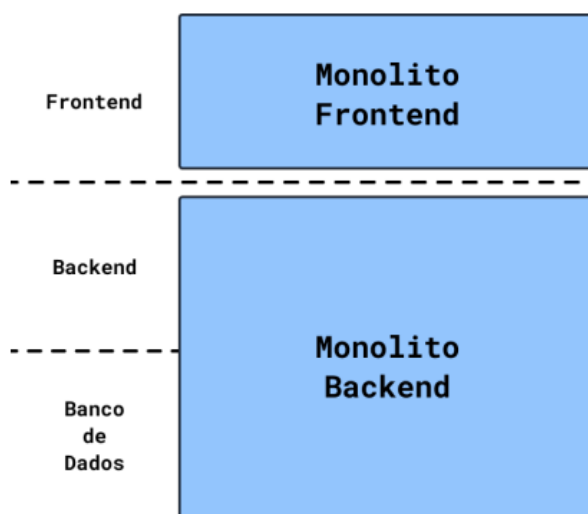


Figura 1 - Abordagem utilizando monolítico no frontend e backend.

Arquitetura Orientada a Microserviços. De acordo com Newman (2020):

Microserviços são serviços que podem ser implantados de forma independente, e são modelados em torno de um domínio de negócio. Eles se comunicam entre si por meio de redes e, como uma opção de arquitetura, oferecem diversos modos de resolver os problemas que você poderá enfrentar.

O principal objetivo desta arquitetura é desacoplar os serviços da aplicação, de modo colaborar com sua escalabilidade, dando mais autonomia aos times, como visualizamos a seguir na Figura 2, onde podemos visualizar a separação do backend em pequenos serviços independentes.

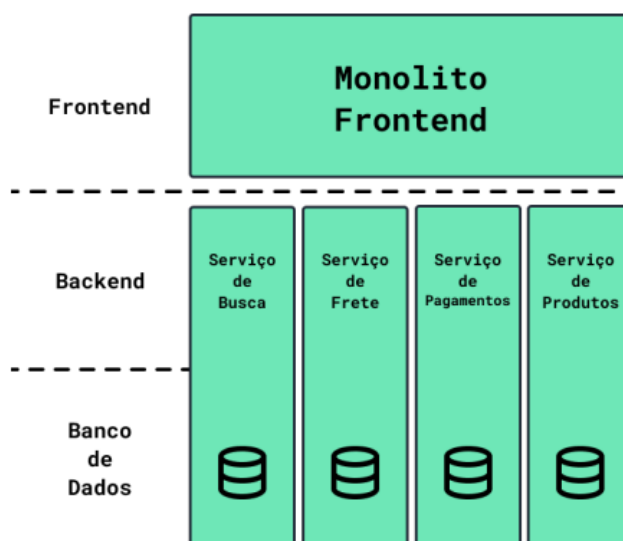


Figura 2 - Abordagem utilizando microsserviços no backend.

Arquitetura de Micro-Frontends. De acordo com (Peltonen; Mezzalana; Taibi, 2021):

Micro-Frontends estende os conceitos de Microsserviços para o frontend da aplicação. Ele transforma aplicativos da Web monolíticos de uma arquitetura de aplicativo baseada em código único em um aplicativo que combina vários pequenos aplicativos de front-end em um todo.

De acordo com Martin Fowler em seu artigo (Fowler, M. 2019), o uso do micro-frontend apresenta um conjunto de benefícios, sendo eles:

- **Atualizações incrementais** – Para muitas organizações, este é o início de sua jornada de micro frontends. O antigo e grande monólito de front-end está sendo retido pela pilha de tecnologia do passado ou pelo código escrito sob pressão de entrega, e está chegando ao ponto em que uma reescrita total é tentadora. Para evitar os perigos de uma reescrita completa, preferimos estrangular o aplicativo antigo peça por peça e, enquanto isso, continuar a fornecer novos recursos aos nossos clientes sem sermos sobrecarregados pelo monólito.
- **Bases de código desacopladas** – O código-fonte de cada micro front-end individual será, por definição, muito menor do que o código-fonte de um único front-end monolítico. Essas bases de código menores tendem a ser mais simples e fáceis de trabalhar para os desenvolvedores.
- **Implantação independente** – Assim como acontece com os microsserviços, a implantação independente de microfront-ends é fundamental. Isto reduz o escopo de qualquer implantação, o que, por sua vez, reduz o risco associado.
- **Equipes autônomas** - Como um benefício de ordem superior de dissociar nossas bases de código e nossos ciclos de lançamento, percorremos um longo caminho para ter equipes totalmente independentes, que podem possuir uma seção de um produto desde a concepção até a produção e além.

Logo, a ideia principal é desacoplar serviços, criando mais autonomia a nível de desenvolvimento, testes e entrega de software. Como podemos visualizar na Figura 3, onde demonstramos a aplicação frontend dividida em pequenos serviços independentes.

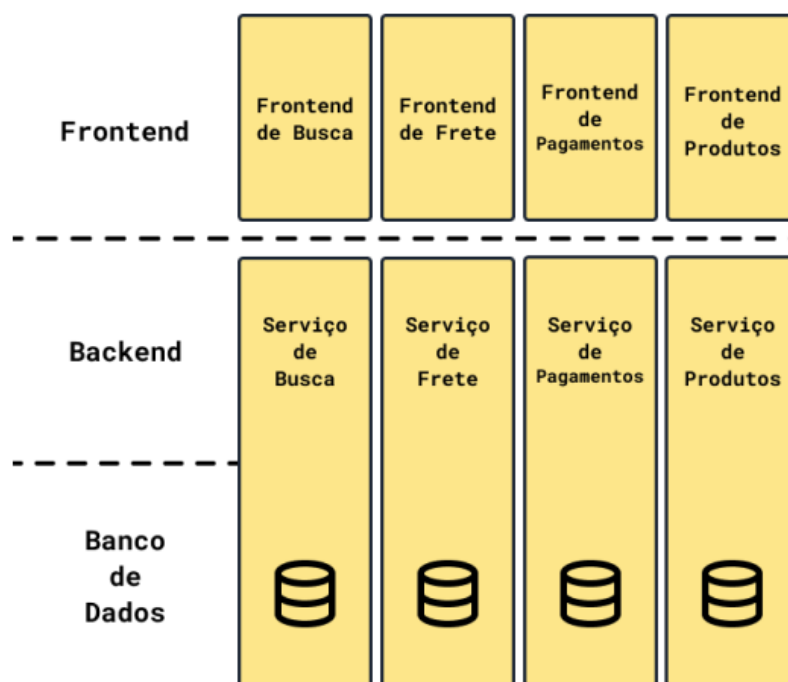


Figura 3 - Abordagem utilizando micro-frontend.

Composição de Micro-frontends. De acordo com (MEZZALIRA, 2021), algumas decisões arquitetonicas precisarão ser tomadas antecipadamente, pois direcionarão decisões futuras, como definir um micro-frontend, como orquestrar as diferentes visualizações, como compor a visualização final para o usuário e como os micro-frontends irão se comunicar e compartilhar dados. Logo, estas decisões são compostas por quatro áreas principais:

1ª Definir: Esta primeira decisão trata-se de definir o que são micro-frontend em nosso contexto do projeto, basicamente existem duas formas de definir:

- **Divisão horizontal:** Neste modelo consideramos *vários micro-frontends na mesma página*, como mostramos na Figura 4, onde demonstramos a divisão de times na aplicação (Team A, Team B e Team C), esta opção nos entrega um grau maior de complexidade no projeto, pois entre os serviços, possivelmente vamos precisar nos preocupar com a comunicação entre eles, bem como suas dependências, isto nos leva a pensar em qual das diversas alternativas no mercado se enquadra melhor em nosso projeto.

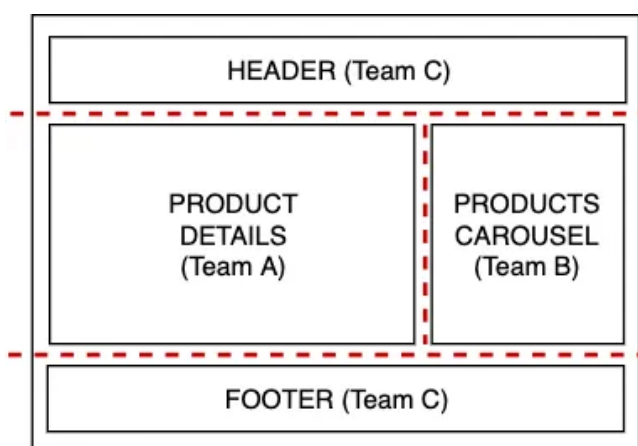


Figura 4 - Abordagem utilizando micro-frontend de divisão horizontal.

- **Divisão vertical:** Neste modelo consideramos um *único micro-frontend por página*, esta opção é a mais tradicional, pois qualquer comunicação ou dependências resolvemos no mesmo projeto, como mostramos na Figura 5, onde cada time (Team A e Team B) é responsável por uma página/serviço, neste cenários os times costumam aplicar a modelagem de DDD (Domain-Driven Design), onde cada equipe fica responsável por um domínio de negócio, como autenticação ou experiência de pagamento.

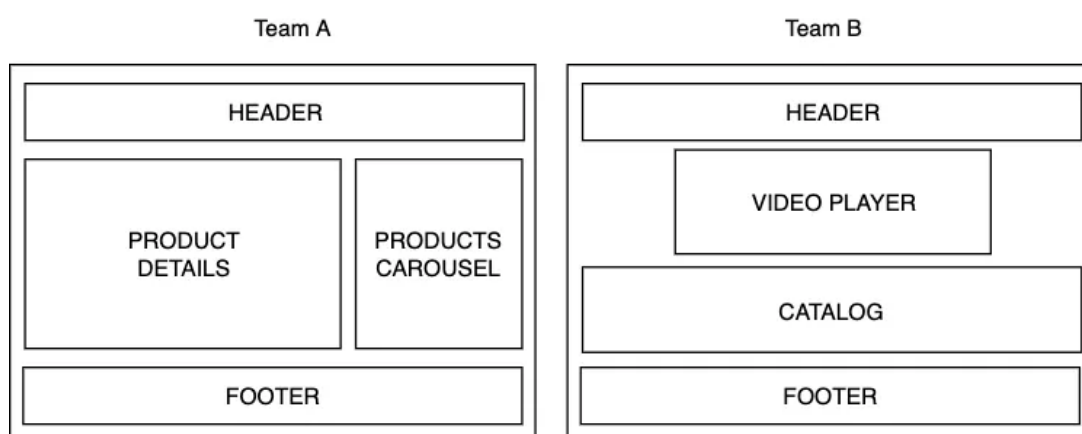


Figura 5 - Abordagem utilizando micro-frontend de divisão vertical.

2ª Compor: Esta segunda decisão trata-se de definir como compor o micro-frontend, considerando que neste momento foi escolhido o modelo de divisão do projeto, temos algumas alternativas.

Caso a escolha tenha sido *divisão vertical*, essa decisão não é necessária, pois para cada página entregamos apenas um único serviço, ou seja, completo, sem a necessidade de nos preocuparmos com a composição de demais serviços.

Caso a escolha tenha sido *divisão horizontal*, temos três formas de compor os serviços, sendo possível adotar apenas uma ou combiná-las. Na Figura 6 demonstramos os três modelos de composição, onde visualizamos a composição ocorrendo.

- **Client-side:** Neste modelo a composição ocorre no cliente, no navegador do usuário.
- **Edge-side:** Neste modelo a composição ocorre a nível de CDN.
- **Server-side:** Neste modelo a composição ocorre a nível de servidor.

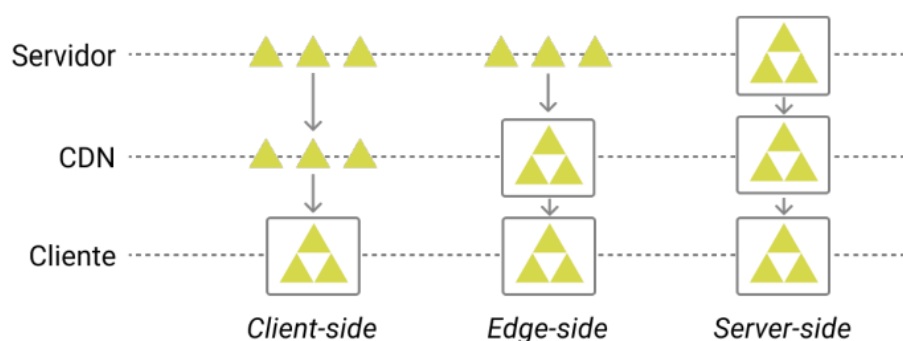


Figura 6 - Abordagens de composição para divisão horizontal.

3ª Rotear: Esta terceira decisão trata-se de definir como cada micro-frontend deve ser acessado.

Caso a escolha tenha sido *divisão vertical*, é mais simples, pois podemos lidar com as mudanças de serviços apenas alterando a URL, considerando que cada URL representa uma página/serviço.

Caso a escolha tenha sido *divisão horizontal*, precisamos de um roteador, para se responsabilizar em orquestrar todos os serviços nas páginas, seja a nível de cliente, CDN ou de servidor.

Existem algumas opções no mercado para lidar com essa questão, como a framework Single-SPA para lidar com roteamento a nível de cliente, ou o Module Federation (funcionalidade disponível a partir da versão 5 do webpack) para lidar com o roteamento a nível de servidor ou o CloudFront (serviço oferecido pela AWS) para lidar com o roteamento a nível de CDN. Vale ressaltar que estas soluções mencionadas anteriormente resolvem não

apenas a questão do roteamento, mas também a questão da comunicação, que iremos explicar a seguir.

4ª Comunicar: Esta quarta e última decisão trata-se de definir como os micro-frontends/serviços irão se comunicar.

Caso a escolha tenha sido *divisão horizontal*, realizamos a comunicação a nível de cliente, no navegador do usuário, uma opção seria utilizar a emissão de eventos nativos do navegador, porém como mencionado anteriormente existem ferramentas disponíveis que resolvem este mesmo problema.

Caso a escolha tenha sido *divisão vertical*, realizamos a comunicação a nível de cliente, no navegador do usuário, uma das opções seria utilizar o armazenamento da web, como o localStorage ou o sessionStorage, ou podemos utilizar strings de consultas, repassando dados de uma página para outra.

2.2 Método de pesquisa survey

O método de pesquisa survey é um tipo de investigação quantitativa e tem como objetivo a coleta de dados sobre características, ações ou opiniões de determinado grupo de pessoas.

De acordo com (Freitas, H. et al., 2000), “a survey é **apropriada** como método de pesquisa quando se deseja responder questões do tipo “o quê?”, “por que?”, “como?” e “quanto?”, ou seja, quando o foco de interesse é sobre “o que está acontecendo” ou “como e por que isso está acontecendo””.

Quanto ao **propósito** elencado por (Freitas, H. et al., 2000), podemos listar:

- **Explanatória** - Tem como objetivo testar uma teoria e as relações causais;
- **Exploratória** - Tem como objetivo familiarizar-se com o tópico ou identificar os conceitos iniciais sobre um tópico, dar ênfase na determinação de quais conceitos devem ser medidos e como devem ser medidas;
- **Descritiva** - Busca identificar quais situações, eventos, atitudes ou opiniões estão manifestos em uma população.

Coleta de Dados. Os dois meios mais comuns de coleta de dados são questionários e entrevistas. Os questionários podem ser fornecidos em papel ou em algum formato eletrônico, por exemplo, email ou páginas da web. (Wohlin, C., 2012). Logo, em nossa realidade podemos utilizar redes sociais, onde conseguimos atingir um número consistente de

participantes. Nesta pesquisa utilizamos o survey, bem como coletamos dados através de questionários, pois torna-se possível obter uma amostra representativa, ou seja, nos possibilita a seleção de uma ampla gama de participantes, de forma relativamente rápida e eficiente. Logo, com base nas respostas coletadas, conseguimos obter uma compreensão mais abrangente das opiniões dos participantes.

3 TRABALHOS RELACIONADOS

Fizemos buscas em repositórios digitais como Google Acadêmico e Elsevier. Selecionamos trabalhos dentro de um range de 5 anos, posteriormente eles foram analisados e comparados.

3.1 Análise de trabalhos relacionados

(Peltonen; Mezzalira; Taibi, 2021). Este trabalho aborda uma pesquisa qualitativa, onde foi avaliado o conhecimento existente de profissionais do setor de tecnologia sobre micro-frontend, com o objetivo de entender as motivações, benefícios e problemas que profissionais e empresas da indústria tem ao aplicar tal abordagem. Devido a falta de amadurecimento sobre o assunto a nível de publicações foi incluída como fonte de pesquisa as literaturas cinzentas, utilizando mecanismos de pesquisas como o Google Search, Twitter Search, Reddit Search, Medium e resultando em conteúdos como postagens de blog, fóruns, sites, white paper e livros, e também foi considerado como fonte de pesquisa literaturas acadêmicas, como Google Scholar, Science Direct, IEEEExplore Digital Library, ACM digital Library e Scopus. Para lidar com a quantidade de resultados de diversas fontes, adotaram o processo de Revisão Sistemática da Literatura Multivocal (MLR), criando critérios de qualidade para a seleção dos conteúdos. Como resultado da pesquisa podemos destacar o aumento da complexidade do frontend como principal motivador de sua adoção, o suporte para diferentes tecnologias como principal benefício do micro-frontend e identificado a consistência de UX como principal possível problema em sua adoção.

(Hideki, R 2021). Este trabalho apresenta o caso da empresa Elo7, onde passaram por uma mudança arquitetural, anteriormente eles tinham um grande acoplamento de serviços, um monolito escrito em Java no backend, bem como um monolito dedicado ao frontend, com esta realidade eles tiveram problemas como a experiência de desenvolvimento, tecnologias

descontinuadas e o próprio acoplamento, a equipe passou a adotar a abordagem de microsserviços, com esta mudança os desenvolvedores puderam aproveitar este novo conhecimento para aplicar o micro-frontend, considerando que ambos seguem a ideia principal, desacoplar os serviços em pequenos pedaços. Após aplicada as mudanças no backend e no frontend, foi aplicado um questionário aos desenvolvedores com o intuito de avaliar o impacto trazido pela nova arquitetura, fizeram uma análise quantitativa, aplicando a escala Likert com questões de múltipla escolha, avaliando o perfil dos participantes, após foi realizada uma análise qualitativa, avaliando a expectativa dos participantes sobre as mudanças arquiteturais ocorridas, passando por tecnologias e arquitetura antigas, tecnologias e arquitetura novas e por último avaliando o entendimento sobre a definição de micro-frontends. Como resultado foi destacado o efeito positivo no contexto de entendimento do projeto e melhoria nos deploys.

(Ferracaku, 2021). Este trabalho tem como objetivo dar uma introdução a arquitetura de micro-frontend, mostrando casos reais de implementação na indústria, onde conseguimos perceber as vantagens e desvantagens desse padrão de arquitetura em cada contexto. Este estudo aborda uma pesquisa qualitativa onde visa avaliar como a abordagem do micro-frontend é percebida por desenvolvedores com diferentes níveis de experiência em relação a uma abordagem mais tradicional de desenvolvimento de aplicação frontend monolítica. Para a devida análise os participantes inicialmente respondem um questionário expondo seu nível de senioridade, bem como sua experiência com o desenvolvimento web utilizando a arquitetura de micro-frontend, posteriormente cada participante é solicitado a configurar uma arquitetura de frontend simples, usando duas abordagens, utilizando um aplicativo de página única e outro utilizando micro-frontend, compartilhando dados extraído de cada cenário num questionário posterior.

(Montelius, 2021). Este trabalho tem como objetivo investigar como o desenvolvimento web é afetado pela adoção da abordagem de micro-frontend, tanto no aspecto arquitetônico quanto no organizacional. Para alcançar este objetivo a pesquisa de natureza exploratória é dividida em duas partes, a primeira é realizada a coleta dados de participantes da área de software que trabalham na empresa para a qual esta pesquisa foi realizada, através de três questões gerais, a primeira questão visa identificar os efeitos positivos e negativos ao adotar a abordagem de micro-frontend, a segunda questão visa validar o suposto aumento de modificabilidade num projeto utilizando micro-frontend e a terceira e última questão visa identificar em quais

cenários faria sentido utilizar a abordagem de micro-frontend em um projeto. A segunda parte desta pesquisa visa estimar o nível de modificabilidade entre dois projetos, para isto foi realizada a implementação de dois protótipos de aplicação, um utilizando a técnica de Single Page Applications (SPA) e outro a abordagem de micro-frontend, para realizar esta comparação foi utilizado um modelo matemático chamado SQMMA.

3.2 Análise comparativa dos trabalhos relacionados

Critério de comparação. Foram definidos quatro Critérios de comparação (CC) para realizar a comparação de similaridades e diferenças entre o trabalho proposto e os artigos selecionados, descritos a seguir:

- **CC1: Contexto** - Estudos que envolvem participantes com experiência na indústria.
- **CC2: Perfil dos participantes** - Estudos que analisam o perfil dos participantes.
- **CC3: Domínio** - Estudos que consideram participantes atuando em diferentes domínios da indústria.
- **CC4: Método de pesquisa** - Estudos que realizaram surveys.
- **CC5: Nível de conhecimento** - Estudos que analisaram o nível de conhecimento dos participantes sobre micro-frontend.
- **CC6: Nível de percepção** - Estudos que analisam o nível de percepção de utilidade de características ou conceitos de micro-frontend.

Oportunidade de pesquisa. A Tabela 1 apresenta a comparação dos estudos selecionados, evidenciando que nenhum deles, exceto o trabalho proposto, atende aos critérios definidos. Com base nessas comparações, foram identificadas as seguintes oportunidades: (1) apenas o trabalho proposto atende a todos os critérios de comparação estabelecidos; (2) tanto o trabalho proposto quanto o trabalho de (Ferracaku, 2021) investigam previamente o nível de conhecimento dos participantes sobre micro-frontend; (3) nenhum estudo empírico se concentra no entendimento dos conceitos essenciais da abordagem de micro-frontend no

desenvolvimento web frontend. Portanto, identificou-se a oportunidade de pesquisar o nível de percepção que os desenvolvedores frontend têm em relação aos conceitos essenciais para a implementação de uma aplicação utilizando a abordagem de micro-frontend.

Tabela 1 - Análise comparativa dos Trabalhos Relacionados selecionados

Trabalho Relacionado	Critério de Comparação					
	CC1	CC2	CC3	CC4	CC5	CC6
Trabalho Proposto	●	●	●	●	●	●
(Peltonen; Mezzalira; Taibi, 2021)	●	○	●	○	○	●
(Hideki, R, 2021)	●	●	○	●	○	●
(Ferracaku, 2021)	●	●	●	●	●	○
(Montelius, 2021)	●	○	○	●	○	●

● Atende ○ Não Atende

4 METODOLOGIA

Esta seção apresenta a metodologia de pesquisa seguida para a realização deste projeto, formada pelas seguintes seções. A Seção 4.1 apresenta o objetivo principal e as questões de pesquisa. A Seção 4.2 descreve o processo experimental adotado. A Seção 4.3 descreve o questionário formulado aplicado na pesquisa. A metodologia adotada foi baseada em estudos anteriores publicados (Júnior; Farias; Silva, 2022; Farias, K. et al, 2018; Júnior, E.; Farias, K.; Silva, B, 2021).

4.1 Objetivo e questões de pesquisa

O objetivo deste estudo trata-se de entender o nível de compreensão dos desenvolvedores em relação ao micro-frontend, para alcançar esse objetivo preparamos 26 questões de estudo. O Apêndice B descreve as questões elaboradas.

4.2 Processo experimental

A Figura 7 apresenta o processo experimental adotado, dividida em três etapas:

Fase 1: Seleção dos participantes - Seleccionamos os participantes com base no seguinte critério: atuantes na área, com experiência prática no setor de desenvolvimento frontend. A população alvo compreende profissionais de tecnologia que trabalham no Brasil, mais especificamente engenheiros de software com diferentes níveis de senioridade, sendo indiferente ter concluído ou ter formação acadêmica. Essa população representa aqueles que estão em posição de responder nosso questionário e a quem os resultados da pesquisa se aplicam.

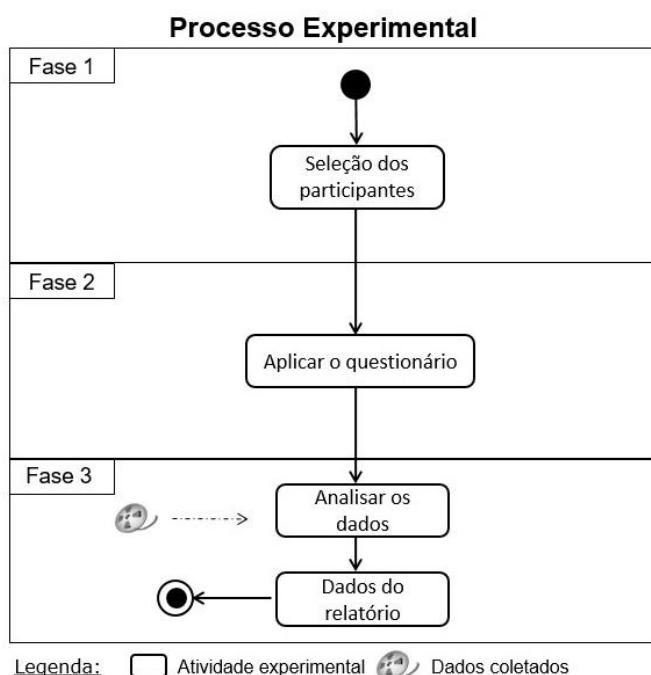


Figura 7 - Processo Experimental.

Fase 2: Aplicar questionário - O experimento não foi realizado em um local específico, com duração aproximada de duas semanas, aplicamos a pesquisa no formato online, enviamos o acesso ao questionário de pesquisa (discutido na Seção 4.3) através de grupos de alunos da universidade presente no aplicativo WhatsApp e profissionais do setor tecnológico presentes em redes profissionais, como o LinkedIn. Dividimos o questionário em duas partes.

Na primeira parte do questionário, coletamos informações relacionadas à experiência dos participantes em desenvolvimento frontend. O objetivo dessa etapa é analisar o perfil dos participantes, ajudando-nos a compreender melhor a diversidade de conhecimento e experiência dentro do grupo.

Na segunda parte do questionário criamos vinte seis questões de múltipla escolha utilizando uma escala Likert, com objetivo de analisar o nível de conhecimento dos desenvolvedores em relação a sua familiaridade com o desenvolvimento prático utilizando a abordagem de micro-frontend.

Fase 3: Análise de dados - Nesta etapa, procedemos à análise dos dados coletados por meio do questionário. Durante o experimento, coletamos dois tipos de dados dos participantes.

A partir dos resultados da primeira pesquisa, na qual aplicamos as questões do Apêndice A, obtivemos dados qualitativos que quantificaram a experiência e a familiaridade dos participantes com o desenvolvimento frontend.

Na segunda pesquisa, na qual aplicamos as questões do Apêndice B, obtivemos dados qualitativos que avaliaram o nível de conhecimento perceptível dos participantes em relação à abordagem de micro-frontend. Realizamos a análise manual dos dados dos questionários, passando de uma visão ampla para uma mais focal, sem divergências.

4.3 Questionário e seleção dos participantes

Os dados do experimento foram coletados a partir de um questionário desenvolvido no aplicativo Google Forms. Os participantes puderam mensurar seu nível de experiência e conhecimento sobre micro-frontend através das questões aplicadas. O Apêndice A apresenta a lista das perguntas utilizadas nesta primeira etapa. Os resultados desta primeira etapa classificaram os participantes para a segunda análise do estudo.

Nesta segunda etapa, elaboramos questões para validar o conhecimento dos participantes, objetivo central desta pesquisa, onde foi baseado em estudos empíricos explorados anteriormente, como o estudo (Peltonen; Mezzalira; Taibi, 2021), bem como numa das principais obras publicadas sobre a temática (Building Micro-Frontends).

As perguntas elaboradas foram separadas em **quatro seções**, com o objetivo de ajudar a organizar e permitir uma análise mais aprofundada dos resultados, sendo elas:

- **Sobre benefícios do uso de micro-frontend:** Esta seção permite que os participantes avaliem seu nível de concordância ou discordância com afirmações relacionadas aos benefícios percebidos do uso de micro-frontends, o que ajudará a quantificar a importância relativa de cada benefício.

- **Sobre implantação de micro-frontend:** Esta seção permite que os participantes avaliem seu nível de concordância ou discordância com afirmações relacionadas a certos conceitos, práticas ou ferramentas comuns de implantação que funcionam para este padrão de implantação.
- **Sobre roteamento entre micro-frontend:** Esta seção permite que os participantes avaliem seu nível de concordância ou discordância com afirmações relacionadas ao roteamento de micro-frontends.
- **Sobre comunicação entre micro-frontend:** Esta seção permite que os participantes avaliem seu nível de concordância ou discordância com afirmações relacionadas a comunicação entre micro-frontends.

Seleção dos Participantes. Para a seleção dos participantes, utilizamos o LinkedIn como fonte principal para identificar potenciais candidatos. Devido à natureza da nossa pesquisa optamos por um critério de conveniência, escolhendo indivíduos que são membros do grupo "Front End Brasil" no LinkedIn, uma comunidade voltada para profissionais de desenvolvimento frontend. A escolha desse grupo se baseou em sua relevância para o escopo da pesquisa.

Cálculo Amostral. O tamanho amostral para esta pesquisa foi determinado levando em consideração a população disponível, que é estimada em 12.000 membros do grupo "Front End Brasil" no LinkedIn. Para garantir a representatividade dos resultados, consideramos um nível de confiança de 90% e um erro amostral de 18%. Com base nesses parâmetros, definimos um tamanho amostral de 21 pessoas. Esses parâmetros foram escolhidos para atender aos requisitos de precisão e confiabilidade de nossa pesquisa.

5 RESULTADOS

Esta seção apresenta os resultados obtidos para as questões de pesquisa (descritas na Seção 4.1). Foram usados histogramas para prover uma visão geral dos dados coletados dos 22 participantes da pesquisa.

5.1 Análise de perfil dos participantes

A Tabela 2 resume o perfil dos participantes, reportando diversas informações como faixa etária, gênero, escolaridade, graduação, tempo estudo em universidade, experiência em desenvolvimento de software e experiência com desenvolvimento frontend. Como algumas questões não eram obrigatórias, a soma das respostas não necessariamente se equivale ao número total de participantes (22).

Faixa etária. A maioria (54.5%) dos participantes estão na faixa de 24 a 29 anos, representando um público relativamente jovem. Além disso, 18.2% dos participantes informaram ter entre 36 e 41 anos, trazendo assim uma variedade boa para a pesquisa.

Gênero. A maioria dos participantes (77.3%) informaram serem do gênero masculino. Essa descoberta ressalta uma predominância significativa da representação masculina na amostra pesquisada.

Escolaridade. A maioria dos participantes (86.4%) informaram terem concluído alguma graduação. Além disso, 13.6% dos participantes informaram terem alcançado o nível técnico, como seu nível mais elevado de educação.

Graduação. A maioria dos participantes (40%) informaram cursam ou cursaram tecnólogo na área de TI, pelo curso de análise de sistemas, outros 45% dos participantes informaram outros cursos, sendo muitos fora da área de TI, esse fenômeno indica uma tendência crescente de pessoas que, embora não tenham uma formação acadêmica tradicional em tecnologia, estão adentrando e contribuindo para o setor de desenvolvimento de software.

Tempo estudado em universidade. A maioria dos participantes (70%) estudou por um período de 2 a 6 anos, enquanto uma minoria significativa (25%) estudou por menos de 2 anos. Notavelmente, nenhum dos participantes estudou por um período de 7 a 8 anos, e apenas um indivíduo (5%) dedicou mais de 8 anos à sua formação universitária. Esses resultados demonstram uma diversidade de trajetórias educacionais entre os participantes da pesquisa.

Experiência em desenvolvimento de software. A maioria dos participantes (40%) possui menos de 2 anos de experiência, enquanto 30% têm entre 2 e 4 anos de prática. Notavelmente, 20% dos participantes possuem entre 5 e 8 anos de experiência, enquanto outros 10% têm mais de 8 anos na área. Essa distribuição reflete uma variedade de níveis de expertise, contribuindo para uma compreensão abrangente do campo de desenvolvimento de software.

Experiência em desenvolvimento frontend. A maioria dos participantes (45%), têm menos de 2 anos de experiência, enquanto 30% têm entre 2 e 4 anos de prática. Além disso, 15% dos participantes acumularam experiência de 5 a 8 anos na área.

Tabela 2 - Dados de perfil dos participantes

Característica	Resposta	#	%
Faixa etária	18 - 23 anos	3	13.6%
	24 - 29 anos	12	54.5%
	30 - 35 anos	3	13.6%
	36 - 41 anos	4	18.2%
Gênero	Masculino	17	77.3%
	Feminino	5	22.7%
Escolaridade	Técnico	3	13.6%
	Graduação	19	86.4%
	Mestrado	1	4.5%
	Doutorado	0	0%
Graduação	Sistemas de Informação	0	0%
	Ciência da computação	3	15%
	Engenharia da Computação	0	0%
	Análise de Sistemas	8	40%
	Outros	9	45%
Tempo estudado em universidade	Menos de 2 anos	5	25%
	De 2 a 4 anos	7	35%
	De 5 a 6 anos	7	35%
	De 7 a 8 anos	0	0%
	Mais de 8 anos	1	5%
Experiência em desenvolvimento de software	Menos de 2 anos	8	40%
	De 2 a 4 anos	6	30%
	De 5 a 6 anos	2	10%

	De 7 a 8 anos	2	10%
	Mais de 8 anos	2	10%
Experiência em desenvolvimento frontend	Menos de 2 anos	9	45%
	De 2 a 4 anos	6	30%
	De 5 a 6 anos	2	10%
	De 7 a 8 anos	1	5%
	Mais de 8 anos	2	10%

5.2 Análise de resultados

Nas próximas seções, apresentaremos os resultados de nossa pesquisa sobre o entendimento dos desenvolvedores sobre micro-frontends. Exploraremos os benefícios percebidos, as estratégias de implantação, as práticas de roteamento e os desafios de comunicação entre micro-frontends. Essas descobertas fornecerão uma visão abrangente do estado atual da adoção e das percepções dos desenvolvedores em relação a essa arquitetura emergente. Cada participante indicou o grau de concordância com as afirmações propostas, de acordo com a escala Likert.

Sobre benefícios do uso de micro-frontend. A Figura 8 apresenta os dados coletados referentes a esses benefícios.

O desacoplamento de serviços, utilizando a abordagem de micro-frontends, apontou ainda gerar dúvidas nos participantes, uma vez que 55% das respostas afirmaram que a abordagem tende a acoplar os serviços. Por outro lado, os resultados apontaram um forte entendimento do benefício da geração de processos de implantações independentes, com 80% de concordância entre os participantes. Esse resultado está alinhado com a descoberta do trabalho relacionado de (Ferracaku, 2021), que também identificou benefícios similares, incluindo um maior controle nas implantações e uma maior facilidade na manutenção do sistema. Além disso, o benefício da governança descentralizada e a geração de equipes especialistas no domínio do negócio demonstraram ser amplamente conhecidos, com 80% de concordância entre os participantes. Esse resultado está alinhado com a descoberta do trabalho relacionado de Montelius (2021), uma vez que ambos identificaram o benefício da geração de equipes especialistas.

Vale ressaltar que uma média de 20% das respostas foi neutra em relação às afirmações propostas, sugerindo que uma parte pequena da amostra não tinha conhecimento suficiente para tomar uma posição em relação a essas afirmações.

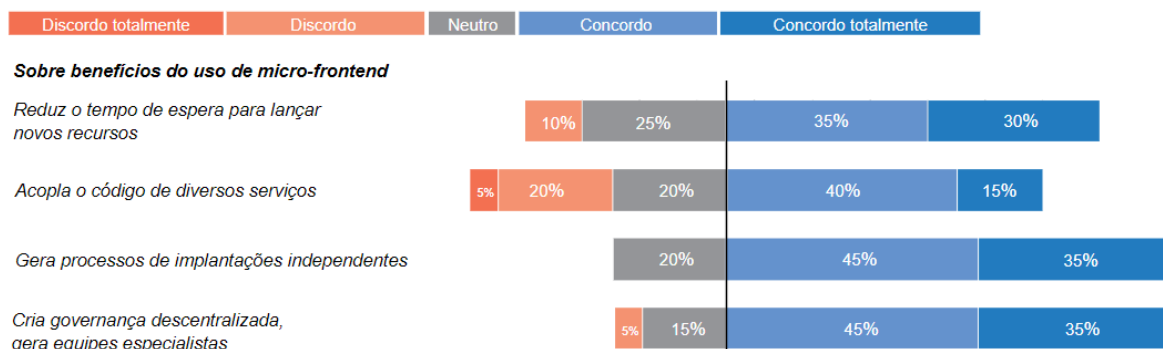


Figura 8 - Sobre benefícios do uso de micro-frontend

Sobre implantação de micro-frontend. A Figura 9 apresenta os dados coletados referentes a seção de implantação de micro-frontend. Os resultados apontaram que a utilização do DDD (Domain-Driven Design) dentro do contexto de micro-frontends demonstrou ser pouco conhecida, com 55% de respostas neutras.

Além disso, a composição de micro-frontends na camada de CDN demonstrou ser pouco conhecida também, com 55% de respostas neutras. Por outro lado, as tecnologias tradicionais para o desenvolvimento de serviços web, como VueJS, ReactJS, NextJS ou AngularJS, demonstraram serem amplamente conhecidas para o desenvolvimento de micro-frontends isolados, com 80% de concordância. Além disso, a utilização de iframes para carregar micro-frontends, demonstrou ser bem conhecida, com 65% de concordâncias.

Em geral, os participantes demonstraram ter pouco conhecimento sobre determinadas tecnologias aplicadas ao contexto de implantação de micro-frontend, como Web Components, ESI (Edge Side Includes) para composição dos micro-frontends na camada de CDN, utilização de SSI (Server-Side Include) para composição dos serviços na camada de servidor e a utilização de um shell no lado do cliente para orquestrar os serviços, com uma média de 44,17% de concordância assertiva.

Vale ressaltar que obtivemos uma média de 43.33% de respostas neutras com as afirmações propostas, esse resultado sugere que uma parte significativa da amostra não tinha conhecimento suficiente para tomar uma posição em relação às afirmações.

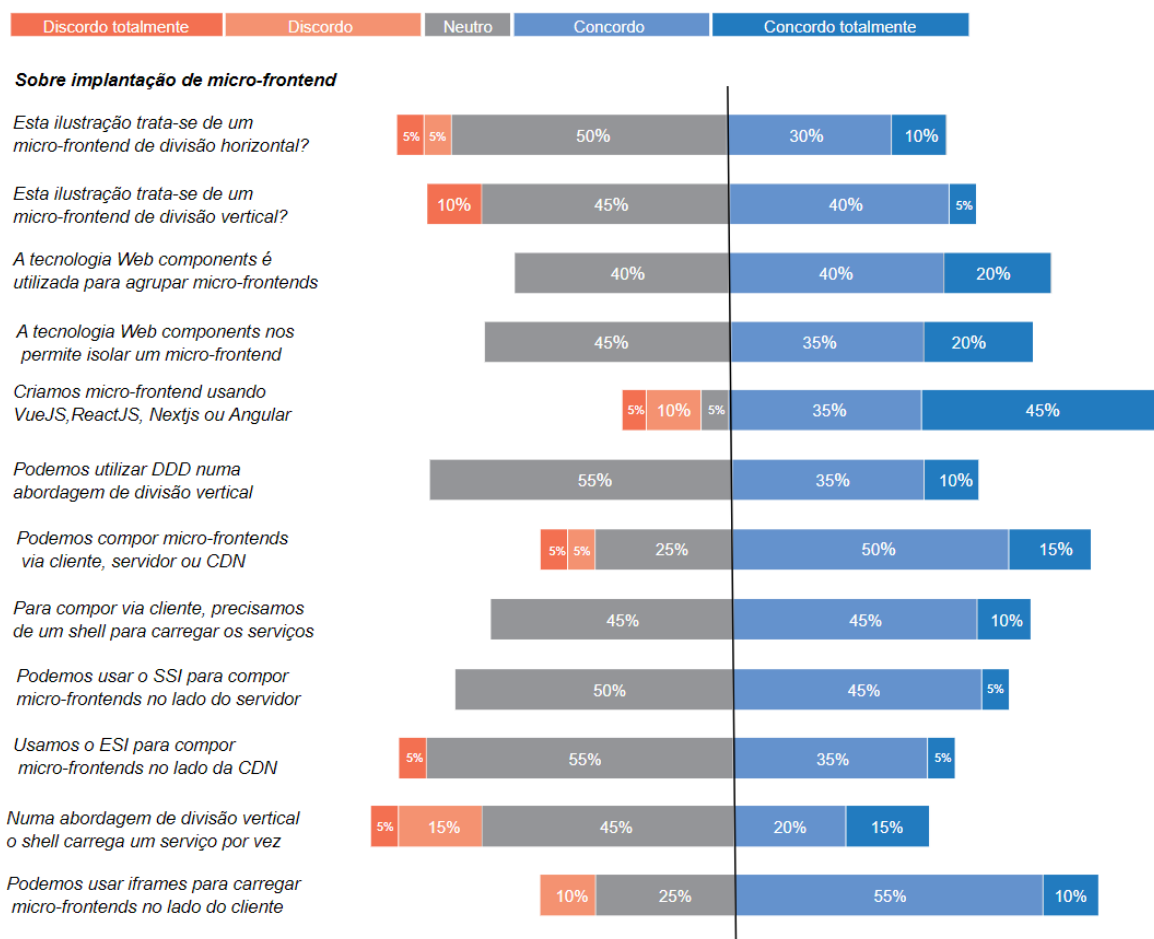


Figura 9 - Sobre implantação de micro-frontend

Sobre roteamento entre micro-frontend. A Figura 10 apresenta os dados coletados referentes a seção de roteamento entre micro-frontend. Cerca de 55% dos participantes indicaram que possuem um entendimento sobre a possibilidade de utilizar solicitações HTTP para realizar o roteamento dos micro-frontends. Isso sugere que uma parcela significativa da amostra não está ciente da capacidade de direcionar o tráfego para diferentes micro-frontends por meio de solicitações HTTP.

Além disso, foi validado que poucos participantes tinham conhecimento sobre tecnologias utilizadas dentro do contexto de micro-frontend para lidar com roteamento de serviços, pois obtivemos uma média de 26,67% de concordância assertivas sobre o papel de tecnologias como Web Components, SingleSPA e Module Federation.

Vale ressaltar que obtivemos uma média de 50% de respostas neutras com as afirmações propostas, esse resultado sugere que uma parte significativa da amostra não tinha conhecimento suficiente para tomar uma posição em relação às afirmações.

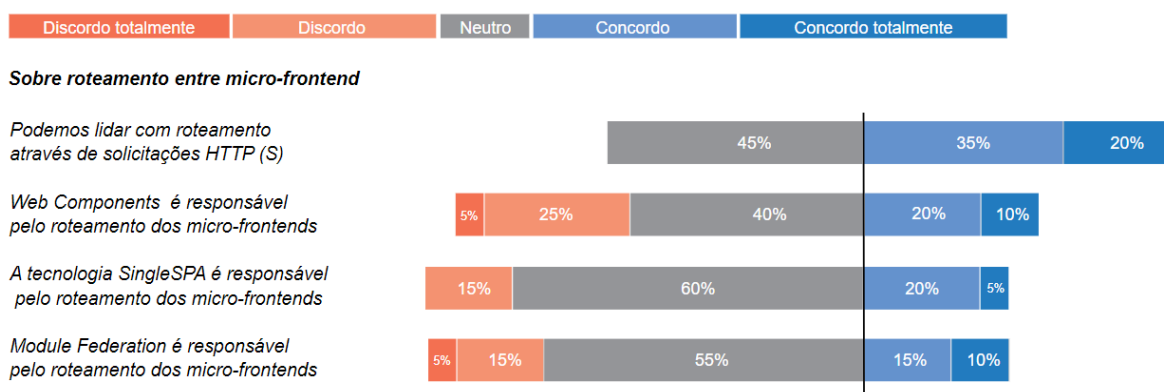


Figura 10 - Sobre roteamento entre micro-frontend

Sobre comunicação entre micro-frontend. A Figura 11 apresenta os dados coletados referentes a seção de comunicação entre micro-frontend. Os resultados apontaram que o princípio da Implantação Independente aplicada ao contexto de micro-frontend demonstrou ser bastante conhecido, com 65% de concordância e 10% de discordância. Isto sugere que a maioria dos participantes tem consciência que cada micro-frontend segue o mesmo princípio de micro-serviços, onde cada micro-frontend deve estar inconsciente dos outros demais existentes. Por outro lado, o mecanismo de comunicação entre componentes chamado Event Bus demonstrou ser pouco conhecido, com 60% de respostas neutras. Também, a utilização de strings de consultas obteve um grau pequeno de entendimento entre os participantes, com 45% de respostas neutras e 15% de discordância. No entanto, a utilização de eventos customizados do javascript demonstrou serem amplamente conhecidos, com 70% de concordância entre os participantes. Além disso, a utilização da API de armazenamento na web, como localStorage ou sessionStorage, demonstrou ter um bom nível de entendimento, com 60% de concordância.

Vale ressaltar que obtivemos uma média de 30% de respostas neutras com as afirmações propostas, esse resultado sugere que uma parte pequena da amostra não tinha conhecimento suficiente para tomar uma posição em relação às afirmações.

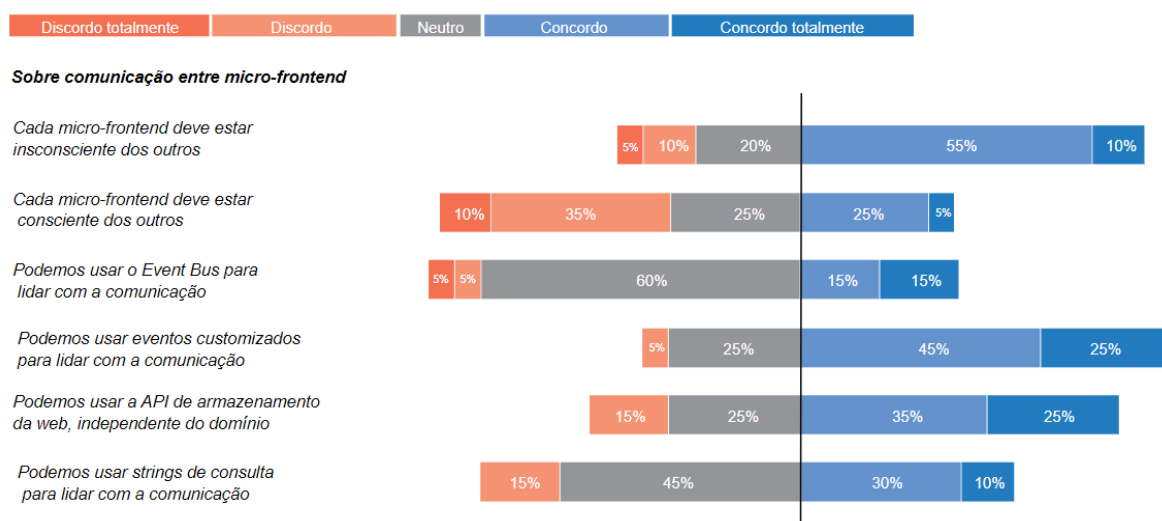


Figura 11 - Sobre comunicação entre micro-frontend

5.3 Discussão adicional

O papel crítico do desacoplamento em micro-frontends. Esta pesquisa investigou o entendimento dos participantes em relação ao desacoplamento dos serviços ao utilizar a abordagem de micro-frontend. O resultado desta investigação apontou que 55% dos participantes não reconhecem este benefício. Esta informação tem implicações significativas tanto para a indústria quanto para a pesquisa acadêmica. Na indústria, a falta de reconhecimento do desacoplamento de serviços pode sugerir a necessidade de uma conscientização e educação mais amplas sobre os benefícios do micro-frontend. Isso pode levar as organizações a não otimizar plenamente essa abordagem de desenvolvimento, perdendo oportunidades de melhorar a escalabilidade, manutenção e a agilidade de suas aplicações.

Do ponto de vista acadêmico, esses resultados indicam uma possível área de estudo adicional. Pesquisas futuras podem se concentrar em aprofundar a compreensão das razões subjacentes à falta de reconhecimento dos benefícios do desacoplamento de serviços no contexto do micro-frontend.

Impulsionando a agilidade com implantações independentes em micro-frontends. Esta pesquisa investigou o entendimento dos participantes em relação ao benefício de gerar processos de implantações independentes ao utilizar a abordagem de micro-frontend. O resultado desta investigação apontou que 80% dos participantes reconhecem este benefício. Isto está alinhado com o resultado da pesquisa de Jonada

Ferracaku (Ferracaku, 2021), que identificou o controle de implantação como uma das principais vantagens do uso do micro-frontend. Esta informação tem implicações significativas tanto para a indústria quanto para a pesquisa acadêmica. Na indústria, o reconhecimento generalizado da vantagem de processos de implantações independentes destaca a eficácia do micro-frontend em melhorar a agilidade e a confiabilidade do desenvolvimento de software. Organizações podem considerar a integração do micro-frontend como uma estratégia para otimizar seus processos de implantação e fornecer atualizações mais rápidas e precisas das aplicações.

Do ponto de vista acadêmico, a alta taxa de reconhecimento deste benefício sugere que o micro-frontend é uma área de pesquisa relevante e que suas vantagens práticas são amplamente percebidas pela indústria. Portanto, pesquisas futuras podem se concentrar em questões mais detalhadas, como as melhores práticas para implementar processos de implantações independentes com micro-frontend, possíveis desafios e como otimizar ainda mais a integração dessa abordagem em diferentes contextos de desenvolvimento de software.

Buscando eficiência com divisões de micro-frontends. Esta pesquisa investigou o entendimento dos participantes em relação às formas de dividir aplicações ao utilizar micro-frontend. O resultado desta investigação apontou que apenas 10% dos participantes souberam reconhecer de forma assertiva os formatos de divisões, conforme proposto por Lucca Mezzalira (Mezzalira, 2021). É importante ressaltar que essas abordagens de divisão são conceitos relativamente recentes e podem não ser amplamente conhecidos devido à sua principal publicação ter ocorrido em 2021 e não existir no atual momento desta pesquisa versões em português. Essa informação destaca um desafio relevante tanto para a indústria quanto para a pesquisa acadêmica.

Na indústria, a falta de familiaridade com essas abordagens pode limitar a adoção eficaz do micro-frontend, impedindo as organizações de colher os benefícios desses métodos de divisão. Portanto, as empresas que desejam adotar o micro-frontend podem enfrentar a necessidade de investir em treinamento e educação para disseminar o conhecimento sobre essas abordagens. Pesquisas futuras podem se concentrar em explorar a aplicação prática dessas divisões em cenários reais de desenvolvimento, avaliando os desafios e as vantagens específicas que surgem ao adotar essas estratégias.

Desafios e oportunidades ao explorar o uso do DDD em micro-frontends. Esta pesquisa investigou o entendimento dos participantes em relação à utilização do DDD

(Domain-Driven Design) no contexto de micro-frontend. O resultado desta investigação apontou que apenas 45% dos participantes tinham entendimento desta utilização. Esta informação destaca um desafio relevante para a adoção de práticas avançadas de arquitetura no desenvolvimento de aplicações com micro-frontend. Isto está alinhado com o resultado da pesquisa de Hulya Vural (Vural, H.; Koyuncu, M, 2021), que citou a dificuldade em encontrar estudos de caso que construam microsserviços baseados em DDD, e destacou também a necessidade de mais pesquisas para avaliar a aplicabilidade e eficácia do DDD em microsserviços.

Esses resultados indicam uma possível falta de conhecimento dos desenvolvedores sobre o DDD. Para a indústria, essa informação destaca a importância de oferecer treinamento e recursos educacionais que ajudem os profissionais a entender e aplicar eficazmente o DDD nas aplicações baseadas em micro-frontend. Além disso, para a pesquisa acadêmica, esse resultado sugere uma área de estudo promissora. Futuras pesquisas podem se concentrar em investigar os desafios e as melhores práticas relacionadas à implementação bem-sucedida do DDD em ambientes de micro-frontend, bem como em avaliar o impacto dessas práticas no desenvolvimento de software.

Desvendando o potencial da composição de micro-frontends via CDN. Esta pesquisa investigou o entendimento dos participantes em relação à composição de um micro-frontend via CDN (Rede de entrega de conteúdo). O resultado desta investigação apontou que apenas 40% dos participantes tinham entendimento desta utilização.

Esta informação destaca um desafio relevante para a adoção deste modelo de composição, tanto na indústria quanto na pesquisa acadêmica. Na indústria, a falta de familiaridade com a composição de micro-frontends via CDN pode limitar a adoção eficaz dessa estratégia, impedindo as organizações de aproveitar os benefícios de escalabilidade e desempenho que essa abordagem pode oferecer. Do ponto de vista acadêmico, esses resultados indicam a necessidade de uma pesquisa mais aprofundada sobre a composição de micro-frontends via CDN. Pesquisas futuras podem se concentrar em explorar as melhores práticas para a implementação bem-sucedida dessa estratégia, abordando possíveis desafios técnicos e avaliando o impacto dessa composição em cenários do mundo real.

6 CONCLUSÃO E TRABALHOS FUTUROS

O trabalho apresentou um survey exploratório que teve como objetivo obter o nível de compreensão dos desenvolvedores sobre o micro-frontend. A metodologia do projeto envolveu três fases: seleção dos participantes, aplicação de questionário e análise de dados. A pesquisa envolveu 22 participantes, predominantemente do sexo masculino, com várias faixas etárias e níveis de experiência.

Os resultados da pesquisa foram divididos em quatro seções: benefícios do uso de micro-frontends, implantação de micro-frontends, roteamento entre micro-frontends e comunicação entre micro-frontends. Em relação aos benefícios, os participantes reconheceram a geração de processos de implantações independentes e a governança descentralizada como vantagens claras, mas mostraram dúvidas em relação ao desacoplamento de serviços. A seção de implantação destacou que tecnologias tradicionais para o desenvolvimento de serviços web são amplamente conhecidas, mas abordagens mais recentes, como DDD e composição de micro-frontends via CDN, ainda são pouco compreendidas. Na seção de roteamento, os participantes tiveram um entendimento limitado de tecnologias usadas nesse contexto, como Web Components e Module Federation. Por fim, na seção de comunicação entre micro-frontends, o Event Bus e eventos customizados do JavaScript eram amplamente conhecidos, mas o mecanismo de comunicação chamado Event Bus foi pouco compreendido.

A pesquisa destaca a falta de conhecimento sobre alguns aspectos do micro-frontend, o que pode ter implicações significativas para a indústria e a pesquisa acadêmica. Isso sugere a necessidade de conscientização e educação mais amplas sobre os benefícios do micro-frontend, além de possíveis áreas de estudo adicionais. Por exemplo, o desacoplamento de serviços e a aplicação do DDD no contexto de micro-frontends são áreas que podem exigir mais atenção e pesquisa. Além disso, a composição de micro-frontends via CDN é uma estratégia que pode não ser amplamente compreendida, mas que pode ter um grande potencial.

Em resumo, o estudo fornece uma visão abrangente do entendimento dos desenvolvedores sobre micro-frontends e identifica áreas onde há oportunidades de melhorar o conhecimento e a adoção dessa abordagem de desenvolvimento de software. Isso pode beneficiar tanto a indústria quanto a pesquisa acadêmica, impulsionando a eficiência e a agilidade no desenvolvimento de aplicações web.

REFERÊNCIAS

- BARZOTTO, T. R.; FARIAS, K. **Evaluation of the impacts of decomposing a monolithic application into microservices: A case study**. p. 12, 2022.
- FARIAS, K. et al. **On the uml use in the brazilian industry: A state of the practice survey (s)**. In SEKE, 2018. p. 372–371.
- FERRACAKU, J. **The state of micro frontends: challenges of applying and adopting client-side microservices**. 2021. 57 f. Master's Thesis - Degree Programme in Computer Science and Engineering, University of Oulu, Oulu, 2021.
- FOWLER, M. 2019. **Micro frontends**. Disponível em: <https://martinfowler.com/articles/micro-frontends.html>. Acesso em: 11 Nov. 2023.
- FREITAS, H. et al. **O método de pesquisa survey**. RAUSP Management Journal, v. 35, n. 3, p. 105-112, 2000.
- HIDEKI, R. **Um guia para micro frontends: estudo aprofundado e aplicação de caso de uso no ecommerce**. 2021. 43 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2021.
- JÚNIOR, E.; FARIAS, K.; SILVA, B. **A survey on the use of uml in the brazilian industry**. In: Proceedings of the XXXV Brazilian Symposium on Software Engineering. [S.l.: s.n.], 2021. p. 275–284
- JÚNIOR, E. W.; FARIAS, K.; SILVA, B. da. **On the use of uml in the brazilian industry: A survey**. Journal of Software Engineering Research and Development, v. 10, p. 10–1, 2022.
- MEZZALIRA, L. **Building Micro-Frontends**, 1. ed. Sebastopol, CA, USA:O'Reilly Media, 2021. 334 p.
- MONTELIUS, A. **An exploratory study of micro frontends**. Master's thesis - Linköping University, Software and Systems, 2021.
- NEWMAN, S. **Migrando sistemas monolíticos para microsserviços**. 1. ed. São Paulo, SP:Novatec Editora Ltda, 2020. 288 p.
- PELTONEN, S.; MEZZALIRA, L.; TAIBI, D. **Motivations, benefits, and issues for adopting micro-frontends: a multivocal literature review**. Information and Software Technology, v. 136, p. 106571, 2021.
- VURAL, H.; KOYUNCU, M. **Does domain-driven design lead to finding the optimal modularity of a microservice?**. IEEE Access, v. 9, p. 32721-32733, 2021.
- WOHLIN, C. et al. **Experimentation in software engineering**. Berlin, Springer Science & Business Media, 2012. 233 p.

APÊNDICES

APÊNDICE A - Questões de pesquisa para análise dos participantes

Questão de pesquisa
1: Qual seu nome?
2: Qual sua idade?
3: Qual seu sexo?
4: Qual sua escolaridade?
5: Qual sua formação acadêmica?
6: Por quanto você estudou/tem estudado em universidades?
7: Quanto tempo de experiência você tem em desenvolvimento de software?
8: Quanto tempo de experiência você tem em desenvolvimento frontend?

APÊNDICE B - Questões de pesquisa investigadas neste artigo

Questão de pesquisa
1: A abordagem de micro-frontend tende a reduzir o tempo de espera para lançamentos de novos recursos
2: A abordagem de micro-frontend tende a acoplar o código de diversos serviços
3: A abordagem de micro-frontend gera processos de implantações independentes
4: A abordagem de micro-frontend cria uma governança descentralizada, permitindo a geração de equipes especialistas no domínio do negócio.
5: Esta ilustração trata-se de uma aplicação utilizando a abordagem de micro-frontend de divisão horizontal ?
6: Esta ilustração trata-se de uma aplicação utilizando a abordagem de micro-frontend de divisão vertical ?
7: A tecnologia Web components é utilizada para agrupar micro-frontends
8: A tecnologia Web components nos permite isolar um micro-frontend
9: Podemos criar um micro-frontend apenas usando tecnologias tradicionais, como VueJS, ReactJS, Nextjs ou Angular
10: Quando trabalhamos com micro-frontend de divisão vertical podemos utilizar DDD (Domain-Driven Design), pois podemos facilmente mapear o frontend e o backend dentro do mesmo contexto delimitado.
11: Podemos compor (montar) os micro-frontends, de três maneiras, do lado do cliente, do lado do servidor e pela camada de CDN (rede de entrega de conteúdo).
12: Para compor (montar) os micro-frontends do lado do cliente, precisamos de um shell/container responsável por carregar os micro-frontends.
13: Para compor (montar) os micro-frontends do lado do servidor, podemos usar o SSI (Server-Side Include)
14: Para compor (montar) os micro-frontends via CDN usamos uma linguagem de marcação baseada em

XML chamada ESI (Edge Side Includes).
15: Numa arquitetura de divisão vertical, a relação entre um micro-frontend e o shell da aplicação é sempre um para um, portanto, o shell da aplicação carrega apenas um micro frontend por vez.
16: Podemos usar uma combinação de iframes para carregar diferentes micro-frontends no lado do cliente.
17: Podemos lidar com roteamento de micro-frontends através de solicitações HTTP/HTTPS
18: A tecnologia Web Components é responsável pelo roteamento dos micro-frontends
19: A tecnologia SingleSPA é responsável pelo roteamento dos micro-frontends
20: A tecnologia Module Federation é responsável pelo roteamento dos micro-frontends
21: Cada micro-frontend deve estar inconsciente dos outros existentes na mesma página, caso contrário, estamos quebrando o princípio da Implantação Independente
22: Cada micro-frontend deve estar consciente dos outros existentes na mesma página, caso contrário, estamos quebrando o princípio da Implantação Dependente.
23: Para lidar com a comunicação entre micro-frontends podemos usar o Event Bus, um mecanismo que permite a comunicação entre componentes independentes por meio de eventos.
24: Para lidar com a comunicação entre micro-frontends podemos usar Eventos customizados do javascript.
25: Para lidar com a comunicação entre micro-frontends podemos usar a API de armazenamento na Web (localStorage, sessionStorage), independente do domínio ou subdomínio.
26: Para lidar com a comunicação entre micro-frontends podemos usar strings de consulta.