INTERFACE TECNOLÓGICA

ARQUITETURA DE MICROFRONTEND

MICROFRONTEND ARCHITECTURE

Marlon Vinicius Batista – marlon.batista@fatec.sp.gov.br
Faculdade de Tecnologia de Taquaritinga (Fatec) – Taquaritinga – SP – Brasil

Giuliano Scombatti Pinto – giuliano.pinto@fatec.sp.gov.br
Faculdade de Tecnologia de Taquaritinga (Fatec) – Taquaritinga – SP – Brasil

DOI: 10.31510/infa.v20i2.1803
Data de submissão: 06/09/2023
Data do aceite: 16/11/2023
Data da publicação: 20/12/2023

RESUMO

Este artigo propõe uma pesquisa bibliográfica com o intuito de abordar de forma abrangente os aspectos positivos e negativos quanto à implementação da arquitetura de microfrontend no cenário atual de desenvolvimento WEB. Além disso, busca explorar em detalhes o conceito dessa abordagem inovadora no contexto do desenvolvimento de aplicações web. A etapa de planejamento para o desenvolvimento de software é fundamental para o sucesso do projeto frente a desafios futuros. Desta forma, a escolha da arquitetura que será utilizada é a principal etapa dentro do planejamento do software, visto que todo o resto será adequado a suas normas e filosofias. Tradicionalmente, o desenvolvimento front-end tem se baseado em monolitos, onde a interface do usuário é construída como um todo indivisível. No entanto, a crescente complexidade das aplicações web modernas tem levado a desafios de escalabilidade, manutenção e colaboração. O microfrontend surge como uma solução para esses problemas, permitindo a divisão de interfaces em módulos independentes, cada um com seu próprio ciclo de vida, tecnologias e times de desenvolvimento. Neste artigo, os princípios do microfrontend foram examinadas, bem como suas vantagens e desvantagens, implementação prática e casos de uso, além de fornecer uma análise crítica das contribuições de autores renomados nesse campo emergente. Os resultados apontaram que apesar dos inúmeros benefícios oferecidos pela arquitetura, é necessário ponderar as questões e desafios que ainda não foram solucionados com a implementação da arquitetura de microfrontends.

Palavras-chave: Microfrontend. Arquitetura Web. *Frontend*. Desenvolvimento Web.

ABSTRACT

This article proposes a bibliographic research aiming to comprehensively address both the positive aspects and inherent challenges of implementing the microfrontend architecture. Furthermore, it seeks to delve into the concept of this innovative approach within the context of web application development. The planning stage for software development is crucial for project success in the face of future challenges. Thus, the selection of the architecture to be used stands as the key phase within software planning, as everything else will align with its standards

INTERFACE TECNOLÓGICA

and philosophies. Traditionally, front-end development has relied on monoliths, where the user interface is built as an indivisible whole. However, the increasing complexity of modern web applications has led to scalability, maintenance, and collaboration challenges. Microfrontends emerge as a solution to these issues, enabling the division of interfaces into independent modules, each with its own lifecycle, technologies, and development teams. In this article, we examine the principles of microfrontends, their advantages and disadvantages, practical implementation, use cases, and provide a critical analysis of contributions from renowned authors in this emerging field. The results showed that despite the numerous benefits offered by the architecture, it is necessary to consider the issues and challenges that have not yet been resolved with the implementation of the microfrontends architecture.

Keywords: Microfrontend. Web Architecture. Frontend. Web development.

1 INTRODUÇÃO

As técnicas utilizadas no desenvolvimento de softwares têm evoluído significativamente nas últimas décadas, acompanhando as demandas crescentes por experiências ricas e escaláveis aos usuários. Segundo Jackson (2019) a arquitetura de microserviços tem se tornando bem popular em meio ao desenvolvimento de software e inúmeras organizações têm adotado essa arquitetura para resolver problemas em aplicações monolíticas no back-end. Jackson (2019) ainda cita que apesar dos avanços feitos nas aplicações do lado do servidor (back-ends), o front-end continua a enfrentar esses dilemas, mantendo a maioria das aplicações como monolíticas, concentrando o código em uma única aplicação.

Segundo Lira (2023), à medida que as aplicações web se tornam cada vez mais complexas, a necessidade de adotar abordagens arquiteturais avançadas se torna premente. Uma dessas abordagens emergentes, que ganhou destaque nos últimos anos, é conhecida como microfrontends.

As microfrontends são uma extensão da arquitetura de microserviços para o front-end, permitindo que equipes de desenvolvimento dividam a interface do usuário em componentes menores, independentes e autônomos. Cada um desses componentes, ou microfrontends, é desenvolvido, testado e implantado separadamente, oferecendo uma série de benefícios que vão desde a escalabilidade até a manutenibilidade. Diferente do que é aplicado ou utilizado por boa parte das aplicações web atuais, que são constituídas geralmente em uma aplicação monolítica, concentrando todo seu código fonte em um único lugar.

A proposta imposta pela arquitetura de micro-frontend sugere a possibilidade de utilizar diferentes frameworks e bibliotecas sem que elas estejam necessariamente na mesma raiz, apresentando algo completamente inovador para o ramo de programação web. Segundo o autor

Jackson (2019) planejar e orquestrar o desenvolvimento de uma aplicação web com qualidade pode ser um desafio, especialmente quando equipes distintas precisam trabalhar simultaneamente na mesma aplicação.

O objetivo do presente artigo é explorar a fundo o conceito de microfrontends, suas vantagens, desafios e tecnologias relacionadas. Além disso, examinar as estratégias de implementação, os desafios comuns enfrentados pelos desenvolvedores e as tendências futuras nesse campo em constante evolução.

2 ORIGEM DO MICROFRONTEND E SEUS BENEFÍCIOS

O cenário do desenvolvimento de software apresenta constante evoluções todos os dias, visando sempre soluções que possibilitem a criação de aplicações mais escaláveis, flexíveis e eficientes. Nesse contexto, a proposta apresentada pela arquitetura do microfrontend, bem como apresentada por Adam Bellemare no livro *"Building Event-Driven Microservices"*, tem se destacado como uma abordagem promissora para enfrentar os desafios proposto pelo desenvolvimento web tradicional.

A abordagem tradicional de desenvolvimento web muitas vezes se apoia em monólitos, onde a interface do usuário é construída como um todo coeso. No entanto, a crescente complexidade das aplicações web modernas trouxe à tona problemas de escalabilidade, manutenção e colaboração. É aqui que o microfrontend entra em jogo como uma solução inovadora, através do modelo single page app.

Segundo Geers (2023, online) a nomenclatura de microfrontend foi relatada pela primeira vez no final ano de 2016, no *ThoughtWorks Tecnhology Radar*, que é considerado uma ferramenta valiosa para profissionais de tecnologia, líderes de TI e empresas que desejam manter-se atualizados sobre as tendências tecnológicas e as melhores práticas da indústria. Ele fornece orientação e insights que podem ajudar na tomada de decisões informadas sobre tecnologia em projetos e estratégias de negócios.

Geers (2023, online) menciona em seu site que a ideia dessa nova arquitetura consistia em trazer os mesmos conceitos do micro serviços para o front end, possibilitando a construção de uma aplicação rica em recursos e extremamente poderosa. Para o autor as aplicações web (monolíticas), muitas vezes desenvolvidas por equipes distintas tendem a aumentar a complexidade de sua manutenção, diante do crescimento da aplicação.

Segundo Adam Bellemare (2020) há vários pontos do conceito de microfrontend que se alinham com a arquitetura de microservices. Enquanto os microservices se concentram na divisão da lógica de negócios em serviços independentes, o microfrontend estende esse conceito para a interface do usuário. Isso permite a criação de módulos independentes, cada um com seu próprio ciclo de vida, tecnologias e equipes de desenvolvimento.

2.1 Dilemas e desafios que o paradigma de desenvolvimento microfrontend se propõe a abordar e resolver

O conceito de microfrontend é uma abordagem arquitetural para desenvolvimento de front-end que visa solucionar vários problemas comuns encontrados em aplicações web complexas. De acordo com Geers (2020), a aplicação dos conceitos de microfrontend é associada a diversos efeitos benéficos, impactando positivamente tanto a organização quanto a aplicação na qual são implementados.

À medida que uma aplicação web cresce, torna-se mais difícil gerenciá-la e manter sua velocidade de desenvolvimento. O microfrontend permite que equipes de desenvolvimento trabalhem de forma independente em partes da aplicação, facilitando o dimensionamento.

Para Jackson (2019) a possiblidade de entregas desacopladas e atualizações incrementais é fundamental para a entrega de valor e a continuidade segura do desenvolvimento da aplicação. O autor também ressalta a liberdade para tomada de decisão caso a caso, pensando nas partes individuais e realizando atualizações ou melhorias incrementas através da arquitetura, depenicas e experiencia do usuário final.

O autor Geers (2020) destaca outros benefícios como:

- Risco de falha reduzido e indisponibilidade da aplicação, componentes isolados.
- Compreensão do código fonte com mais facilidade.
- Código fonte menor, permitindo o processo de refatoração quando necessário.
- Comportamento previsível.

A pesquisa conduzida por Pavlenko et al (2020) identifica diversos benefícios adicionais. Um deles é a capacidade de resiliência em situações em que a nova demanda possa falhar após sua implementação, mantendo-a isolada das outras partes do código. Além disso, os autores ressaltam a importância da liberdade de escolha concedida à equipe de desenvolvimento em relação à tecnologia a ser utilizada, sem afetar as demais partes do sistema.

INTERFACE TECNOLÓGICA

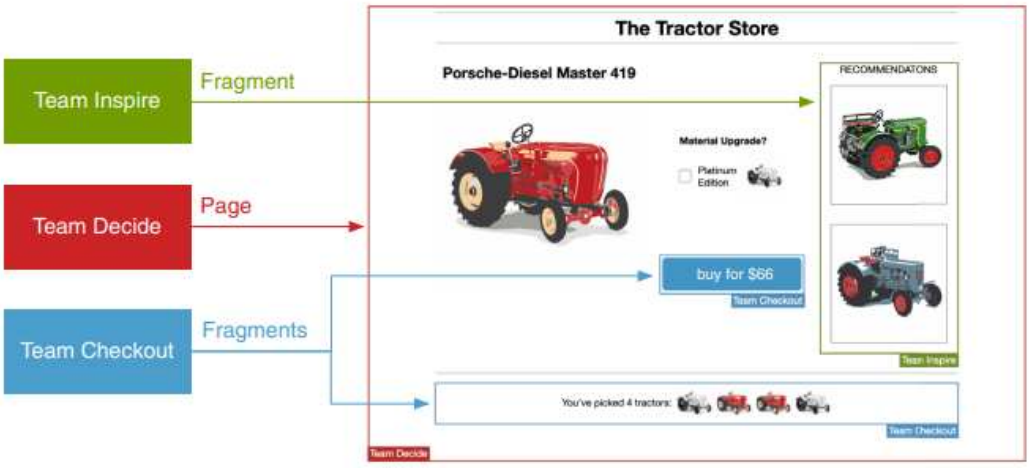
No entanto, a pesquisa suscita questões futuras que requerem soluções, tais como a orquestração, a gestão das rotas, a resolução de colisões no ambiente, a comunicação, a manutenção da consistência no design da aplicação e o controle de dependências (no caso de carregar uma biblioteca mais de uma vez).

É importante observar que o uso do microfrontend também traz desafios, conforme mencionado anteriormente, e nem sempre é a solução certa para todos os projetos. A escolha de adotar o microfrontend deve ser baseada nas necessidades específicas do projeto e na compreensão dos prós e contras dessa abordagem.

2.2 Fragmentação

De acordo com Geers (2020), o processo de fragmentação de uma aplicação desempenha um papel fundamental na distribuição de responsabilidades e regras entre os diferentes componentes, vinculando cada um a uma equipe específica. Isso possibilita que a equipe responsável se concentre com maior precisão no comportamento esperado e nas regras de negócio exclusivas desse componente em particular. A Figura 1 ilustra um exemplo de fragmentação aplicada a uma página de venda de tratores, na qual estão presentes diversos fragmentos, sendo que a cada um deles está associada uma equipe responsável específica.

Figura 1 – Exemplo de fragmentação dos componentes e divisão de responsabilidade por time.



Fonte: Geers (2020)

Embora as equipes de Inspiração (*Team Inspire*) e Verificação (*Team Checkout*) tenham a responsabilidade exclusiva por seus respectivos fragmentos, cabe ao time de decisão (*Team Decide*) a tarefa de orquestrar a disposição desses componentes na tela de forma a apresentá-los ao cliente.

2.3 Single-SPA Framework

O *framework* (estrutura) Single-SPA, segundo sua documentação tem por objetivo orquestrar e reunir inúmeros microfrontends Javascript em um único aplicativo ou página web (SINGLE-SPA, 2023). A implementação dessa estrutura agrega inúmeros benefícios, bem como:

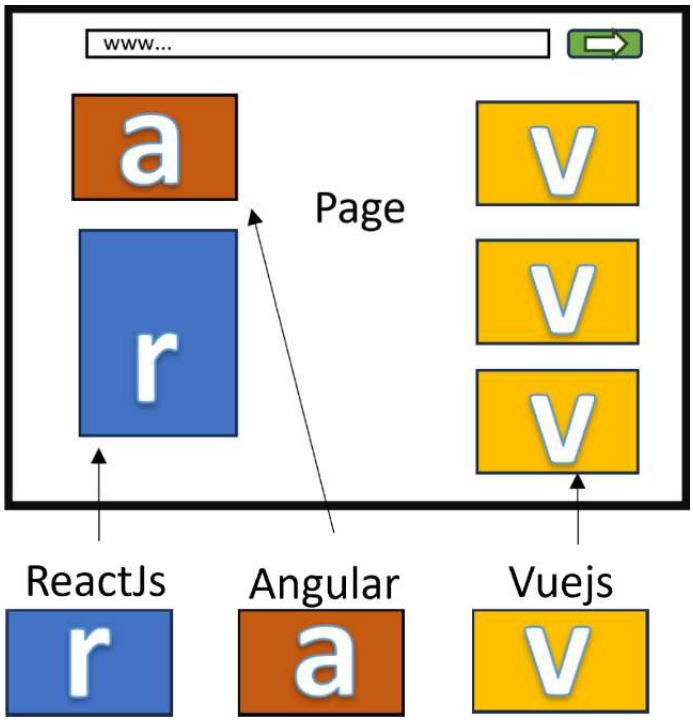
- Utilização de distintos frameworks e bibliotecas na mesma página sem que seja necessário o *refresh* (atualização).
- Implementação independente dos componentes (microfrontends)
- Escrita do código fonte utilizando um nova ferramenta ou biblioteca sem que sobrepor o código atual.
- Carregamento preguiçoso, carregando apenas o que for solicitado.

A documentação do Single-SPA indica que o *framework* foi inspirado nos frameworks modernos e seus ciclos de vida para cada componente. A ideia por traz dele surgiu através do desejo de utilizar diferentes frameworks em uma mesma aplicação web.

A utilização do Single-SPA consiste na renderização de uma página HTML onde será apresentado a parte visual ao usuário e outra página Javascript onde serão orquestrados os componentes, tendo cada um deles três atributos, um nome e duas funções. A primeira será responsável por gerenciar o estado do componente (aplicativo), caso ele esteja ativo ou não. A segunda função possui a responsabilidade de carregar o código do componente. A Figura 2 apresenta um exemplo visual da utilização de microfrontends com o *framework* Single-SPA.

INTERFACE TECNOLÓGICA

Figura 2 - Exemplo de utilização de Single-SPA com microfrontends



Fonte: Elaborada pelo autor

O *framework* Single-SPA faz a orquestração de diferentes componentes, sendo cada um deles desenvolvimento por um framework (Angular, Vuejs) ou biblioteca (ReactJs) diferentes, todos gerenciados e apresentados na mesma single page ao usuário final, sem a necessidade de realizar o recarregamento do DOM *(Document Object Model)*.

2.4 Estilização

Conforme a documentação do MDN Web Docs (2023), o CSS (*Cascading Style Sheets*) ou Folha de Estilo em Cascata é uma linguagem de estilização que capacita os desenvolvedores a personalizar o estilo e a apresentação dos elementos HTML em páginas web. No entanto, ao implementar a arquitetura de microfrontends em projetos, como observado por Jackon (2019), surgem considerações importantes a serem feitas. Isso ocorre porque, mesmo que as aplicações estejam em repositórios diferentes e tenham sido desenvolvidas em momentos distintos, existe o potencial de conflitos e sobreposições de estilos.

O autor ilustra essa questão com um exemplo concreto. Suponha que um microfrontend contenha uma *tag* <h2> com a propriedade *color* definida como preta, enquanto outro

componente gerenciado pela mesma página tenha a mesma *tag* e a mesma propriedade, mas com a cor definida como azul. Nesse caso, a estilização será sobreposta. É importante notar que esse não é um problema novo, mas se torna mais complexo quando esses componentes são desenvolvidos por equipes separadas e possuem códigos independentes.

Jackson (2019) enfatiza que, embora esse desafio não seja intransponível, ele pode aumentar a complexidade da resolução de conflitos de estilo, pois as raízes dos problemas podem estar dispersas em diferentes partes dos microfrontends. Portanto, ao adotar a arquitetura de microfrontends, é crucial que as equipes de desenvolvimento estejam cientes desse cenário e implementem estratégias de gerenciamento de estilo eficazes, como convenções de nomenclatura, encapsulamento de estilos e comunicação colaborativa entre as equipes, para garantir a consistência e evitar conflitos indesejados no design e na apresentação das páginas web.

3 PROCEDIMENTOS METODOLÓGICOS

A metodologia utilizada no trabalho foi a de pesquisa e revisão bibliográfica. Essa metodologia foi escolhida com o objetivo de revisar e comparar os trabalhos já apresentados à sociedade referentes ao tema proposto. Os materiais utilizados foram os livros, artigos, documentações e páginas *web.* Inicialmente foi realizado um levantamento dos trabalhos (artigos e monografias) nacionais e internacionais que tratavam do tema de arquitetura de microfrontends. Posteriormente, ocorreu a pesquisa por livros e páginas web que explorassem os aspectos da arquitetura bem como a sua implementação. Por fim, fora realizado a comparação desses materiais e a elaboração do artigo.

4 RESULTADOS E DISCUSSÃO

Considerando todos os pontos apresentados até o momento, fica evidente que a arquitetura de microfrontends oferece uma série de benefícios notáveis. Um aspecto que merece destaque é a liberdade de escolha ao trabalhar com diferentes frameworks em uma mesma página web. No entanto, como Geers (2020) aponta que a liberdade na escolha de tecnologias nem sempre é a abordagem mais adequada para todos os times e projetos. O uso de frameworks semelhantes facilita a implementação de boas práticas no desenvolvimento de aplicações.

INTERFACE TECNOLÓGICA

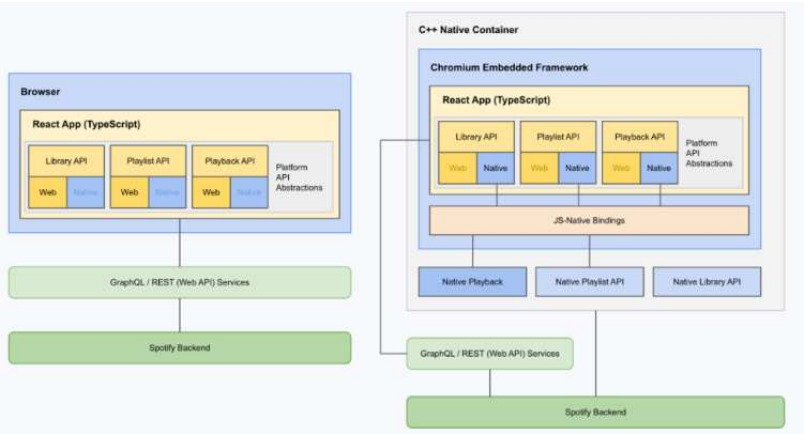
Além disso, essa abordagem traz economias significativas desde o início do projeto, incluindo definições para estruturas iniciais, tratamento de erros, controle de formulários e configurações básicas. Esse processo culmina na criação de um modelo que outras equipes podem clonar e utilizar como ponto de partida para seus próprios desenvolvimentos. Isso não apenas aumenta a produtividade na construção de aplicações subsequentes, mas também agiliza a entrega de valor aos usuários.

Portanto, ao adotar uma abordagem de frameworks semelhantes na arquitetura de microfrontends, as organizações podem colher os benefícios da padronização e da eficiência, permitindo que equipes construam e iterem sobre aplicações de forma mais rápida e consistente, ao mesmo tempo em que mantêm a flexibilidade necessária para atender às necessidades específicas de cada projeto.

4.1 Aplicações que implementam o microfrontend

Embora a arquitetura de microfrontends esteja ganhando notoriedade ela ainda não é tão difundida quanto outras abordagens arquiteturais. No entanto, destacam-se dois exemplos notáveis de aplicações que incorporam os princípios da arquitetura de microfrontends: o Spotify, um aplicativo de reprodução de músicas, e a Netflix, uma renomada plataforma de streaming. Segundo o portal Spotify.R&D Engineering (2021) os conceitos do microfrontends foram aplicados durante a evolução de suas aplicações (*Desktop* e *WEB*). A Figura 3 ilustra a arquitetura utilizada nas aplicações do Spotify, sendo o lado esquerdo a representação do aplicativo web e o lado direto a aplicação *desktop*.

Figura 3 - Arquitetura do Spotify



Fonte: Spotify.R&D Engineering (2021)

5 CONCLUSÃO

Através da pesquisa realizada é possível afirmar que a arquitetura de microfrontend emergiu como uma abordagem inovadora e poderosa para o desenvolvimento de aplicações web na era moderna. Ao dividir a interface do usuário em módulos independentes e autônomos, os microfrontends oferecem uma série de benefícios notáveis, ao mesmo tempo em que apresentam desafios a serem enfrentados.

Entre os benefícios mais evidentes destaca-se a capacidade de desenvolver em paralelo, acelerando o tempo de desenvolvimento e permitindo que equipes trabalhem de forma mais eficaz. A escalabilidade é uma realidade, pois partes específicas de uma aplicação podem ser dimensionadas conforme necessário, tornando a gestão do tráfego e dos recursos mais eficiente. A reutilização de componentes é fomentada, promovendo a consistência e economizando tempo. A facilidade de manutenção é aprimorada, uma vez que cada microfrontend é uma entidade independente, minimizando os riscos de regressão.

Entretanto, esses benefícios vêm acompanhados de desafios. O gerenciamento complexo de vários microfrontends requer sistemas sólidos de controle de versão e roteamento dinâmico. A comunicação entre microfrontends deve ser cuidadosamente planejada para evitar conflitos e garantir a coesão da experiência do usuário. O aumento do número de requisições HTTP pode afetar o desempenho da aplicação se não for bem otimizado. Além disso, a compatibilidade entre navegadores pode ser um obstáculo, exigindo estratégias adicionais.

Em conclusão, a arquitetura de microfrontend representa um avanço significativo no desenvolvimento de aplicações web, permitindo uma abordagem modular e escalável. No entanto, para colher todos os seus benefícios, as equipes de desenvolvimento devem abordar os desafios com cuidado, adotando melhores práticas, padrões e estratégias eficazes. Com a devida atenção aos detalhes e uma compreensão sólida de suas implicações, os microfrontends podem se tornar uma ferramenta poderosa para a criação de aplicações web modernas, ágeis e de alto desempenho.

REFERÊNCIAS

ANDREY PAVLENKO, NURSULTAN ASKARBEKULY, SWATI MEGHA1, AND MANUEL MAZZARA. Micro-frontends: application of microservices to web front-ends. Journal of Internet Services and Information Security (JISIS), volume: 10, number: 2 (May

2020), pp. 49-66. Disponivel em: <https://jisis.org/wp-content/uploads/2022/11/jisis-2020-vol10-no2-04.pdf>. Acesso em: 10 de setembro de 2023.

BELLEMARE, Adam. Building Event-Driven Microservices. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol. 2020

GEERS, Michael. Micro Frontends in Action. Manning Publications Co. Shelter Island, 2020.

GEERS, Michal. Micro frontends: Estendendo a ideia de microservices para o desenvolvimento frontend. Disponível em: <https://tautorn.github.io/micro-frontends/>. Acessado em: 04 set. 2023

JACKSON, Cam. Micro Frontends. Martin Fowler. 19 jun 2019. Disponível em: <https://martinfowler.com/articles/micro-frontends.html>. Acessado em: 04 set. 2023

LIRA, Kevin. O poder dos Micro-Frontends: Uma abordagem moderna para a Arquitetura Frontend. AWS Editorial Team. 22 jun. 2023. Disponível em: < https://aws.amazon.com/pt/blogs/aws-brasil/o-poder-dos-micro-frontends-uma-abordagem-moderna-para-a-arquitetura-frontend/>. Acessado em: 15 set. 2023.

MDN contributors. MDN WEB DOCS. CSS. 6 de nov. de 2022. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/CSS>. Acessado em: 12 de set. 2023

Michael Possumato, Nick Tomlin, Jordan Andree, Andrew Shim, and Rahul Pilani. Netflix Technology Blog. How We Build Micro Frontends With Lattice. 27 set. 2021. Disponível em: <https://netflixtechblog.com/how-we-build-micro-frontends-with-lattice-22b8635f77ea>. Acesso em: 21 set. 2023

SINGLE-SPA(2023). Single-SPA: Getting Started with single-spa, Docs, 2023. Disponível em:<https://single-spa.js.org/docs/getting-started-overview>. Acesso em: 12 de set. de 2023

Spotify.R&D Engineering. Building the Future of Our Desktop Apps. 7 abr. 2021. Disponível em:<https://engineering.atspotify.com/2021/04/building-the-future-of-our-desktop-apps/>. Acesso em: 22 de set. 2023