

# Analizando o Desempenho de Angular, React e Vue em Projetos Web: Um Estudo Comparativo

Júlia Llorente<sup>1</sup>, Nicole Carolina Mendes<sup>1</sup>

<sup>1</sup>Departamento de Ciência da Computação – Universidade do Estado de Santa Catarina (UDESC)  
89219-710 – Joinville – SC – Brazil

**Abstract.** *The article provides a comprehensive analysis of the performance of three popular front-end frameworks: Angular, React, and Vue. Using rigorous benchmarks and runtime metrics, it reveals that Angular stands out as a leader in efficiency in most tasks, thanks to its optimized architecture. However, it is observed that in specific scenarios, such as handling tables on web pages, Angular may face challenges, with longer runtimes compared to React and Vue. These findings underscore the need for careful analysis when choosing a framework, taking into account the context and performance requirements of a project.*

**Resumo.** *O artigo apresenta uma análise abrangente do desempenho de três populares frameworks front-end: Angular, React e Vue. Utilizando benchmarks rigorosos e métricas de tempo de execução, revela que o Angular se destaca como líder em eficiência na maioria das tarefas, graças à sua arquitetura otimizada. No entanto, observa-se que em cenários específicos, como a manipulação de tabelas em páginas da web, o Angular pode enfrentar desafios, com tempos de execução mais longos em comparação com React e Vue. Essas descobertas destacam a necessidade de uma análise cuidadosa ao escolher um framework, considerando o contexto e os requisitos de desempenho de um projeto.*

## 1. Introdução

Este artigo visa realizar uma análise comparativa detalhada entre os frameworks JavaScript Vue.js, Angular e React. Este estudo é particularmente relevante dado o cenário tecnológico atual, onde a escolha do framework certo pode impactar significativamente a eficiência do desenvolvimento e o desempenho das aplicações web. Utilizaremos um conjunto de métricas específicas e dados empíricos para conduzir nossa análise.

Atualmente, os navegadores funcionam como a principal interface entre usuários e a Internet, facilitando o acesso a documentos e recursos disponíveis na World Wide Web (WWW). Tais recursos são geralmente apresentados como sites ou aplicações web, desenvolvidos com tecnologias interpretáveis pelos navegadores, como HTML, CSS e JavaScript.

O HyperText Markup Language (HTML) serve para estruturar o conteúdo de aplicações web, utilizando elementos de marcação para representar texto, imagens, links e outros componentes visuais em navegadores. Além disso, é responsável pelo gerenciamento de dependências da aplicação. Por outro lado, Cascading Style Sheets (CSS) determina o estilo visual dos elementos anotados em HTML, especificando seu layout, cores e posicionamento. JavaScript (JS) atua como uma linguagem de programação que permite a criação e manipulação dinâmica do conteúdo da aplicação [Ferreira Jr et al. 2021].

Enquanto HTML e CSS são tecnologias voltadas principalmente para a criação de conteúdo estático e apresentação visual, JavaScript oferece uma gama mais extensa de funcionalidades. Isso se reflete na ampla variedade de ferramentas, bibliotecas e frameworks desenvolvidos especificamente para otimizar e expandir suas capacidades.

Conforme indicado pela pesquisa realizada pelo Stack Overflow em 2021, o JavaScript é a linguagem de programação mais amplamente utilizada globalmente, com uma adoção de 64,96%. Quando se trata de frameworks e bibliotecas web, React.js lidera o ranking, sendo utilizado por 40,14% dos desenvolvedores. Angular e Vue.js ocupam, respectivamente, o quarto e quinto lugares, com taxas de adoção de 22,96% e 18,97% [Overflow 2021].

O crescimento do JavaScript pode ser atribuído a um ecossistema robusto que não só abriga uma ampla gama de tecnologias, mas também é impulsionado por uma energia inovadora [Elliott 2017], conforme traduzido por [Anjos 2017]. Este dinamismo é ainda mais evidenciado pela rica variedade de frameworks disponíveis, como documentado por [Francisco 2017].

É crucial enfatizar que não há uma resposta definitiva sobre qual framework é objetivamente superior aos outros. Em vez disso, a eficácia de um framework deve ser avaliada com base em cenários específicos frequentemente encontrados no mercado. Para orientar essa avaliação, a seção sobre critérios de avaliação fornecerá parâmetros que nos ajudarão a abordar essas questões.

## **2. Revisão de Literatura**

Um framework é uma estrutura predefinida que oferece bibliotecas e componentes para agilizar o desenvolvimento de aplicações web, melhorando sua produtividade, flexibilidade, manutenibilidade e testabilidade [Brans 2014]. Por outro lado, uma biblioteca é um conjunto de código reutilizável que auxilia o desenvolvimento de um projeto existente, fornecendo classes, funções e scripts [Ferreira Jr et al. 2021]. Embora os termos "framework" e "biblioteca" sejam frequentemente usados de forma intercambiável, neste estudo, consideramos o React como um framework, em sintonia com sua utilização prática mais comum.

### **2.1. Vue**

Evan You, o criador do Vue.js (<http://evanyou.me/>), trabalhava no Google Creative Labs, e ele sentia a necessidade de prototipar o mais rápido possível em vez de se preocupar com uma grande interface visual. A consequência era que ele repetia muito código HTML, o que fazia com que ele se sentisse improdutivo, pois consumia muito de seu tempo. [Incau 2017]

Para o tipo de projeto que ele trabalhava, as prioridades eram algo flexível e leve. Quando ele viu que isto ainda não existia no mercado, e por ser um ótimo programador como é, decidiu então ele mesmo resolver este problema, criando o Vue.js. Apesar de o Vue.js ter nascido como uma ferramenta de prototipação rápida, agora ele já pode ser usado para criação de grandes e escaláveis aplicações reativas. [Incau 2017]

Vue (pronuncia-se / vju:/, como view) é uma estrutura progressiva para construção de interfaces de usuário. Ao contrário de outras estruturas monolíticas, o Vue foi projetado

desde o início para ser adotável de forma incremental. A biblioteca principal concentra-se apenas na camada de visualização e é fácil de escolher e integrar com outras bibliotecas ou projetos existentes. Por outro lado, o Vue também é perfeitamente capaz de alimentar aplicativos sofisticados de página única quando usado em combinação com ferramentas modernas e bibliotecas de suporte. [Team 2022]

Uma característica do Javascript, e por consequência também do Vue.js, é a possibilidade de ser utilizado em qualquer projeto que possua front-end. Independente da linguagem de programação escolhida, é possível usar o Vue.js. [Incau 2017]

O Vue se diferencia por ser uma biblioteca não intrusiva, ou seja, ele não tem um código que o force a seguir o padrão dele, como é o caso do Angular 2 com TypeScript. A maioria do código para Vue é escrito em JavaScript puro. Ele possui uma sintaxe muito clara e limpa. Outro grande diferencial é sua flexibilidade. Veja que nos referenciamos ao Vue como uma biblioteca (lib) e não um framework, pois ele trabalha com outras bibliotecas que, juntas, formam um framework. [Incau 2017]

Os componentes desse framework funcionam em apenas um arquivo, com a extensão .vue, encapsulando a lógica em JS, a estilização e o HTML em seu conteúdo, porém caso o usuário prefira há também a possibilidade de fazer os componentes em JS puramente. Algumas das vantagens de usar a arquitetura que engloba tudo em um arquivo citadas na documentação são: templates pré-compilados e otimização de tempo de compilação, influenciando na melhora da performance. [Pinto et al. 2023]

## **2.2. React**

ReactJS começou através do XHP, uma versão do PHP que o Facebook lançou em 2010. XHP foi principalmente desenvolvido para minimizar os ataques de Cross Site Scripting (XSS), que é uma vulnerabilidade do sistema quando uma pessoa tenta inserir códigos nas páginas web. Mas o XHP não resolveu uma demanda: aplicações web dinâmicas geralmente precisam de muitas requisições, e XHP não conseguia lidar com essa parte. [Morais 2021]

Assim, um engenheiro do Facebook (Jordan Walke) negociou com o seu gestor para levar XHP para o browser usando JavaScript e entraram em acordo para experimentar por 6 meses. O resultado desse experimento é o nascimento do ReactJS. [Morais 2021]

React JS é uma biblioteca JavaScript para a criação de interfaces de usuário — ou UI (user interface). Criado em 2011 pelo time do Facebook, o React surgiu com o objetivo de otimizar a atualização e a sincronização de atividades simultâneas no feed de notícias da rede social, entre eles chat, status, listagem de contatos e outros. A princípio, todas essas atividades, chamadas de estados, tinham uma descrição muito complexa. Com o React, esta descrição torna-se mais simples, bem como também é simplificada a conexão entre HTML, CSS e JavaScript e todos os componentes de uma página. [Roveda 2023]

O React busca facilitar a conexão entre diferentes partes de uma página, portanto seu funcionamento acontece através do que chamamos de componentes. Em outras palavras, podemos imaginar que o React divide uma tela em diversos componentes para, então, trabalhar sobre eles de maneira individual. Os componentes são utilizados para reaproveitamento de código e padronização de interface. Isso torna o React uma tecnologia muito flexível para a solução de problemas e para a construção de interfaces

reutilizáveis, uma vez que cada um destes componentes pode ser manipulado de maneira distinta. [Roveda 2023]

De acordo com [Pinto et al. 2023], por se basear no paradigma declarativo, o React consegue permitir ao usuário dizer o que quer que aconteça, e não necessariamente citando o passo a passo. Isso permite ao código ficar mais fácil de entender, e mais limpo. Esse conceito é contrário ao paradigma imperativo, em que há a necessidade de ditar o passo a passo do código.

[Pinto et al. 2023] Também discorre que, o segundo pilar do React é ele ser baseado em componentes, logo é possível reutilizar elementos passando parâmetros de diferentes tipos de dados, tendo funções de maneira encapsulada, e cuidando dos estados de mudança de forma individual. Isso aproveita o conceito de reusabilidade de código, permitindo que não precise haver trechos de códigos repetidos no programa.

O terceiro pilar do React é o lema “Aprenda uma vez, use em qualquer lugar”, isso se dá pelo fato do React trabalhar com outras tecnologias, não precisando descartar o seu código já existente. [Pinto et al. 2023]

### 2.3. Angular

AngularJS - também chamado de Angular 1 após o lançamento da segunda versão - foi criado em 2009, época da segunda geração de *frameworks*, pelos sócios Miško Hevery e Adam Abrons, na Brat Tech LLC. [Anjos 2017]

Hevery em seguida começou a trabalhar em um projeto no Google chamado Google Feedback. Em 6 meses, com uma equipe de 3 desenvolvedores, escreveram mais de 17 mil linhas de código, e quanto mais o projeto crescia, mais difícil se tornava testar e fazer alterações. Foi por causa dessa frustração que Miško resolveu fazer uma aposta com seu supervisor, em que ele disse que seria possível reescrever a aplicação inteira em 2 semanas usando seu *framework*, chamado na época, *GetAngular*. Miško perdeu a aposta, mas conseguiu realizar o feito em não duas, porém, três semanas e reduziu de 17 mil linhas para aproximadamente 1500 linhas, o que foi considerado um grande sucesso, e levou seu supervisor a acelerar o desenvolvimento do *framework* dentro do Google. [Anjos 2017]

O AngularJS do Google é um framework JavaScript MVC (Model-View-Controller, ou Modelo-Visão-Controlador) completo, que facilita bastante a criação rápida de aplicações que executem adequadamente em qualquer plataforma desktop ou móvel. Em um período de tempo muito curto, o AngularJS passou de uma opção de código aberto desconhecida para um dos frameworks JavaScript do lado cliente mais conhecido e mais amplamente utilizado. O AngularJS 1.3 e as versões mais recentes, combinados com a jQuery e o Twitter Bootstrap, oferecem tudo que é necessário para implementar rapidamente frontends HTML5 de aplicações JavaScript que usem web services REST para os processos em backend. [Williamson 2015]

Angular é construído inteiramente sobre o TypeScript, e usá-lo no desenvolvimento é recomendado, embora não seja obrigatório. O TypeScript é um superconjunto de JavaScript, que adiciona tipificação estática ao código. Uma das mais importantes características e vantagens desta plataforma é o uso de dependency injection como forma de adotar a inversão de controle, permitindo interligar partes de código e módulos, mantendo o código com baixo acoplamento. [da Cruz Scotti et al. 2019]

Angular traz em sua caixa de ferramentas alguns serviços, tais como o *\$http* para realizar requisições AJAX, o *\$timeout* para definir atrasos programáticos na execução de algum código, e também permite que se crie serviços customizados para a necessidade de cada aplicação. Para que o desenvolvedor possa alterar ou formatar os dados que vem na resposta da requisição, Angular possibilita a criação de filtros e *interceptors*. [Anjos 2017]

O framework, segundo seu site, pode ser utilizado para desenvolver aplicações web e native, de modo a contemplar de smartphones a computadores desktops. Outro ponto é a escalabilidade de dados, assegurando a possibilidade de crescer suas aplicações. [Pinto et al. 2023]

### 3. Metodologia

Para a coleta de dados sobre cada framework, foi empregada uma abordagem multimétodo que incluiu a consulta de literatura acadêmica, fóruns online, bem como a documentação oficial fornecida pelos respectivos desenvolvedores. Os dados coletados foram então meticulosamente analisados e documentados, servindo como fundamento para a avaliação subsequente.

### 4. Critérios de Avaliação

Ao avaliar tecnologias e frameworks para uso em projetos web, é essencial considerar cuidadosamente dois critérios fundamentais: popularidade e desempenho. A popularidade reflete a aceitação e adoção pela comunidade de desenvolvedores, o que pode afetar o suporte, a disponibilidade de recursos e a facilidade de encontrar desenvolvedores qualificados. Por outro lado, o desempenho é crucial para garantir que a aplicação atenda às expectativas de resposta e eficiência. A combinação desses dois critérios permite uma escolha informada, equilibrando a relevância e a eficácia da tecnologia escolhida em um cenário competitivo de desenvolvimento web.

#### 4.1. Popularidade

De acordo com a pesquisa realizada pelo Stack Overflow em 2021, o ecossistema de desenvolvimento web é fortemente dominado pela linguagem JavaScript, com uma taxa de adoção global de quase 65%. Em relação aos frameworks e bibliotecas JavaScript, React.js lidera o cenário. Isso sugere que a biblioteca, frequentemente tratada como framework, tem conseguido manter uma presença significativa. Esses dados oferecem uma perspectiva fascinante sobre as preferências e tendências atuais no desenvolvimento web, e servem como um indicativo valioso para desenvolvedores e empresas na seleção de tecnologias para projetos futuros.[Overflow 2021].

A metodologia empregada pelo StateofJs [Sta 2020] para quantificar a preferência pelos diferentes frameworks JavaScript envolveu a fórmula "Uso: (usaria novamente + não usaria novamente) / total". Segundo os resultados de 2020, o React liderou com um índice de 80%, apontando para uma notável aceitação e vontade de reutilização entre os desenvolvedores. Angular marcou 56%, revelando uma receptividade moderada na comunidade de desenvolvedores. Vue.js registrou um índice de 49%, sugerindo que ainda mantém um grupo significativo de usuários propensos a optar por ele novamente.

Taxa de Adoção de Frameworks e Bibliotecas em 2021

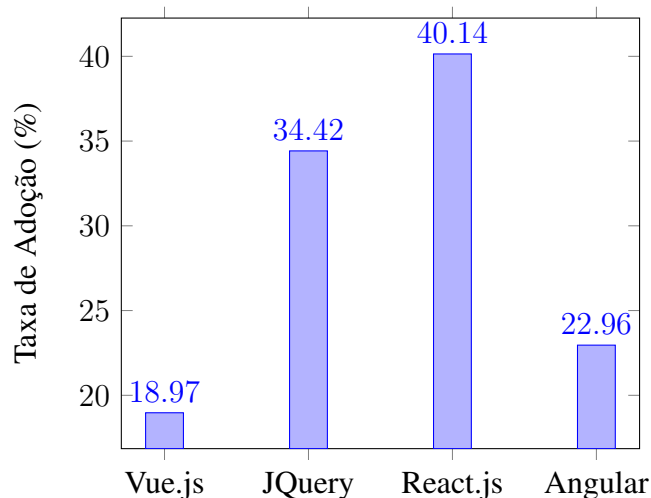


Figura 1. Dados fornecidos pelo StackOverflow, gráfico autoral.

Adoção de Frameworks JavaScript em 2020

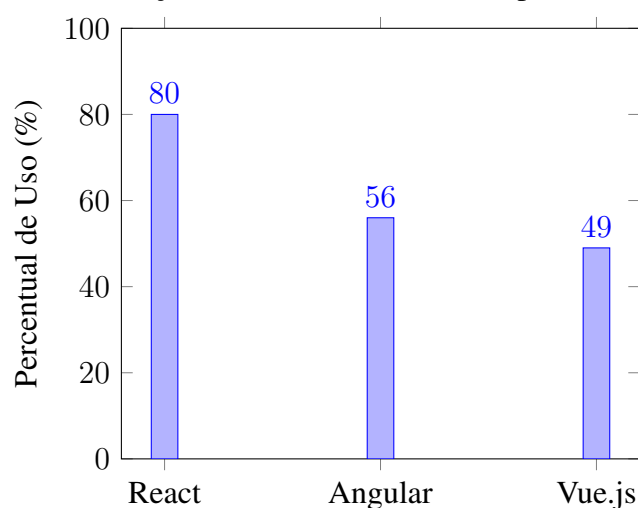
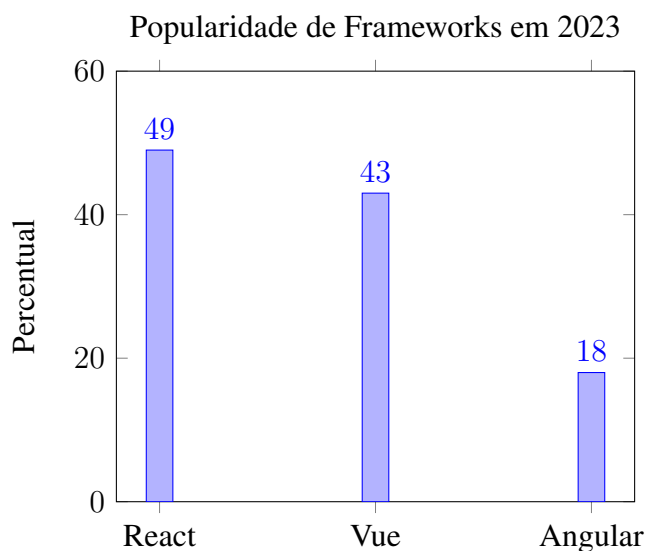


Figura 2. Dados fornecidos pelo StateofJs, gráfico autoral.

Na análise de dados recente fornecida pela JetBrains em 2021 [JetBrains 2021], é observável uma tendência clara nas preferências tecnológicas na comunidade de desenvolvimento web. De acordo com o gráfico, o React domina o cenário com uma fatia significativa de 49%, indicando sua ampla adoção e possível satisfação entre os desenvolvedores. Em contrapartida, o Vue.js também apresenta um forte desempenho com 43%, sugerindo que tem ganhado terreno como uma alternativa viável e popular ao React para desenvolvimento de aplicações de página única.

O Angular, embora uma escolha estabelecida em ambientes empresariais, demonstra uma menor presença com apenas 18%. Esse dado pode refletir uma preferência por outros frameworks mais modernos ou simplesmente indicar que o Angular está sendo cada vez mais reservado para projetos mais complexos e de grande escala, em vez de ser

a escolha padrão para desenvolvimento web geral.



**Figura 3. Dados fornecidos por JetBrains, gráfico autoral.**

Além disso, de acordo com um estudo comparativo realizado por [Pinto et al. 2023], no site do NPM, o framework que mais obteve downloads nesse ano de 2022 (até 12/11/2022) foi o React com mais de 705 milhões de downloads. Em segundo lugar está o Vue, com mais de 146 milhões, seguido pelo @angular/cli com cerca de 106 milhões.

Já no que diz respeito aos seus repositórios no GitHub, o React está na frente no requisito de uso em repositórios públicos, com 12.346.988 repositórios, enquanto o Angular possui 2.141.284 repositórios (visto em 12/11/2022, GitHub). Esse dado não foi encontrado para o Vue em sua página, logo não foi possível fazer a análise desse quesito para o framework. [Pinto et al. 2023]

Em suma, os dados apresentados por diferentes fontes, como Stack Overflow, State of JS, JetBrains e no estudo comparativo realizado por Sandy Hoffmann, Luana Alves Pinto e Luiz Ricardo Uriarte, proporcionam um panorama elucidativo sobre o atual ecossistema de frameworks e bibliotecas JavaScript. O React.js se destaca como uma escolha primordial entre os desenvolvedores, respaldado por métricas de satisfação e taxas de reutilização elevadas. Enquanto o Angular permanece uma opção sólida, especialmente em contextos empresariais, sua presença está diminuindo em comparação com alternativas mais recentes. O Vue.js, por sua vez, emerge como um competidor robusto, ganhando terreno e preferência entre os desenvolvedores.

Essas tendências refletem não apenas a evolução contínua das tecnologias, mas também os critérios mutáveis que os desenvolvedores utilizam para escolher uma tecnologia sobre outra. As informações compiladas são de valor inestimável para qualquer profissional ou organização em busca de insights para tomar decisões informadas na seleção de tecnologias para futuros projetos de desenvolvimento web.

## 4.2. Desempenho

No trabalho de [Anjos 2017] apresenta uma comparação metódica dos tempos de execução para várias operações realizadas em três frameworks front-end populares: Angular, React e Vue. É notável que Angular geralmente tem os tempos de execução mais rápidos para a maioria das tarefas, servindo como uma base de referência (indicado por percentuais próximos a zero). De forma mais ampla, o Angular se destaca como o líder em termos de eficiência, conquistando o menor tempo de execução em oito dos dez testes realizados. Este resultado pode ser atribuído à sua arquitetura otimizada e algoritmos de detecção de mudanças eficazes.

No entanto, é imperativo não negligenciar as áreas onde Angular falha significativamente. Nos testes 7 e 9, os tempos de execução de Angular foram drasticamente piores, o que sugere que há cenários específicos onde React ou Vue podem ser mais apropriados. Por exemplo, o React mostrou-se aproximadamente 18 vezes mais rápido que Angular no teste 7, e Vue foi quase três vezes mais rápido no teste 9. Essas são diferenças substanciais que não só são estatisticamente significativas, mas também seriam perceptíveis ao usuário final.

**Tabela 1. Comparação do tempo de execução entre todas as tecnologias**

Ação	Angular		React		Vue	
	ms	%	ms	%	ms	%
1. Adicionar 1.000 linhas em uma tabela vazia	312,47	0	419,64	34	448,11	43
2. Adicionar 10.000 linhas em uma tabela vazia	2129,09	0	3383,51	59	3345,84	57
3. Adicionar 1.000 linhas em uma tabela com 10.000 linhas	315,39	0	817,70	159	1198,41	280
4. Clicar na linha 5.000 em uma tabela com 10.000 registros	37,49	0	583,04	1455	805,85	2050
5. Clicar na primeira linha de uma tabela com 1.000 linhas	9,36	0	121,34	1197	137,90	1374
6. Excluir a linha 5.000 em uma tabela com 10.000 registros	702,85	0	1227,87	75	1432,71	104
7. Limpar tabela com 10.000 linhas	4273,17	1853	218,77	0	251,45	15
8. Realizar atualização parcial em uma tabela com 10.000 registros	597,29	0	1105,87	85	1433,65	140
9. Realizar ordenação crescente em uma tabela com 10.000 registros	7727,14	198	2739,84	6	2594,69	0



10. Trocar linhas em uma tabela com 10.000 registros	105,69	0	2440,54	2209	998,57	845
Total	16209,92	28	13058,11	3	12647,17	0

Os valores apresentados na tabela 2 são métricas de desempenho obtidas a partir de um teste de benchmarking que avaliou várias tecnologias e frameworks utilizados na criação e manipulação de tabelas em páginas da web. Benchmarking é muito útil para os desenvolvedores que desejam melhorar o desempenho de seus projetos de software. Essa técnica envolve medir e comparar o desempenho de um sistema com o de outro, a fim de identificar quais partes do código estão mais lentas e precisam ser otimizadas.

Cada linha na tabela corresponde a uma ação específica e fornece informações sobre a "Duração" dessa ação em milissegundos, acompanhada do respectivo "Desvio Padrão". Além disso, a tabela apresenta uma métrica denominada "Desaceleração" (Slowdown), que indica a relação entre a duração da ação e a duração da ação mais rápida.

O teste foi conduzido por [Krause 2023], e os valores de desaceleração (Slowdown) representam o quão mais lenta cada tecnologia ou framework é em comparação com a mais rápida em cada ação específica. Valores maiores indicam um desempenho inferior, onde uma desaceleração de 2, por exemplo, significa que a tecnologia é duas vezes mais lenta do que a mais rápida, enquanto uma desaceleração de 1 indica que elas têm desempenho equivalente.

A tabela de benchmarking oferece uma análise detalhada do desempenho das tecnologias Angular, Vue e React em várias ações relacionadas à criação e manipulação de tabelas em páginas da web. Ao avaliarmos essas três tecnologias amplamente utilizadas, observamos que o Angular, embora seja conhecido por sua robustez, apresentou tempos de execução relativamente mais longos em muitas ações, resultando em desacelerações significativas em relação à tecnologia mais rápida. Por outro lado, o Vue demonstrou um desempenho mais equilibrado, oferecendo tempos de execução competitivos na maioria das ações.

O React, reconhecido por sua popularidade e eficiência, manteve tempos de execução notavelmente curtos, resultando em desacelerações menores em comparação com as outras tecnologias. Essas conclusões enfatizam a importância de levar em consideração o desempenho ao escolher a tecnologia adequada para projetos web, especialmente quando a manipulação de tabelas desempenha um papel fundamental na aplicação.

**Tabela 2. Comparação de Desempenho entre Angular, React e Vue**

Ação	Angular	React	Vue
Criação de 1000 linhas após a página carregar	251.8 (1.8)	197.3 (1.4)	193.1 (1.4)
Atualização de todas as 1000 linhas da tabela	278.3 (1.9)	201.3 (1.4)	197.4 (1.3)
Atualização parcial a cada 10ª linha	12.5 (1.0)	12.8 (1.0)	13.0 (1.0)
Destaque de uma linha em resposta a um clique	8.1 (1.0)	4.9 (1.0)	3.4 (1.0)
Troca de posições de 2 linhas em uma tabela de 1000	14.7 (1.0)	13.5 (1.0)	13.4 (1.0)
Remoção de uma linha	47.4 (1.1)	46.4 (1.1)	46.1 (1.1)
Criação de 10.000 linhas	3108.7 (2.3)	1866.7 (1.4)	1946.0 (1.5)
Adição de 1000 linhas em uma tabela de 10.000 linhas	454.8 (1.6)	365.1 (1.3)	324.6 (1.1)
Limpeza de uma tabela com 10.000 linhas	817.6 (4.7)	389.9 (2.2)	379.9 (2.2)
Tempo de inicialização	118.1 (2.9)	104.9 (2.6)	84.3 (2.1)
Desaceleração (Geometric Mean)	1.69	1.36	1.31

## 5. Considerações Finais

Em conclusão, a pesquisa e análise realizadas aqui fornecem insights valiosos sobre as tecnologias de desenvolvimento front-end, com foco nas populares Angular, Vue e React, considerando os critérios de popularidade e desempenho. As descobertas destacam a importância de uma abordagem equilibrada ao escolher uma tecnologia para projetos web.

Primeiramente, a popularidade é um fator crucial a ser considerado. De acordo com o Stack Overflow Developer Survey 2021, React é uma das tecnologias mais populares, o que pode traduzir-se em maior suporte da comunidade, abundância de recursos de aprendizado e disponibilidade de profissionais qualificados. No entanto, é importante notar que o Vue também está ganhando popularidade rapidamente, o que sugere um crescimento contínuo em seu ecossistema de desenvolvedores.

É interessante notar que os resultados obtidos a partir da pesquisa de [Anjos 2017], que comparou os tempos de execução dos frameworks front-end Angular, React e Vue, e os resultados de benchmarks apresentados por [Krause 2023], que analisaram o desempenho dessas mesmas tecnologias em cenários específicos relacionados à criação e manipulação de tabelas em páginas da web, mostram algumas discrepâncias notáveis. Enquanto o estudo de [Anjos 2017] apontou o Angular como líder em eficiência na maioria das tarefas, os benchmarks indicam que o Angular geralmente apresentou tempos de execução mais longos e desacelerações significativas em comparação com o React e o Vue em várias ações específicas.

Essa discrepância pode ser atribuída a diferenças na metodologia de teste, nos cenários de aplicação ou até mesmo nas versões das tecnologias avaliadas. Essa divergência de resultados destaca a importância de considerar múltiplas fontes de informação e realizar testes específicos para o contexto de um projeto ao selecionar a tecnologia mais adequada para alcançar os objetivos de desempenho desejados.

Em última análise, a seleção da tecnologia certa deve ser baseada em uma análise aprofundada das necessidades do projeto, considerando tanto a popularidade quanto o desempenho. É importante lembrar que o cenário de desenvolvimento web está em constante evolução, e as tendências podem mudar ao longo do tempo. Portanto, manter-se atualizado com informações recentes e continuar monitorando o desempenho das tecnologias escolhidas é essencial para o sucesso contínuo de projetos web.

## Referências

- (2020). State of js 2020: Front-end frameworks. Acessado em: 9 de setembro de 2023.
- Anjos, L. F. R. d. (2017). Evolução do javascript em aplicações multiplataforma.
- Branas, R. (2014). *AngularJS essentials: Design and construct reusable, maintainable, and modular web applications with AngularJS*. Packt Publishing, Birmingham, United Kingdom.
- da Cruz Scotti, G. A. et al. (2019). Análise comparativa de front-ends de frameworks baseados em javascript.
- Elliott, E. (2017). Top javascript frameworks & topics to learn in 2017. Acessado em: 09 de Setembro de 2023.
- Ferreira Jr, J. et al. (2021). Comparação dos principais frameworks javascript para desenvolvimento web.
- Francisco, T. C. I. S. (2017). Stanford uni's intro to compsci course adopts javascript, bins java: Java's days are numbered – but it's a very large number. Acessado em: 09 de Setembro de 2023.
- Incau, C. (2017). *Vue.js: Construa aplicações incríveis*. Editora Casa do Código.
- JetBrains (2021). Desenvolvimento em javascript: Ecossistema developer 2021. Acessado em: 9 de setembro de 2023.
- Krause, S. (2023). Javascript frameworks benchmark.
- Morais, P. (2021). A história do react! Acessado em: 13 de Setembro de 2023.
- Overflow, S. (2021). Stack overflow developer survey 2021: Most popular technologies. Acessado em: 09 de Setembro de 2023.
- Pinto, L. A., Hoffmann, S., and Uriarte, L. R. (2023). Análise comparativa entre as tecnologias de front-end react, angular e vue. *REVISTA DE EXTENSÃO E INICIAÇÃO CIENTÍFICA DA UNISOCIESC*, 10(1).
- Roveda, U. (2023). O que é react: Para que serve, como funciona e características. Acessado em: 13 de Setembro de 2023.
- Team, V. (2022). Vue.js documentation. Acessado em: 13 de Setembro de 2023.
- Williamson, K. (2015). *Introdução ao AngularJS: Um guia para o desenvolvimento com o AngularJS*. Novatec Editora.