

**Uma Revisão da Literatura sobre Tendências e  
Tecnologias no Desenvolvimento Front-End  
A Literature Review on Trends and  
Technologies in Front-End Development**

**Robson Kerner Coelho dos Santos<sup>1</sup>, Rafael Kerner Coelho dos Santos<sup>2</sup>, André Luiz  
Antunes Lopes<sup>3</sup>**

<sup>1</sup>Faculdade Unis São Lourenço, São Lourenço, Minas Gerais,  
[robson.santos@professor.unis.edu.br](mailto:robson.santos@professor.unis.edu.br), 0000-0003-4163-5486

<sup>2</sup>Faculdade Unis São Lourenço, São Lourenço, Minas Gerais,  
[rafael.santos@professor.unis.edu.br](mailto:rafael.santos@professor.unis.edu.br)

<sup>3</sup>Faculdade Unis São Lourenço, São Lourenço, Minas Gerais,  
[andre.lopes@professor.unis.edu.br](mailto:andre.lopes@professor.unis.edu.br)

## **RESUMO**

Este estudo apresenta uma revisão da literatura sobre tendências e tecnologias no desenvolvimento *Front-End*, com o objetivo de mapear as principais ferramentas, frameworks e abordagens utilizadas no cenário contemporâneo da engenharia de software. Foram analisadas quatorze publicações recentes (2024–2025), contemplando artigos, dissertações e trabalhos de conclusão de curso que abordam comparações de desempenho, modernização tecnológica e o impacto da inteligência artificial na automação do desenvolvimento web. Os resultados apontam que frameworks como *React*, *Angular* e *Vue.js* continuam dominando o ecossistema de desenvolvimento, embora soluções emergentes como *Svelte* e *Next.js* apresentem desempenho competitivo e crescente adoção. A literatura evidencia, ainda, a incorporação de arquiteturas limpas (*Clean Architecture*) e de modelos baseados em IA como tendência de convergência entre engenharia de software e automação cognitiva. Entre os desafios identificados, destacam-se a ausência de padronização nas métricas de comparação e a necessidade de constante atualização profissional. Conclui-se que o desenvolvimento *Front-End* moderno caminha para um modelo mais inteligente, modular e colaborativo, no qual a inteligência artificial atua como ferramenta de apoio, potencializando a produtividade e a qualidade das interfaces sem substituir o papel criativo e estratégico do desenvolvedor.

**Palavras-chave:** Desenvolvimento *web*; *Front-End*; *Frameworks*; Inteligência artificial; Engenharia de software.

# 1 INTRODUÇÃO

O avanço contínuo das tecnologias web e o surgimento de novos frameworks de desenvolvimento *Front-End* têm transformado profundamente a forma como aplicações e interfaces digitais são projetadas, desenvolvidas e mantidas. Essa área, que outrora era dominada por linguagens e ferramentas estáticas, hoje é caracterizada por um ambiente dinâmico e competitivo, no qual decisões tecnológicas afetam diretamente a produtividade, a escalabilidade e a experiência do usuário. Nesse contexto, compreender as tendências e os critérios de escolha das ferramentas Front-End tornou-se essencial para desenvolvedores, empresas e instituições de ensino que buscam acompanhar a evolução do setor.

O desenvolvimento *Front-End* não envolve apenas a criação de interfaces visuais, mas também a integração entre usabilidade, desempenho e arquitetura de software. A escolha do *framework* ideal impacta diretamente o tempo de desenvolvimento, a manutenibilidade e a curva de aprendizado das equipes envolvidas. Segundo Marchioro (2025), essa decisão configura-se como um dilema recorrente, uma vez que envolve trade-offs entre desempenho, facilidade de uso e adequação técnica. Enquanto o React demonstra melhor equilíbrio entre performance e modularidade, o Vue.js destaca-se pela acessibilidade e simplicidade de adoção, e o Angular pela robustez e suporte corporativo. Ainda assim, não há consenso sobre qual tecnologia se sobressai de forma universal, pois a efetividade depende fortemente do contexto do projeto e da maturidade da equipe.

No cenário brasileiro, Araújo e Muniz (2025) identificam um ecossistema híbrido, no qual convivem soluções modernas e legadas, refletindo as diferenças de ritmo de adoção tecnológica entre empresas de diversos portes e setores. Tecnologias amplamente difundidas, como React, Angular e Vue.js, dominam o mercado, mas bibliotecas como jQuery ainda mantêm presença significativa em sistemas corporativos. Essa coexistência revela uma lacuna importante: apesar da expansão tecnológica, faltam estudos que sistematizem e comparem as tendências atuais do desenvolvimento Front-End no país, considerando tanto aspectos técnicos quanto estratégicos de sua adoção.

Além dos desafios técnicos, a literatura destaca a necessidade de se adotar princípios arquiteturais sólidos para sustentar a modernização dos sistemas. Lima (2025) observa que práticas como componentização, modularização e a aplicação de arquitetura limpa são fundamentais para garantir escalabilidade e longevidade aos projetos web. A

modernização, portanto, vai além da simples substituição de frameworks: envolve repensar a estrutura e o ciclo de vida das aplicações, considerando desempenho, testabilidade e integração contínua.

Diante desse contexto, o problema de pesquisa que orienta este estudo é: quais são as principais tendências e tecnologias que moldam o desenvolvimento Front-End contemporâneo, e como os diferentes frameworks se posicionam em termos de desempenho, curva de aprendizado e arquitetura? O objetivo é revisar a literatura recente sobre o tema, identificando as tecnologias predominantes, as métricas de comparação mais utilizadas e os desafios enfrentados pelos profissionais da área.

Este trabalho busca oferecer uma síntese crítica que auxilie pesquisadores, docentes e desenvolvedores a compreenderem a evolução do desenvolvimento Front-End e suas implicações práticas. Ao reunir evidências sobre as principais ferramentas, metodologias e diretrizes arquiteturais, pretende-se contribuir para a formação de um referencial teórico que apoie decisões técnicas e acadêmicas voltadas à inovação, eficiência e sustentabilidade no desenvolvimento de interfaces web.

## **2 MATERIAL E MÉTODOS**

O presente estudo caracteriza-se como uma pesquisa básica, de natureza qualitativa e exploratória, voltada à análise das tendências e tecnologias emergentes no ecossistema de desenvolvimento Front-End moderno. O objetivo central é mapear, compreender e interpretar como diferentes frameworks, bibliotecas e arquiteturas têm evoluído e influenciado a prática do desenvolvimento web contemporâneo, considerando tanto aspectos técnicos quanto suas implicações no desempenho e na experiência do usuário.

Trata-se de uma revisão da literatura, desenvolvida a partir da análise de produções científicas publicadas entre os anos de 2024 e 2025 — período marcado pela consolidação de frameworks reativos, pela integração crescente de soluções baseadas em Inteligência Artificial (IA) e pela modernização das práticas de engenharia de software voltadas à camada de apresentação.

A pesquisa adotou uma abordagem predominantemente qualitativa, com ênfase na interpretação dos achados descritos nos estudos analisados. No entanto, foram considerados também elementos quantitativos presentes nas publicações, como métricas

de desempenho técnico (tempo de renderização, uso de CPU e memória) e indicadores de adoção tecnológica, de modo a ampliar a compreensão sobre a eficiência e a popularidade das soluções examinadas.

A estratégia de busca envolveu o uso de bases de dados e repositórios acadêmicos amplamente reconhecidos, como Google Scholar, IEEE Xplore, ACM Digital Library, SpringerLink, Taylor & Francis Online, arXiv e repositórios institucionais de universidades brasileiras, como os da Universidade Federal do Ceará (UFC), Universidade Federal de Uberlândia (UFU) e Instituto Federal da Paraíba (IFPB). Foram priorizados estudos revisados por pares (peer-reviewed) e trabalhos acadêmicos de acesso aberto, garantindo a confiabilidade e a atualidade das fontes utilizadas.

Os descritores empregados nas buscas incluíram termos em português e inglês, de forma a ampliar o escopo dos resultados e assegurar a representatividade internacional do tema. As principais expressões utilizadas foram: “comparativo entre frameworks front-end”, “framework front end comparison”, “React Angular Vue performance”, “modernização front-end”, “technical performance”, “curva de aprendizado” e “learning curve”.

A coleta de dados foi realizada entre os meses de agosto e outubro de 2025, considerando publicações a partir de 2024. Os resultados foram inicialmente filtrados pela leitura de títulos e resumos, selecionando-se os estudos mais aderentes ao tema central. Em seguida, procedeu-se à leitura integral dos textos para identificar metodologias aplicadas, resultados obtidos e conclusões relevantes à análise comparativa de frameworks e tendências tecnológicas.

Foram incluídos na amostra artigos e trabalhos que apresentavam comparações entre dois ou mais frameworks Front-End (como React, Angular, Vue, Svelte ou Next.js), avaliações de desempenho técnico, análises de curva de aprendizado e usabilidade, bem como estudos que abordavam tendências arquiteturas (Progressive Web Apps, Microfrontends e Clean Architecture). Excluíram-se trabalhos com foco exclusivo em desenvolvimento Back-End, em interface de usuário (UI) sem abordagem tecnológica ou em temas de baixo nível sem relação direta com o Front-End.

Os dados obtidos foram organizados de modo a evidenciar quatro eixos de análise principais: (1) frameworks e bibliotecas mais citados, (2) métricas de desempenho

técnico, (3) aspectos arquiteturais e boas práticas de engenharia de software e (4) tendências emergentes relacionadas ao uso de Inteligência Artificial e à automação do desenvolvimento Front-End. Essa sistematização permitiu uma leitura crítica e integrada da literatura, oferecendo uma visão abrangente sobre o estado atual e as perspectivas de evolução das tecnologias que sustentam o desenvolvimento de interfaces modernas na web.

### 3 REFERENCIAL TEÓRICO

O referencial teórico deste estudo foi construído a partir da análise de 14 artigos publicados entre 2024 e 2025, que abordam o desenvolvimento Front-End sob diferentes perspectivas: comparações entre frameworks, desempenho técnico, evolução arquitetural, usabilidade, modernização de sistemas e impacto de novas tecnologias emergentes, como inteligência artificial e automação de código. Esses estudos permitem compreender como o ecossistema Front-End vem se consolidando e diversificando nos últimos anos, revelando as tendências de adoção de frameworks, as boas práticas de engenharia e as métricas mais utilizadas para avaliar desempenho e produtividade.

O Quadro 1 sintetiza as principais informações dos trabalhos analisados, destacando seus autores, ano de publicação, foco temático e contribuições para a discussão sobre as tecnologias e tendências no desenvolvimento Front-End contemporâneo.

**Quadro 1** ARTIGOS SELECIONADOS

Título do Estudo	Ano
Tecnologias web utilizadas nas maiores empresas do Brasil	2025
Uma análise comparativa de frameworks de desenvolvimento web	2025
PSD2Code: Automated front-end code generation from design files via multimodal LLMs	2025
Two approaches to fast bytecode frontend for static analysis	2025
Um relato de uso dos frameworks Ruby on Rails, Laravel e Django	2025
Análise comparativa do desenvolvimento de aplicativos mobile: híbrido vs. nativo	2025
Um estudo comparativo de plataformas de desenvolvimento full-stack orientadas por IA	2025
Benchmarking modern frontend frameworks: A comparative analysis of rendering performance	2025
Technical Performance Comparison of Modern Frontend Frameworks	2025
Análise de desempenho de frameworks para aplicações distribuídas	2025
Desenvolvimento front-end da versão web do aplicativo AMO	2025
Modernização tecnológica de front-end: Uma proposta de interface de interação com usuário	2025
Comparativa de Frameworks de Desarrollo Backend	2025
FrontendBench: A benchmark for evaluating LLMs on front-end development	2025

**Fonte:** elaborado pelos autores com os dados gerados no portal Google Acadêmico (2025).

Os trabalhos de Putra et al. (2025) e Ouchaib (2025) abordam a comparação de desempenho entre frameworks modernos de Front-End, oferecendo análises complementares que reforçam a importância da escolha tecnológica no desenvolvimento de interfaces web. Ambos os estudos partem da premissa de que frameworks como React, Vue.js e Svelte desempenham papel central na criação de aplicações dinâmicas, e que a avaliação de métricas técnicas é fundamental para decisões de arquitetura e otimização de sistemas.

Em seu estudo experimental, Putra et al. (2025) desenvolveram aplicações equivalentes utilizando Svelte, React e Vue, comparando-as em métricas como tempo de renderização, tamanho do bundle, uso de memória e eficiência do DOM. Os resultados indicaram que o Svelte apresentou desempenho superior em tamanho de arquivo e velocidade de renderização, enquanto o React demonstrou melhor equilíbrio entre desempenho e estabilidade. O trabalho destaca ainda a necessidade de padronizar critérios de avaliação, uma vez que muitos estudos anteriores se concentravam em métricas isoladas, o que dificultava comparações precisas.

De forma complementar, Ouchaib (2025) realizou uma análise aprofundada de performance de renderização em frameworks modernos, considerando diferentes cenários de carga e complexidade de interface. Os resultados corroboram parte das conclusões de Putra et al. (2025), reforçando o destaque de React em ambientes corporativos e de Svelte em contextos que priorizam leveza e baixo consumo de recursos. Ambos os autores concluem que não há um framework universalmente superior, mas sim tecnologias mais adequadas a contextos específicos de uso, o que reforça a importância de abordagens comparativas e empíricas na pesquisa em desenvolvimento Front-End. Os estudos de Cavalcante (2025) e Araújo e Muniz (2025) contribuem de forma complementar para a compreensão do cenário contemporâneo de desenvolvimento web e da adoção de tecnologias front-end no contexto organizacional. Enquanto Cavalcante (2025) realiza uma análise comparativa entre frameworks de desenvolvimento web, com foco em desempenho e curva de aprendizado, Araújo e Muniz (2025) apresentam uma visão empírica do uso dessas tecnologias em grandes empresas brasileiras, destacando as ferramentas mais consolidadas e os fatores que influenciam suas escolhas.

Cavalcante (2025) ressalta que a decisão sobre qual framework adotar é estratégica e impacta diretamente na produtividade e escalabilidade dos projetos. Em sua análise experimental, React e Angular demonstraram desempenho superior em aplicações corporativas de alta complexidade, enquanto Vue.js destacou-se pela simplicidade e menor curva de aprendizado. Já Araújo e Muniz (2025) identificaram um ecossistema híbrido no ambiente empresarial, onde tecnologias modernas coexistem com soluções legadas. No cenário brasileiro, React se sobressai como a tecnologia mais adotada, seguida de Angular e Vue.js, embora o uso de bibliotecas como jQuery ainda seja recorrente em sistemas consolidados.

Ambos os trabalhos convergem na conclusão de que a seleção tecnológica deve considerar não apenas métricas de desempenho, mas também aspectos humanos e organizacionais, como a qualificação das equipes, os custos de migração e o suporte comunitário. As evidências apresentadas reforçam a importância de decisões baseadas em evidências empíricas e estratégicas, equilibrando inovação e estabilidade no ecossistema de desenvolvimento web moderno.

Os trabalhos de Santos et al. (2025) e Lima (2025) exploram o desempenho e a aplicabilidade de frameworks em diferentes contextos de desenvolvimento, contribuindo para o entendimento das relações entre eficiência técnica e experiência prática no uso dessas ferramentas. Enquanto Santos et al. (2025) abordam o desempenho de frameworks voltados a aplicações distribuídas, ampliando a discussão sobre a escalabilidade e o consumo de recursos, Lima (2025) apresenta um relato prático de uso dos frameworks Ruby on Rails, Laravel e Django, destacando percepções reais de desempenho e produtividade em projetos web.

No estudo de Santos et al. (2025), os autores realizaram uma análise experimental comparando frameworks de back-end utilizados em sistemas distribuídos, como Flask, Actix e ASP.NET, avaliando parâmetros como uso de CPU, tempo de resposta e consumo de memória. Embora o foco principal seja o back-end, os resultados oferecem insights relevantes para o desenvolvimento full-stack, uma vez que o desempenho do servidor impacta diretamente a experiência do usuário final. O estudo conclui que a escolha do framework ideal depende da natureza da aplicação e do perfil de carga de trabalho, destacando a importância do alinhamento entre arquitetura, desempenho e eficiência energética.

Por sua vez, Lima (2025) apresenta uma análise qualitativa fundamentada na experiência prática de uso de três frameworks amplamente reconhecidos no mercado. A autora destaca que, embora Ruby on Rails ofereça rapidez no desenvolvimento inicial, Laravel apresenta maior flexibilidade e Django se sobressai pela robustez e segurança. O relato evidencia que a produtividade e a curva de aprendizado variam conforme o perfil do desenvolvedor e o tipo de aplicação, reforçando que a escolha do framework deve equilibrar aspectos técnicos e humanos. Ambos os estudos convergem ao demonstrar que a performance de um sistema não depende apenas de métricas técnicas, mas também da adequação da ferramenta ao contexto de uso e à competência da equipe envolvida.

Os trabalhos de Nogueira (2025) e Valle (2025) abordam a modernização tecnológica e o uso de inteligência artificial no contexto do desenvolvimento web, evidenciando uma transição da engenharia tradicional para modelos de desenvolvimento mais integrados, inteligentes e orientados à experiência do usuário. Ambos os estudos exploram a relação entre inovação tecnológica e eficiência de projeto, ainda que sob perspectivas distintas: Nogueira (2025) analisa plataformas de desenvolvimento full-stack impulsionadas por IA, enquanto Valle (2025) propõe a modernização de interfaces Front-End em sistemas acadêmicos.

Em sua pesquisa, Nogueira (2025) realiza uma análise comparativa de plataformas de desenvolvimento full-stack baseadas em inteligência artificial, avaliando o impacto dessas soluções na automação de código, redução de tempo de desenvolvimento e consistência de resultados. O autor destaca que as ferramentas orientadas por IA — como geradores automáticos de código e assistentes inteligentes — vêm modificando o papel do desenvolvedor, deslocando o foco de tarefas repetitivas para o aprimoramento do design e da arquitetura. A pesquisa conclui que essas tecnologias ampliam a produtividade, mas exigem maturidade técnica e revisão contínua da qualidade do código gerado.

Já o estudo de Valle (2025) propõe a modernização tecnológica de uma interface de interação com o usuário, redesenhando o Front-End do sistema “Classroom eXperience”. O autor fundamenta a proposta em princípios de usabilidade, componentização e arquitetura limpa (Clean Architecture), buscando aprimorar a experiência do usuário e garantir maior escalabilidade e manutenção. A pesquisa demonstra como a adoção de padrões modernos de engenharia pode revitalizar sistemas

legados, mantendo a compatibilidade com tecnologias anteriores. Em conjunto, os dois estudos reforçam a tendência de convergência entre automação, arquitetura limpa e experiência do usuário como pilares do desenvolvimento Front-End contemporâneo.

Os estudos de Chen e Chen (2025) e Li et al. (2025) exploram a interseção entre inteligência artificial e desenvolvimento Front-End, evidenciando como algoritmos avançados e abordagens baseadas em aprendizado profundo estão redefinindo os limites da automação no ciclo de desenvolvimento de software. Ambos os trabalhos partem do princípio de que o uso de IA pode reduzir significativamente o esforço humano em tarefas complexas, como geração de código e modelagem de interfaces, sem comprometer a qualidade técnica das entregas.

O trabalho de Chen e Chen (2025) apresenta o PSD2Code, um sistema de geração automatizada de código Front-End a partir de arquivos de design, utilizando modelos de linguagem multimodais (LLMs). A pesquisa demonstra como a IA é capaz de interpretar elementos visuais — como cores, camadas e hierarquias — e transformá-los em componentes de interface prontos para uso. Os resultados indicam ganhos expressivos em velocidade de desenvolvimento e consistência de design, além de reduzirem a dependência de prototipação manual. No entanto, os autores ressaltam a necessidade de supervisão humana para ajustes de usabilidade e aderência ao contexto do projeto.

De modo complementar, Li et al. (2025) introduzem o sistema CVD-SfM, uma arquitetura baseada em visão computacional que propõe um Front-End profundo para reconstrução de estruturas tridimensionais em cenários de múltiplas altitudes. Embora voltado à área de visão computacional e robótica, o estudo contribui para o campo de desenvolvimento Front-End ao demonstrar o potencial da IA em processar e renderizar dados espaciais complexos em tempo real. Juntos, os dois trabalhos ilustram o avanço do paradigma de “IA como co-desenvolvedora”, no qual a automação não substitui o programador, mas amplia suas capacidades criativas e analíticas no processo de desenvolvimento. Os trabalhos de Zhu et al. (2025) e Yánez Flores e Jaramillo Flores (2025) expandem a discussão sobre o desenvolvimento web ao abordar, respectivamente, o papel da inteligência artificial na avaliação automatizada de desempenho de frameworks Front-End e as comparações estruturais entre plataformas de desenvolvimento Back-End. Embora atuem em dimensões distintas da engenharia de

software, ambos os estudos reforçam a importância da avaliação empírica e da integração entre camadas para aprimorar a eficiência e a escalabilidade das aplicações modernas.

O estudo conduzido por Zhu et al. (2025) introduz o FrontendBench, um benchmark automatizado para avaliar a capacidade de modelos de linguagem (LLMs) em tarefas de desenvolvimento Front-End. A pesquisa demonstra que a IA pode ser utilizada não apenas para gerar código, mas também para avaliar a qualidade, desempenho e aderência às boas práticas de programação. O trabalho estabelece métricas de correção funcional, eficiência e legibilidade, propondo um novo paradigma de testes baseados em aprendizado de máquina. Os autores concluem que a aplicação de benchmarks inteligentes pode otimizar o ciclo de desenvolvimento, promovendo maior padronização e reprodutibilidade dos resultados.

Já o estudo de Yáñez Flores e Jaramillo Flores (2025) analisa comparativamente os frameworks Spring e Struts, identificando diferenças significativas em termos de arquitetura, desempenho e manutenção. Embora o foco seja o desenvolvimento Back-End, a pesquisa contribui para o entendimento do ecossistema full-stack, evidenciando a importância da integração entre camadas de aplicação. Os resultados apontam que o Spring oferece melhor modularidade e suporte à escalabilidade, enquanto o Struts se destaca pela simplicidade de configuração. Em conjunto, os dois trabalhos reafirmam a tendência de integração entre automação, inteligência artificial e engenharia de software, consolidando uma visão holística do desenvolvimento web contemporâneo.

#### **4 RESULTADOS E DISCUSSÃO**

Os estudos analisados revelam um cenário de transformação acelerada no ecossistema de desenvolvimento Front-End, marcado pela coexistência entre tecnologias consolidadas e novas abordagens baseadas em inteligência artificial e arquitetura limpa. De modo geral, observa-se um consenso entre os autores (Putra et al., 2025; Ouchaib, 2025; Cavalcante, 2025; Araújo & Muniz, 2025) de que a escolha de frameworks e bibliotecas front-end não é apenas uma decisão técnica, mas estratégica, influenciando diretamente o desempenho, a produtividade e a sustentabilidade dos projetos.

Em relação ao desempenho técnico, os trabalhos de Putra et al. (2025) e Ouchaib (2025) destacam que frameworks como Svelte, React e Vue.js apresentam diferenças significativas em termos de renderização, consumo de memória e tamanho do bundle,

sendo o Svelte mais leve e eficiente em projetos menores, enquanto o React se mostra mais estável em aplicações de larga escala. Essa conclusão é reforçada por Cavalcante (2025), que aponta a necessidade de equilibrar desempenho com curva de aprendizado e maturidade da equipe. Em complemento, Santos et al. (2025) evidenciam que a eficiência do back-end (como em Flask, Actix ou ASP.NET) também impacta diretamente o desempenho do Front-End, sugerindo uma visão integrada de performance entre camadas.

No que se refere à adoção tecnológica, Araújo e Muniz (2025) identificam um ecossistema híbrido nas empresas brasileiras, no qual tecnologias modernas convivem com soluções legadas, como o uso persistente de jQuery em sistemas corporativos. Essa coexistência também é observada em Lima (2025), que, ao comparar Ruby on Rails, Laravel e Django, reforça que a manutenção de ferramentas tradicionais está associada à estabilidade e à familiaridade das equipes, mesmo diante da ascensão de soluções mais recentes. Essa resistência à substituição total de tecnologias demonstra que o processo de modernização é gradual e fortemente condicionado a fatores humanos e organizacionais.

A modernização tecnológica é um ponto de convergência importante entre diversos estudos. Nogueira (2025) e Valle (2025) destacam que o avanço de ferramentas baseadas em IA e a adoção de padrões arquiteturais modernos, como a Clean Architecture e a componentização, têm redefinido a forma como aplicações são projetadas e mantidas. O uso de IA em tarefas de automação e geração de código, conforme demonstrado por Chen & Chen (2025) e Zhu et al. (2025), amplia o potencial produtivo das equipes, permitindo o desenvolvimento de interfaces e protótipos a partir de modelos de linguagem multimodais e benchmarks automatizados. Entretanto, os autores alertam que essas ferramentas ainda dependem de validação humana, tanto para garantir a usabilidade quanto para evitar erros estruturais no código.

Além disso, os estudos de Li et al. (2025) e Zhu et al. (2025) evidenciam que a integração entre IA e Front-End está se expandindo para domínios mais complexos, como visão computacional e análise de desempenho automatizada, demonstrando que a inteligência artificial não apenas apoia o desenvolvimento, mas também avalia a eficiência e a qualidade do software. Essa nova fronteira, também observada nos trabalhos de Yánez Flores e Jaramillo Flores (2025), aponta para uma integração crescente entre Front-End, Back-End e camadas inteligentes, reforçando o conceito de ecossistema full-stack inteligente.

De maneira geral, as pesquisas convergem em três grandes eixos de discussão: (a) o aprimoramento do desempenho técnico e arquitetural dos frameworks, (b) a incorporação da inteligência artificial e automação no ciclo de desenvolvimento, e (c) a modernização de sistemas legados para padrões mais sustentáveis e escaláveis. Apesar dos avanços, os estudos também apontam desafios persistentes, como a falta de padronização em métricas de comparação, a dependência da experiência da equipe de desenvolvimento e a necessidade de constante atualização tecnológica.

Em síntese, a literatura analisada indica que o desenvolvimento Front-End caminha para um modelo mais inteligente, integrado e orientado à performance, no qual a IA atua como ferramenta de apoio e não de substituição, e as decisões de arquitetura assumem caráter cada vez mais estratégico. A convergência entre engenharia de software, usabilidade e automação cognitiva define o novo paradigma do desenvolvimento web contemporâneo, confirmando que o sucesso dos projetos depende tanto da inovação tecnológica quanto da maturidade das equipes e da sustentabilidade das práticas adotadas.

## **5 CONSIDERAÇÕES FINAIS**

O estudo evidencia que o desenvolvimento Front-End tem passado por uma transformação profunda, impulsionada por avanços tecnológicos, pela disseminação de novas arquiteturas e pelo uso crescente de inteligência artificial em etapas do ciclo de desenvolvimento. As análises mostram que as tecnologias modernas têm ampliado a capacidade das equipes de desenvolvimento em criar sistemas mais eficientes, responsivos e sustentáveis, ao mesmo tempo em que exigem maior qualificação técnica e constante atualização profissional. Entre os principais benefícios observados, destacam-se a melhoria no desempenho das interfaces, a padronização de componentes e a integração entre ferramentas de automação e frameworks reativos.

Os resultados apontam que React, Angular e Vue.js permanecem como as principais referências no ecossistema Front-End contemporâneo, devido à maturidade, ao suporte da comunidade e à ampla documentação. No entanto, frameworks emergentes como Svelte e Next.js vêm ganhando espaço por oferecer simplicidade, leveza e desempenho superior em determinados contextos. Além disso, o uso de arquiteturas limpas (Clean Architecture) e a crescente integração com plataformas baseadas em IA, conforme relatado por autores como Nogueira (2025), Chen e Chen (2025) e Zhu et al.

(2025), sinalizam uma tendência de convergência entre engenharia de software tradicional e desenvolvimento assistido por algoritmos inteligentes.

Entre os desafios identificados, a literatura aponta a ausência de padronização em métricas de desempenho e a dependência da experiência da equipe de desenvolvimento como fatores críticos para a adoção efetiva das tecnologias. A modernização de sistemas legados, embora necessária, enfrenta barreiras estruturais e culturais, como resistência à mudança, custos de migração e limitação de mão de obra especializada. Esses obstáculos reforçam a importância de estratégias organizacionais que combinem inovação tecnológica, capacitação profissional e governança de software.

O estudo também demonstra que a inteligência artificial não substitui o desenvolvedor, mas atua como coadjuvante estratégico, ampliando a produtividade e reduzindo a carga de tarefas repetitivas. Ferramentas baseadas em IA, como geradores automáticos de código e avaliadores de desempenho (Chen & Chen, 2025; Zhu et al., 2025), têm potencial para redefinir práticas de desenvolvimento, mas ainda demandam validação humana para garantir qualidade, acessibilidade e aderência ao contexto de uso.

Para pesquisas futuras, recomenda-se o aprofundamento de estudos empíricos que mensurem o impacto da automação e da IA no desempenho real das aplicações, bem como investigações sobre modelos híbridos de desenvolvimento que integrem colaboração entre humanos e sistemas inteligentes. Também se sugere a criação de frameworks avaliativos padronizados, que considerem não apenas métricas técnicas, mas também aspectos de usabilidade, acessibilidade e sustentabilidade do código.

De forma geral, conclui-se que o desenvolvimento Front-End contemporâneo caminha para um modelo mais inteligente, modular e colaborativo, no qual a sinergia entre inovação tecnológica, arquitetura de software e capacitação humana será determinante para o sucesso e a longevidade das soluções digitais.

### ***ABSTRACT***

*This study presents a literature review on trends and technologies in Front-End development, aiming to map the main tools, frameworks, and approaches that characterize the current landscape of software engineering. Fourteen recent publications (2024–2025) were analyzed, including articles, theses, and undergraduate projects that explore performance comparisons, technological modernization, and the impact of artificial intelligence on web development automation. The results indicate that frameworks such as React, Angular, and Vue.js remain dominant in the ecosystem, while*

*emerging solutions like Svelte and Next.js show competitive performance and growing adoption. The literature also highlights the integration of Clean Architecture principles and AI-driven models, signaling a convergence between traditional software engineering and cognitive automation. Among the main challenges identified are the lack of standardized performance metrics and the ongoing need for professional upskilling. Overall, the findings suggest that modern Front-End development is evolving toward a smarter, modular, and collaborative model, in which artificial intelligence functions as a support tool that enhances productivity and interface quality without replacing the developer's creative and strategic role.*

**Keywords:** *Web development; Front-End; Frameworks; Artificial intelligence; Software engineering.*

### **Agradecimentos**

A realização deste trabalho foi possível graças ao empenho conjunto e à colaboração entre colegas e profissionais que compartilham o mesmo compromisso com a pesquisa e o desenvolvimento científico. Agradecemos à instituição de ensino, pelo incentivo constante à produção acadêmica e pela disponibilização de recursos que viabilizaram o desenvolvimento deste estudo. Estendemos nossa gratidão aos colegas de trabalho e professores parceiros, que contribuíram com discussões, ideias e revisões construtivas ao longo do processo, enriquecendo a qualidade e a profundidade da análise. Reconhecemos também o apoio institucional e a troca de experiências no ambiente acadêmico, fundamentais para a consolidação deste projeto. Por fim, expressamos nosso apreço a todos que, direta ou indiretamente, ofereceram incentivo, colaboração e inspiração para a realização deste artigo. Este trabalho reflete não apenas um esforço individual, mas o resultado de uma rede de cooperação e comprometimento com o avanço do conhecimento na área de tecnologia e desenvolvimento de software.

### **REFERÊNCIAS**

Araújo, M. E. C. S., & Muniz, R. V. C. (2025). **Tecnologias web utilizadas nas maiores empresas do Brasil** [Trabalho de Conclusão de Curso, Instituto Federal da Paraíba, Campus Campina Grande]. Repositório Digital do IFPB.

Cavalcante, G. dos S. (2025). **Uma análise comparativa de frameworks de desenvolvimento web** [Trabalho de Conclusão de Curso, Universidade Federal do Ceará]. Repositório Institucional da UFC. <http://repositorio.ufc.br/handle/riufc/83076>

Creswell, J. W. (2010) **Projeto de pesquisa: métodos qualitativo, quantitativo e misto**. 2. ed. Porto Alegre: Artmed, 2010.

Chen, Y., & Chen, L. (2025). **PSD2Code: Automated front-end code generation from design files via multimodal large language models**. arXiv preprint arXiv:2511.04012. <https://arxiv.org/abs/2511.04012>

Li, C., Lin, H., Tan, T., & Li, Y. (2025). **Two approaches to fast bytecode frontend for static analysis**. *Proceedings of the ACM on Programming Languages*, 9(OOPSLA2), Article 303. <https://doi.org/10.1145/3763081>

Lima, V. N. (2025). **Um relato de uso dos frameworks Ruby on Rails, Laravel e Django** [Trabalho de Conclusão de Curso, Universidade Federal do Ceará]. Repositório Institucional da UFC. <http://repositorio.ufc.br/handle/riufc/80188>

Marchioro, G. D. N. (2025). **Análise comparativa do desenvolvimento de aplicativos mobile: híbrido vs. nativo**. *Revista Vincci – Periódico Científico do UniSATC*, 10(1), 117–133. <https://doi.org/10.70185/2525-6025.2025.v10.445>

Nogueira, L. F. (2025). **Um estudo comparativo de plataformas de desenvolvimento full-stack orientadas por IA** [Trabalho de Conclusão de Curso, Universidade Federal do Ceará]. Repositório Institucional da UFC. <http://repositorio.ufc.br/handle/riufc/83068>

Ouchaib, J. (2025). **Valutazione comparativa delle prestazioni di rendering in framework frontend moderni = Benchmarking modern frontend frameworks: A comparative analysis of rendering performance** [Master's thesis, Politecnico di Torino]. Webthesis. <https://webthesis.biblio.polito.it/35401/>

Putra, F. P. E., Hasbullah, H., Muslim, F., & Paradina, R. (2025). **Technical Performance Comparison of Modern Frontend Frameworks Study on Svelte, React, and Vue**. *Brilliance: Research of Artificial Intelligence*, 5(1), 355–364.

Santos, A. L. G. dos, Carneiro, C. A., Ivanchechen, J. H., Vendramin, A. C. B. K., & Pigatto, D. F. (2025). **Análise de desempenho de frameworks para aplicações distribuídas**. *Anais do Computer on the Beach*, 16, 486–493.

Silva, F. C. de S. (2025). **Desenvolvimento front-end da versão web do aplicativo AMO (Ambiente de Monitoria Online)** [Trabalho de Conclusão de Curso, Universidade Federal do Ceará]. Repositório da UFC. <http://repositorio.ufc.br/handle/riufc/82344>

Valle, M. M. do. (2025). **Modernização tecnológica de front-end: Uma proposta de interface de interação com usuário para o Classroom eXperience** [Trabalho de Conclusão de Curso, Universidade Federal de Uberlândia]. Repositório da Universidade Federal de Uberlândia. <https://repositorio.ufu.br/handle/123456789/46033>

Yáñez Flores, D. A., & Jaramillo Flores, P. D. C. (2025). **Comparativa de Frameworks de Desarrollo Backend: estudio de caso entre Spring y Struts**. *Revista Ingenio Global*, 4(2), 114–128. <https://doi.org/10.62943/rig.v4n2.2025.344>

Zhu, H., Zhang, Y., Zhao, B., Ding, J., Liu, S., Liu, T., Wang, D., Liu, Y., & Li, Z. (2025). **FrontendBench: A benchmark for evaluating LLMs on front-end development via automatic evaluation**. arXiv:2506.13832. <https://arxiv.org/abs/2506.13832>