

Hiago de Oliveira Mendes e Lucas Sales Salvo Petrucci

Uma Análise Comparativa entre *Client-Side Rendering* e *Server-Side Rendering* em Aplicações Web

Campos dos Goytacazes-RJ

Setembro de 2025

Hiago de Oliveira Mendes e Lucas Sales Salvo Petrucci

Uma Análise Comparativa entre *Client-Side Rendering* e *Server-Side Rendering* em Aplicações Web

Trabalho de Conclusão apresentado ao curso Bacharelado em Sistemas de Informação do Instituto Federal de Educação, Ciência e Tecnologia Fluminense, como parte dos requisitos para a obtenção do título de Bacharel em Sistemas de Informação.

Instituto Federal de Educação, Ciência e Tecnologia Fluminense

Orientador: Prof. D.Sc. Ronaldo Amaral Santos

Campos dos Goytacazes-RJ

Setembro de 2025

Hiago de Oliveira Mendes e Lucas Sales Salvo Petrucci

Uma Análise Comparativa entre *Client-Side Rendering* e *Server-Side Rendering* em Aplicações Web

Trabalho de Conclusão apresentado ao curso Bacharelado em Sistemas de Informação do Instituto Federal de Educação, Ciência e Tecnologia Fluminense, como parte dos requisitos para a obtenção do título de Bacharel em Sistemas de Informação.

Campos dos Goytacazes-RJ, 31 de Setembro de 2025.

Prof. D.Sc. Ronaldo Amaral Santos (orientador)
Instituto Federal Fluminense (IFF)

Prof. D.Sc. banca 1
Instituto Federal Fluminense (IFF)

Prof. D.Sc. banca 2
Instituto Federal Fluminense (IFF)

Campos dos Goytacazes-RJ
Setembro de 2025

Agradecimentos

agradecimentos

Resumo

Resumo aqui **Palavras-chaves:** SPA X MPA.

Abstract

abstract here

Keywords: Spa, Mpa.

Lista de Figuras

Figura 1 – Etapas de desenvolvimento da pesquisa	14
--	----

Lista de quadros

Lista de codigos

Siglas

CSR Client-Side Rendering

SEO Search Engine Optimization

SSR Server-Side Rendering

Sumário

1	INTRODUÇÃO	11
1.1	Problema e contexto	11
1.2	Justificativa	12
1.3	Objetivos	13
1.3.1	Objetivo Geral	13
1.3.2	Objetivos Específicos	13
1.4	Metodologia	13
1.5	Estrutura do Trabalho	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	S1	15
2.1.1	<i>Aggregate</i>	15
2.2	S2	15
3	TRABALHOS RELACIONADOS	16
4	ESTUDO DE CASO: REQUISITOS, ORGANIZAÇÃO E MÉTODOS	17
4.1	Contexto	17
5	ESTUDO DE CASO: DESIGN, IMPLEMENTAÇÃO E TESTES	18
5.1	Ciclo de Vida do Desenvolvimento de Software	18
6	RESULTADOS E DISCUSSÕES	19
6.1	Resultados	19
7	CONCLUSÃO	20
	REFERÊNCIAS	21

1 Introdução

1.1 Problema e contexto

O crescimento acelerado da web e o aumento da complexidade das aplicações modernas impuseram novos desafios ao desenvolvimento e à entrega de conteúdos na internet. Com o crescimento exponencial da web, estima-se que aproximadamente 252 mil novos sites sejam desenvolvidos diariamente, demonstrando não apenas a rapidez com que aplicações são criadas, mas também a necessidade crescente de estratégias eficientes para otimização de desempenho e escalabilidade ([SITEEFY, 2021](#)). A escolha da abordagem de renderização tornou-se um fator determinante para a experiência do usuário e a escalabilidade dos sistemas. Inicialmente, os sites eram compostos por páginas estáticas, cujo conteúdo era carregado diretamente do servidor. Com a evolução das tecnologias frontend, novas abordagens surgiram, destacando-se *Client-Side Rendering (CSR)* e *Server-Side Rendering (SSR)*. Cada uma dessas técnicas possui características específicas que influenciam diretamente o desempenho e a experiência do usuário.

A performance em websites é um fator determinante para o sucesso de qualquer aplicação web. O desempenho, frequentemente medido pelo tempo de carregamento das páginas, desempenha um papel fundamental na experiência do usuário e na taxa de conversão de visitantes ([WAGNER, 2016](#)). Uma página que carrega rapidamente proporciona uma navegação mais fluida, reduzindo a taxa de rejeição e aumentando a retenção de usuários. Além disso, o desempenho da página não se limita a impactar a experiência do usuário, mas também interfere diretamente no *Search Engine Optimization (SEO)*, tornando-se um critério essencial de indexação e ranqueamento em plataformas como o Google ([GOOGLE, 2010](#)).

Um exemplo notável de desafios enfrentados na escolha da estratégia de renderização ocorreu no *Twitter*. Em 2010, a empresa lançou uma nova versão de sua plataforma, conhecida como New Twitter, que utilizava extensivamente a renderização no lado do cliente (*CSR*) para aprimorar a interatividade e a experiência do usuário. No entanto, essa abordagem resultou em problemas significativos de desempenho, especialmente para usuários com conexões de internet mais lentas ou dispositivos menos potentes. Além disso, a dependência intensa de JavaScript dificultou a indexação de conteúdo pelos mecanismos de busca, impactando negativamente a otimização para motores de busca (*SEO*) ([TRADEOFFS... , 2015](#)). Reconhecendo essas limitações, o Twitter decidiu retornar à renderização no lado do servidor (*SSR*) em 2012, visando melhorar o desempenho e a acessibilidade de sua plataforma.

A arquitetura de frontend desempenha papel fundamental ao definir o fluxo de desenvolvimento e a escolha entre [CSR](#) e [SSR](#), sendo indispensável a adoção de um sistema modular e eficiente, capaz de ser mantido e escalado de forma sustentável ([GODBOLT, 2016](#)). Na abordagem [CSR](#), a renderização ocorre diretamente no navegador do usuário, reduzindo a carga no servidor, mas exigindo mais processamento no cliente; já na [SSR](#), o conteúdo é gerado no servidor antes de ser enviado ao cliente, o que proporciona carregamento mais rápido e melhor desempenho em dispositivos menos potentes. A decisão entre essas estratégias está diretamente ligada à performance da aplicação e deve considerar fatores como tempo de carregamento, complexidade da página e número de requisições HTTP ([WAGNER, 2016](#)), já que diferentes abordagens afetam não apenas a experiência do usuário, mas também os custos operacionais e a infraestrutura necessária para suportar a aplicação.

1.2 Justificativa

Nos últimos anos, observou-se um notável crescimento na adoção das abordagens de renderização no lado do cliente ([CSR](#)) e no lado do servidor ([SSR](#)) no desenvolvimento de aplicações web. Contudo, essas técnicas são frequentemente empregadas de maneira inadequada em muitos projetos, seja pela falta de entendimento de suas vantagens e limitações, seja por uma análise superficial das necessidades do produto ([EMADAMERHOTORI, 2024](#)). Um exemplo ilustrativo dessa realidade pode ser visto na experiência do *Airbnb*, que optou por uma abordagem de [SSR](#) com o intuito de melhorar o desempenho em dispositivos com recursos limitados e, sobretudo, otimizar a indexação de seu vasto catálogo de acomodações em mecanismos de busca ([NEARY, 2017](#)). Por outro lado, a equipe do *Instagram* enfrentou desafios ao equilibrar o carregamento dinâmico de conteúdo no cliente com a necessidade de garantir uma experiência fluida aos usuários, levando-os a adotar soluções híbridas que envolvem tanto [CSR](#) quanto [SSR](#) em diferentes partes da aplicação ([CONNER, 2019](#)).

Paralelamente a esses casos, identifica-se uma carência de estudos de caso reais que analisem de forma aprofundada o impacto da adoção de [CSR](#) e [SSR](#), principalmente no contexto nacional. Enquanto algumas publicações se concentram em apenas uma dessas abordagens, outras fornecem exemplos excessivamente simplificados, limitando a compreensão dos desafios técnicos e de negócios ao combinar essas estratégias em sistemas complexos.

Diante desse cenário, o presente trabalho visa suprir essa lacuna ao oferecer uma análise comparativa detalhada sobre a implementação de casos com [CSR](#) e [SSR](#), avaliando seus efeitos no desempenho, na experiência do usuário, segurança, otimização do [SEO](#) e na escalabilidade de aplicações web modernas. Por meio de um estudo de caso abrangente,

pretende-se não apenas enriquecer a literatura acadêmica, mas também fornecer insights práticos que possam orientar equipes de desenvolvimento e gestores na seleção e aplicação adequada dessas técnicas, contribuindo para a construção de sistemas mais robustos, eficientes e alinhados às demandas de um mercado em constante evolução.

1.3 Objetivos

1.3.1 Objetivo Geral

O objetivo geral deste trabalho de conclusão de curso é apresentar uma análise comparativa detalhada sobre a implementação de casos com [CSR](#) e [SSR](#), avaliando seus efeitos no desempenho, na experiência do usuário, segurança, otimização do [SEO](#) e na escalabilidade de aplicações web modernas.

1.3.2 Objetivos Específicos

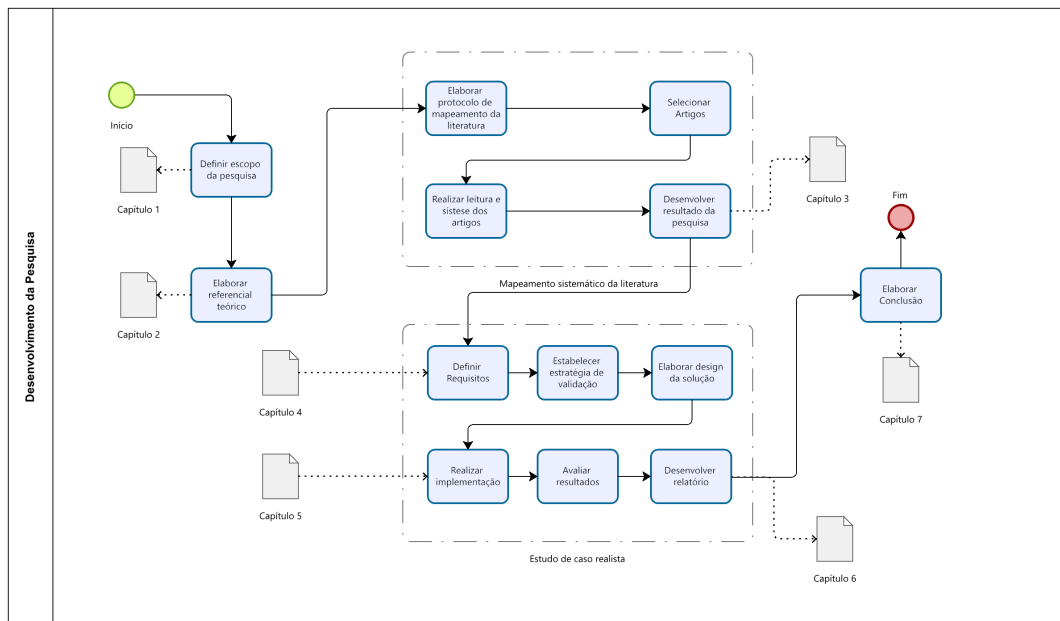
- Apresentar estratégias de escolhas entre [CSR](#) e [SSR](#), analisando métricas de desempenho, tempo de resposta, experiência do usuário e carga no servidor.
- Identificar as principais limitações e desafios enfrentados na escolha entre [CSR](#) e [SSR](#), considerando otimização de [SEO](#), escalabilidade e requisitos de infraestrutura.
- Apresentar recomendações práticas para desenvolvedores e gestores, auxiliando na tomada de decisão sobre qual abordagem utilizar com base nos objetivos do projeto e nas demandas do mercado.

1.4 Metodologia

Pode-se observar na [Figura 1](#) as etapas de execução dessa pesquisa. Inicialmente, o escopo é definido e o primeiro capítulo é elaborado. Nesse momento, busca-se apresentar o contexto do estudo, as justificativas e os objetivos a serem atingidos. Em seguida, a fundamentação teórica com os conceitos-chaves é construída, visando fornecer uma base sólida para o desenvolvimento do estudo de caso. Posteriormente, é realizado um mapeamento da literatura buscando trabalhos similares. Esse mapeamento é essencial para identificar lacunas no conhecimento e definir o escopo do estudo de caso. Além disso, ele permite a identificação de desafios, práticas e padrões comuns na utilização de [CSR](#) e [SSR](#). Por fim, um estudo de caso realista é desenvolvido, explorando a implementação prática de ambas as abordagens. Nesse estudo, são apresentados os requisitos, métodos e organização do sistema, além de trechos de código relevantes para demonstrar a aplicação das técnicas. Além disso, são realizados testes de desempenho para avaliar o impacto de

CSR e SSR em termos de tempo de resposta, carga no servidor, experiência do usuário e otimização para SEO. Finalmente, um relatório final é elaborado, contendo análises quantitativas e qualitativas dos resultados obtidos, além de uma discussão sobre trabalhos futuros, desafios e benefícios do uso de CSR e SSR em aplicações web modernas.

Figura 1 – Etapas de desenvolvimento da pesquisa



Fonte: os autores

1.5 Estrutura do Trabalho

Este trabalho está dividido em sete capítulos. O **Capítulo 1** expõe o contexto do estudo, as justificativas desta pesquisa e os objetivos a serem atingidos. O **Capítulo 2** apresenta conceitos fundamentais sobre CSR e SSR, abordando suas principais características, vantagens e desafios. O **Capítulo 3** expõe o protocolo e o resultado do mapeamento da literatura, analisando estudos relacionados e identificando lacunas no conhecimento sobre a adoção dessas abordagens. Da mesma forma, o **Capítulo 4** descreve os requisitos, métodos e organização do estudo de caso. Em seguida, o **Capítulo 5** apresenta o estudo de caso desenvolvido, incluindo o *design* do sistema e trechos de código chave da implementação. Posteriormente, o **Capítulo 6** apresenta os resultados obtidos com a execução dos testes de desempenho, analisando métricas como tempo de resposta, consumo de recursos, impacto no SEO e experiência do usuário. Por fim, o **Capítulo 7** apresenta as conclusões obtidas com o desenvolvimento deste trabalho, destacando os principais achados, desafios e recomendações para a escolha entre CSR e SSR em aplicações web modernas.

2 Fundamentação Teórica

2.1 *S1*

2.1.1 *Aggregate*

2.2 *S2*

3 Trabalhos Relacionados

4 Estudo de Caso: Requisitos, Organização e Métodos

Este capítulo apresenta o contexto, os requisitos, a organização e os métodos utilizados no desenvolvimento do estudo de caso.

4.1 Contexto

O estudo de caso é realizado em uma empresa fictícia.

5 Estudo de Caso: Design, Implementação e Testes

Este capítulo apresenta o desenvolvimento do estudo de caso.

5.1 Ciclo de Vida do Desenvolvimento de Software

6 Resultados e Discussões

Este capítulo apresenta os resultados obtidos com a execução dos testes de carga descritas.

6.1 Resultados

Essa seção apresenta uma análise gráfica dos resultados obtidos com a execução dos testes de carga.

7 Conclusão

Este trabalho apresentou um estudo de caso.

Referências

CONNER, G. Tornando o instagram.com mais rápido: Parte 2. *Engenharia do Instagram*, 2019. Disponível em: <<https://instagram-engineering.com/making-instagram-com-faster-part-2-f350c8fba0d4>>. Citado na página 12.

EMADAMERHO-ATORI, N. Client-side rendering (csr) vs. server-side rendering (ssr). *Prismic Blog*, 2024. Disponível em: <<https://prismic.io/blog/client-side-vs-server-side-rendering>>. Citado na página 12.

GODBOLT, M. *Frontend Architecture for Design Systems*. O'Reilly Media, Inc., 2016. ISBN 9781491926734. Disponível em: <<https://www.oreilly.com/library/view/frontend-architecture-for/9781491926772/>>. Citado na página 12.

GOOGLE. 2010. Acessado em: 14 de fevereiro de 2025. Disponível em: <<https://developers.google.com/search/blog/2010/04/using-site-speed-in-web-search-ranking>>. Citado na página 11.

NEARY, A. Rearchitecting airbnb's frontend. *Medium - Airbnb Engineering*, 2017. Disponível em: <<https://medium.com/airbnb-engineering/rearchitecting-airbnbs-frontend-5e213efc24d2>>. Citado na página 12.

SITEEFY. *How many websites are there?* 2021. Disponível em: <<https://siteefy.com/how-many-websites-are-there/>>. Citado na página 11.

TRADEOFFS in Server Side and Client Side Rendering. 2015. Acessado em: 14 de fevereiro de 2025. Disponível em: <<https://www.industrialempathy.com/posts/tradeoffs-in-server-side-and-client-side-rendering/>>. Citado na página 11.

WAGNER, J. L. *Web Performance in Action*. Manning Publications, 2016. ISBN 9781617293771. Disponível em: <https://www.manning.com/books/web-performance-in-action?a_aid=webopt&a_bid=63c31090>. Citado 2 vezes nas páginas 11 e 12.