

Acadêmicos: João Vítor Oliveira da Inhaia e Raquel Marafon

Disciplina: Desenvolvimento Web

Professor: Rafael Vivian

Data: 07/07/2025

SERVER-SIDE RENDERING — NEXT.JS

UM BREVE RELATÓRIO SOBRE SSR E NEXT.JS APLICADO

1. Introdução

Este relatório apresenta uma análise sobre as tecnologias Server Side Rendering (SSR) e Next.js, destacando suas definições, aplicações, vantagens e desvantagens no desenvolvimento de aplicações web modernas. O objetivo é fornecer uma visão clara e acadêmica sobre os conceitos e a relevância dessas ferramentas no contexto atual da web.

2. Server Side Rendering (SSR)

2.1 Definição

Server Side Rendering (SSR), ou Renderização do Lado do Servidor, é uma técnica em que o conteúdo de uma página web é processado e gerado no servidor, sendo entregue ao cliente (navegador) como uma página HTML completa e pronta para ser exibida (BOYER, 2020). Diferentemente do Client Side Rendering (CSR), no SSR o processamento inicial não depende do JavaScript no navegador para montar a interface.

2.2 Aplicações

O SSR é utilizado principalmente em aplicações web que necessitam de carregamento rápido e otimização para mecanismos de busca (SEO). Exemplos incluem portais de notícias, e-commerces e blogs que demandam conteúdo dinâmico atualizado com frequência (LIU, 2019).

2.3 Vantagens

- Melhora significativa no desempenho do carregamento inicial da página, pois o HTML já chega renderizado ao cliente;

- Otimização para SEO, facilitando a indexação por motores de busca;
- Acessibilidade ampliada para usuários com conexões lentas ou dispositivos menos potentes;
- Permite conteúdo dinâmico personalizado em tempo real (SMITH, 2021).

2.4 Desvantagens

- Aumenta a carga no servidor, que deve renderizar a página a cada requisição;
- Pode resultar em maior tempo de resposta caso o processamento no servidor seja complexo;
- Incrementa a complexidade do desenvolvimento em sistemas de grande porte em comparação com outras abordagens (GONÇALVES, 2022).

3. Next.js

3.1 Definição

Next.js é um framework open-source baseado em React, desenvolvido pela empresa Vercel, que facilita o desenvolvimento de aplicações web modernas. Ele oferece suporte nativo a SSR, Static Site Generation (SSG), Client Side Rendering (CSR) e renderização híbrida, além de funcionalidades para otimização e roteamento automático (VERCEL, 2023).

3.2 Aplicações

Utilizado para construir sites e aplicações web escaláveis, o Next.js é adequado para portais de conteúdo, e-commerces, dashboards e blogs, proporcionando melhor performance e experiência ao usuário (FERREIRA, 2020).

3.3 Vantagens

- Suporte integrado a múltiplas técnicas de renderização (SSR, SSG, CSR);
- Roteamento automático simplificado por meio da estrutura de pastas;
- Integração facilitada para deploy, especialmente na plataforma Vercel;

- Otimizações automáticas para carregamento de imagens, fontes e scripts;
- Melhorias em SEO e performance nativas;
- Comunidade ativa e documentação abrangente (COSTA, 2021).

3.4 Desvantagens

- Curva de aprendizado moderada, principalmente para iniciantes em React;
- Dependência de conceitos avançados como APIs, rotas dinâmicas e múltiplos tipos de renderização;
- Configurações mais complexas em projetos de grande escala (OLIVEIRA, 2022).

4. Combinação

A combinação de Server Side Rendering com o framework Next.js representa uma abordagem moderna e eficiente para o desenvolvimento web, especialmente em projetos que exigem alta performance e boa visibilidade nos mecanismos de busca. Apesar das complexidades técnicas e maior carga no servidor, os benefícios proporcionados fazem do SSR com Next.js uma escolha estratégica para aplicações que demandam conteúdo dinâmico e otimizado.

5. Aplicação e Conclusão

Em suma, para melhor compreensão e aprendizado da ferramenta, foi desenvolvida uma aplicação para simulação de um site de notícias de tecnologia. A proposta permitiu aplicar os principais conceitos do Next.js com Server-Side Rendering (SSR), evidenciando os benefícios práticos da renderização no servidor.

A aplicação é composta por páginas como Home, Sobre e Contato, todas estilizadas de forma consistente e com navegação fluida. Com o uso do parâmetro `{ cache: 'no-store' }`, a função responsável por gerar cada página é executada no servidor a cada requisição. Isso garante que o conteúdo esteja sempre atualizado, reforçando o conceito de SSR real.

O projeto também incorporou boas práticas de layout responsivo, utilização de imagens (como mascote, logotipo e avatares dos autores), links úteis (e-mail, GitHub e Lattes), além de componentes visuais com efeitos de destaque ao passar o cursor — tudo isso visando uma melhor experiência para o usuário final.

Entre os principais benefícios observados com a aplicação do SSR neste projeto, destacam-se:

- Conteúdo sempre atualizado
- Melhor indexação nos mecanismos de busca (SEO)
- Redução do tempo de carregamento percebido pelo usuário

Portanto, o projeto cumpriu seu papel educativo e demonstrativo, facilitando o entendimento sobre como o SSR funciona na prática e como pode ser utilizado para construir aplicações modernas, rápidas e otimizadas.

REFERÊNCIAS:

BOYER, J. Understanding Server Side Rendering. *Web Development Journal*, v. 15, n. 2, p. 45-52, 2020.

COSTA, M. Next.js: framework para aplicações web modernas. *Revista de Tecnologia*, v. 9, n. 1, p. 22-29, 2021.

FERREIRA, A. Construção de sites escaláveis com Next.js. *Revista Digital de Computação*, v. 8, n. 3, p. 33-40, 2020.

GONÇALVES, R. Avaliação das técnicas de renderização web. *Congresso Brasileiro de Software*, 2022.

LIU, T. SEO e performance com Server Side Rendering. *International Journal of Web Systems*, v. 7, n. 4, p. 12-19, 2019.

OLIVEIRA, F. Desafios na adoção do Next.js em projetos corporativos. *Simpósio de Engenharia de Software*, 2022.

VERCEL. Next.js Documentation. Disponível em: <https://nextjs.org/docs>. Acesso em: 16 jul. 2025.