

Cronograma de estudos de NextJS

feito único e exclusivamente para a Ju ♥

Primeiro de tudo:

- ☐ **Ter uma boa base de ReactJS**
- ☐ [React Foundations | Next.js \(nextjs.org\)](#)

Depois que ver o link acima (só pra revisar), agora pode prosseguir.

1. Entender como funciona o NextJS (Importantíssimo)

- ☐ [Learn Next.js | Next.js \(nextjs.org\)](#)

O link acima é da própria documentação do NextJS atualizadíssima que te ensina, desde o início até o fim, a como criar um dashboard. Isso mesmo.

Entender o que diabo é SSR, CSR, SSG... Vai ser bom.

2. Roteamento:

- ☐ Entenda como o roteamento funciona no Next.js.
- ☐ [Building Your Application: Routing | Next.js \(nextjs.org\)](#)

3. Layout e Páginas:

- ☐ Explore como criar layouts e páginas no Next.js.
- ☐ [Routing: Pages and Layouts | Next.js \(nextjs.org\)](#)

4. Link e Navegação:

- ☐ Aprenda a usar o componente Link para facilitar a navegação entre páginas.
- ☐ [Components: <Link> | Next.js \(nextjs.org\)](#)

5. Route Groups:

- ☐ Compreenda como organizar rotas em grupos para uma melhor estrutura.
- ☐ [Routing: Route Groups | Next.js \(nextjs.org\)](#)

6. Rotas Dinâmicas:

- ☐ Implemente rotas dinâmicas para lidar com diferentes tipos de conteúdo.

7. **Suspense** (que é o componente que você tava tendo problema, pois é um componente que renderiza dados no lado do cliente):

- ☐ Como criar interfaces de carregamento.
- ☐ O Suspense é uma funcionalidade introduzida no React para facilitar o tratamento de suspensões de renderização enquanto aguarda a conclusão de operações assíncronas, como a busca de dados. No contexto do Next.js, o Suspense pode ser usado para melhorar a experiência do usuário ao carregar dados dinâmicos durante a navegação entre páginas.
- ☐ Ao usar o Suspense no Next.js, geralmente estamos nos referindo à capacidade de carregar dados de maneira assíncrona e aguardar a conclusão dessa operação antes de renderizar a página. Isso é especialmente útil quando você está trabalhando com rotas dinâmicas e precisa buscar dados específicos para uma página antes de renderizá-la.
- ☐ *<React.Suspense>*:
Em componentes React, você pode usar `<React.Suspense>` para envolver partes da árvore de componentes onde ocorrem operações assíncronas. Isso informa ao React para "suspender" a renderização dessa parte até que a operação assíncrona seja concluída.
- ☐ `useEffect` e `fetch` assíncrono
Em páginas Next.js, você pode usar o hook `useEffect` para realizar operações assíncronas, como buscar dados. O uso de `fetch` é comum para essa finalidade.
- ☐ `getServerSideProps` e `getStaticProps`
Além do Suspense, o Next.js oferece métodos específicos, como `getServerSideProps` e `getStaticProps`, para buscar dados durante a renderização do servidor ou para gerar páginas estáticas com antecedência.

8. **Rotas Paralelas:**

- ☐ Explore o conceito de rotas paralelas e como implementá-las.
- ☐ [Routing: Parallel Routes | Next.js \(nextjs.org\)](https://nextjs.org/docs/routing/parallel-routes)

9. **API com Route Handlers:**

- ☐ Descubra como criar APIs usando os handlers de rotas no Next.js.