

Introdução ao **NextJS**

O que é, qual diferença e quando usar?

// O NextJS é um **Framework Web**
desenvolvido em **ReactJS**
lançado em 2016 por Guillermo Rauch.

“Um **Framework Web** é um sistema opinativo com **estrutura** e **ferramentas** já definidas.

// Por que usar um **Framework Web**?

NEXT.js

// Webpack? Rollup? E o Parcel? Como faz com as **rotas**?
E como **gerar estático**? E para **produção**?



$$A = \pi r^2$$

$$C = 2\pi r$$

	30°	45°	60°
sin	$\frac{1}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{3}}{2}$
cos	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{1}{2}$
tan	$\frac{\sqrt{3}}{3}$	1	$\sqrt{3}$



$$\int \sin x dx = -\cos x + C$$

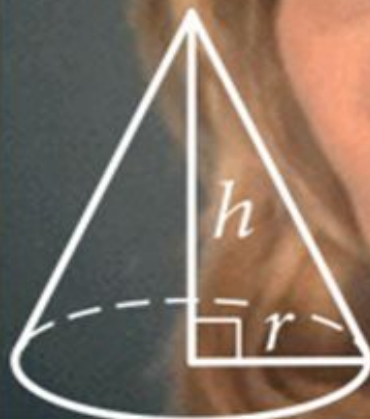
$$\int \frac{dx}{\cos^2 x} = \tan x + C$$

$$\int \tan x dx = -\ln|\cos x| + C$$

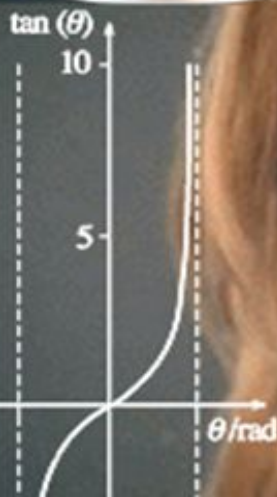
$$\int \frac{dx}{\sin x} = \ln\left|\tan \frac{x}{2}\right| + C$$

$$\int \frac{dx}{a^2 + x^2} = \frac{1}{a} \arctan \frac{x}{a} + C$$

$$V = \frac{1}{3} \pi r^2 h$$



$$V = \pi r^2 h$$



$$ax^2 + bx + c = 0$$

$$a\left(x^2 + \frac{b}{a}x + \frac{c}{a}\right) = 0$$

$$x^2 + 2\frac{b}{a}x + \left(\frac{b}{a}\right)^2 - \left(\frac{b}{a}\right)^2 + \frac{c}{a} = 0$$

O que o **NextJS** tem/faz?

- Renderização no servidor (Server Side Rendering - SSR)
- Geração de estáticos (Static Site Generation - SSG)
- CSS-in-JS (Styled-jsx, Styled Components, Emotion, etc)
- Zero Configuration (rotas, hot reloading, code splitting...)
- Completamente extensível (controle completo do Babel/Webpack, plugins, etc)
- Otimizado para produção

Mas o mercado de trabalho usa?

NETFLIX

GitHub

scale

ticketmaster®

twitch

coinbase

Uber

VISION

trovit

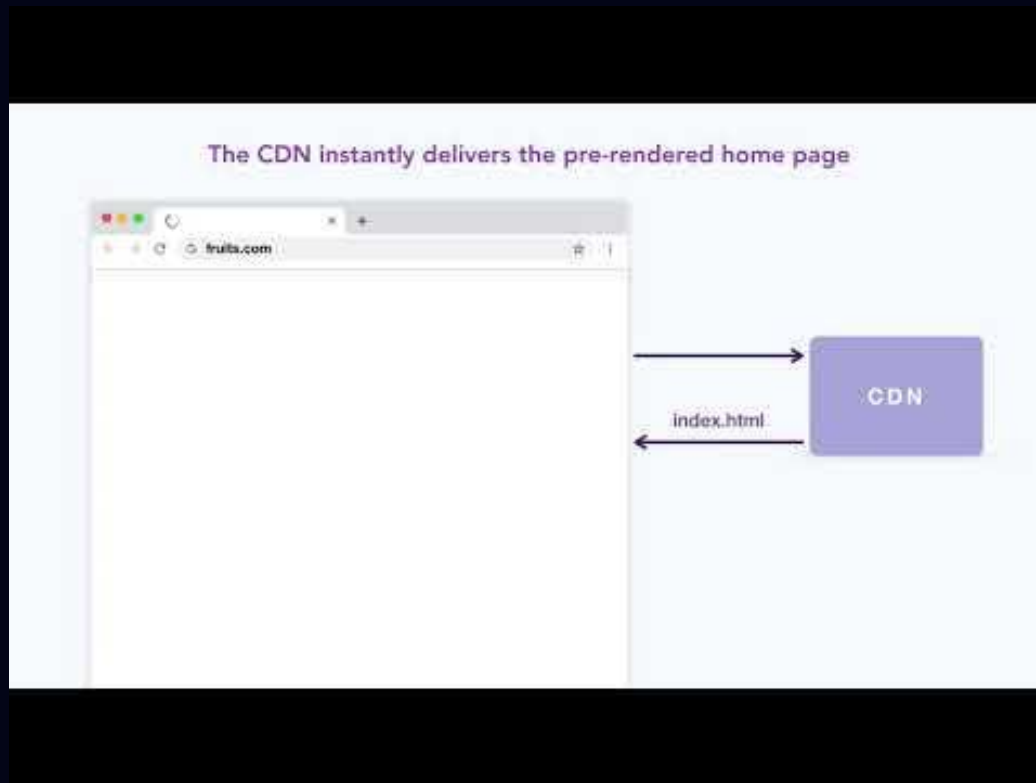
BINANCE

“E quais as diferenças para o **Gatsby**, **Create React App**, insira outro framework React aqui.

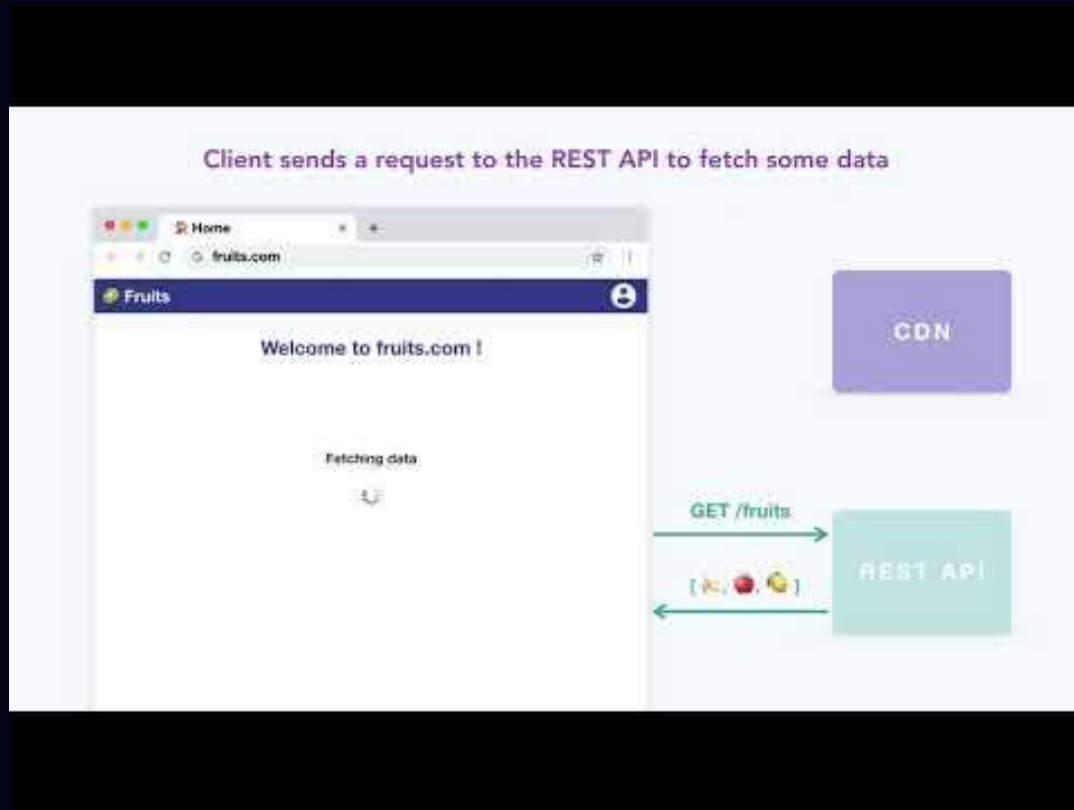
Tipos de uma aplicação

- Static Site (HTML/CSS/JS) – GatsbyJS, Hexo
- Client Side Rendering (Single Page Application – SPA) – Create React App
- Server Side Rendering (SSR) – NextJS

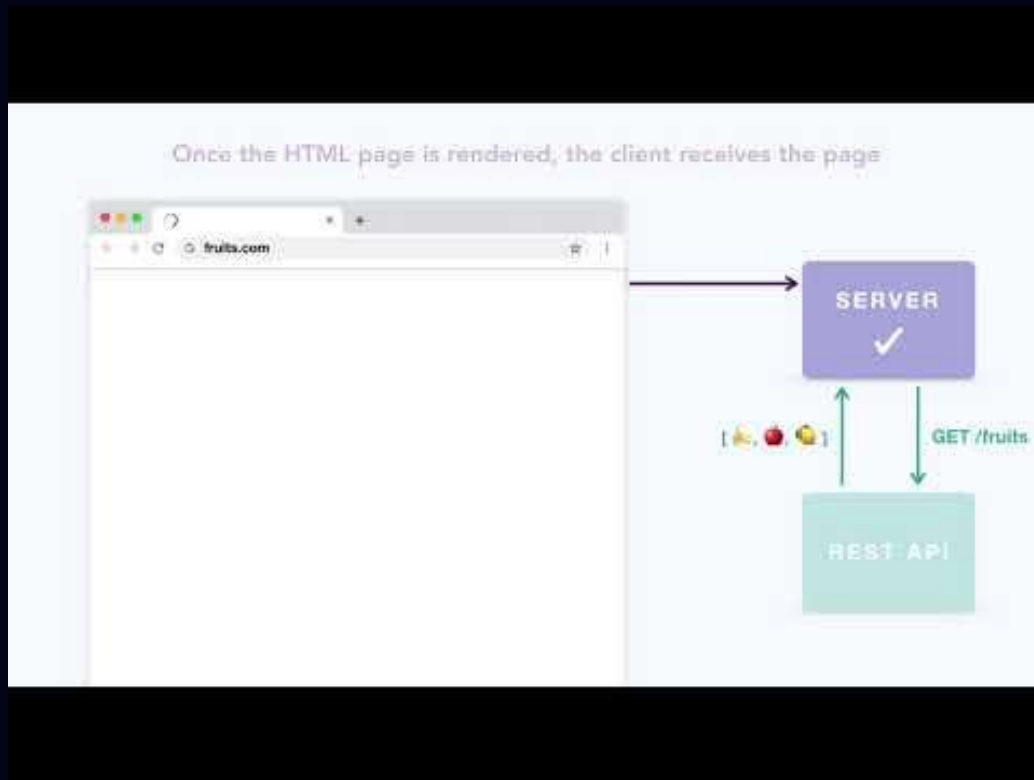
Static Site



Client Side Rendering (SPA)



Server Side Rendering (SSR)



Static Site Generation

Vantagens

- Uso quase nulo do servidor
- Pode ser servido numa CDN (melhor cache)
- Melhor performance entre todos
- Flexibilidade para usar qualquer servidor

Desvantagens

- Tempo de build pode ser muito alto
- Dificuldade para escalar em aplicações grandes
- Dificuldade para realizar atualizações constantes

Single Page Application (SPA)

Vantagens

- Permite páginas ricas em interações sem recarregar
- Site rápido após o load inicial
- Ótimo para aplicações web
- Possui diversas bibliotecas

Desvantagens

- Load inicial pode ser muito grande
- Performance imprevisível
- Dificuldades no SEO
- A maioria do conteúdo não é indexado

Server Side Rendering (SSR)

Vantagens

- Ótimo em SEO
- Meta tags com previews mais adequados
- Melhor performance para o usuário (o conteúdo vai ser visto mais rápido)
- Código compartilhado com o backend em Node
- Menor processamento no lado do usuário

Desvantagens

- TTFB (Time to first byte) maior, o servidor vai preparar todo o conteúdo para entregar
- HTML maior
- Reload completo nas mudanças de rota*

//

E o **NextJS** tem suporte para SSG, SSR e até SPA!

//

Quando usar cada um então?

Static Site Generation (Gatsby, NextJS)

- Site simples sem muita interação do usuário
- Se você é a única pessoa que publica conteúdo
- Se o conteúdo muda pouco
- Se o site é simples, sem tantas páginas
- Quando a performance é extremamente importante

Exemplos: Landing Page, Blogs, Portfólios

Single Page Application (SPA) – CRA

- Quando não tem tanta necessidade de indexar informações no Google
- Quando o usuário faz muitas interações na página e não quero refreshes
- Quando as informações vão ser diferentes para cada usuário (autenticação, por exemplo)

Exemplos: Twitter Web, Facebook Web, Spotify Web

Server Side Rendering (SSR) – NextJS

- Quando tem necessidades de um SPA, mas precisa de um loading mais rápido
- Quando o conteúdo muda frequentemente
- Quando trabalha com um número muito grande de páginas
- Quando precisa de uma boa indexação no Google

Exemplos: Ecommerce, Sites de Notícias

Mais links para você estudar!

- <https://nextjs.org/learn/basics/create-nextjs-app> (Tutorial oficial passo-a-passo)
- <https://dev.to/kefranabg/demystifying-ssr-csr-universal-and-static-rendering-with-animations-m7d> (animações vistas nos slides)
- <https://marinaaisa.com/blog/cook-websites-based-on-your-needs/> (explicações sobre quando usar um ou outro)
- <https://github.com/vercel/next.js/tree/canary/examples> (vários exemplos oficiais do NextJS com outras tecnologias)