

React JS

★ AULA 06 ★

★ PROFESSOR
MAROMO ★

React JS

Next.js

Introdução ao Next.js



Agenda

- Criar um Projeto em Next.js
- Fazer o Deploy
- Hospedar gratuitamente na Vercel



Next.js

Next.js é uma estrutura de JavaScript para desenvolvimento de aplicações web baseada em React. Foi criada pela Vercel e está disponível como um projeto de código aberto.

O Next.js oferece uma série de recursos que ajudam a otimizar o processo de desenvolvimento e a melhorar o desempenho da aplicação



Recursos do Next.js

1. Renderização do lado do servidor (SSR): Isso permite que as aplicações sejam renderizadas no servidor antes de serem enviadas ao navegador, o que pode melhorar o desempenho e a otimização para motores de busca (SEO).



Recursos do Next.js

2. Geração de site estático (SSG): O Next.js permite que você gere sites estáticos a partir de seus aplicativos React. Isso é particularmente útil para sites com conteúdo que não muda frequentemente, já que permite que a página seja gerada uma vez e servida a cada pedido, resultando em tempos de carregamento de página mais rápidos.

3. Roteamento baseado em sistema de arquivos: O Next.js segue a convenção sobre a configuração para as rotas, onde os arquivos na pasta "pages" se tornam rotas automaticamente.



Recursos do Next.js

4. Suporte para TypeScript: O Next.js tem suporte integrado para TypeScript, uma variante tipada do JavaScript.

5. API Routes: Permite a criação de rotas de API REST ou API endpoints dentro do mesmo projeto Next.js, facilitando a construção de um back-end junto com o seu front-end

6. Hot Code Reloading: Recarrega automaticamente a página sempre que você faz uma alteração no código.



Algumas empresas que usam o Next.js

- Netflix,
- Github,
- Starbucks,
- Docker,
- Twitch,
- Uber,
- Airbnb,
- TikTok,
- Nike, etc...



O que é
Deploy ?





Deploy

"Deploy" é um termo amplamente usado na área de desenvolvimento de software para se referir ao processo de fazer um aplicativo ou um serviço estar disponível para uso.

Em termos mais práticos, o deploy pode ser visto como a série de etapas que são executadas para mover o código do software de um ambiente de desenvolvimento ou teste para um ambiente de produção, onde o software pode ser acessado e utilizado pelos usuários finais.

Atividades

Compilação: Em alguns casos, o código fonte do aplicativo é compilado para criar um executável ou outros artefatos que podem ser executados em um servidor ou outra infraestrutura.

Empacotamento: Os artefatos de software criados são frequentemente agrupados em um pacote que pode ser facilmente movido entre ambientes.

Transporte: O pacote de software é então movido do ambiente de desenvolvimento ou teste para o ambiente de produção.

Instalação: O pacote de software é instalado no ambiente de produção, o que pode envolver a extração dos artefatos do pacote e a realização de outras tarefas de configuração.

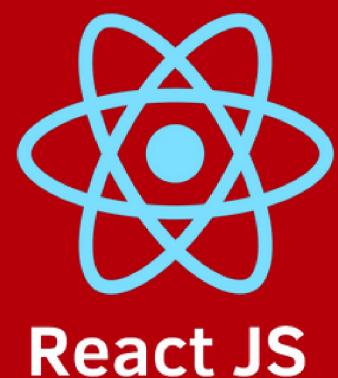
Configuração: Quaisquer configurações necessárias para o ambiente de produção são aplicadas ao software. Isso pode incluir a configuração de conexões de banco de dados, chaves de API, variáveis de ambiente, etc.

Ativação: O software é então iniciado e começa a aceitar solicitações de usuários ou outros sistemas.

Laboratório



Desenvolvimento de uma aplicação SPA usando o React JS. Nome da aplicação **sample06..**



Primeiro passo:

Criar um repositório no github, chamado sample06

Owner * Repository name *

 maromo71 / sample06

 sample06 is available.

Great repository names are short and memorable. Need inspiration? How about [animated-doodle](#)?

Description (optional)

 Public
Anyone on the internet can see this repository. You choose who can commit.

 Private
You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file
This is where you can write a long description for your project. [Learn more about READMEs](#).

Add .gitignore

.gitignore template: Node

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

Choose a license

License: None



Passo 1: Configuração do ambiente de desenvolvimento

```
6 pasta(s)   160.417.779.712 bytes disponíveis
```

```
C:\javascript\react01>git clone https://github.com/maromo71/sample06
```

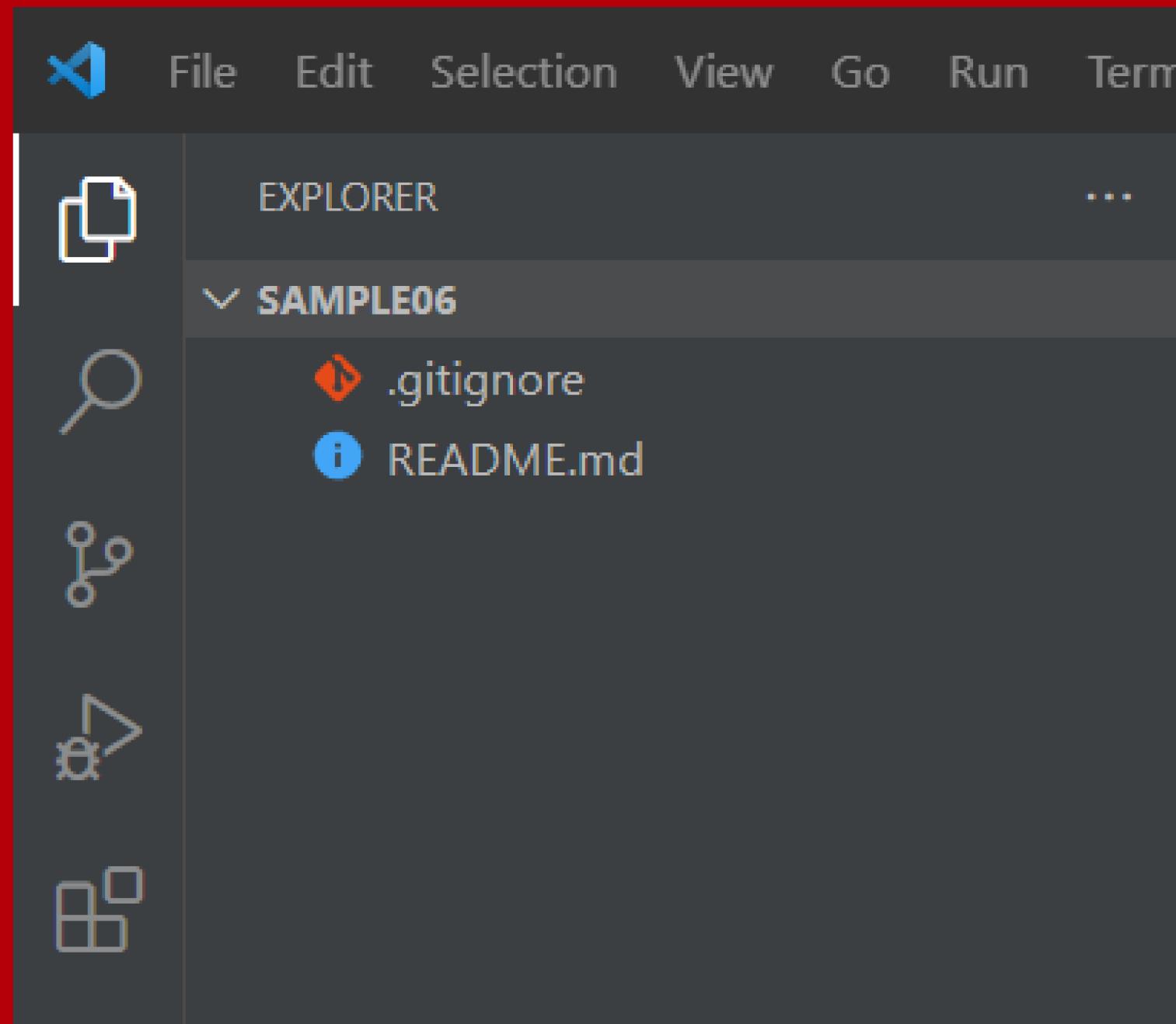
```
Receiving objects: 100% (77/77), done.
```

```
C:\javascript\react01>cd sample06
```

```
C:\javascript\react01\sample06>code .
```



Passo 2: Estrutura do projeto clonado



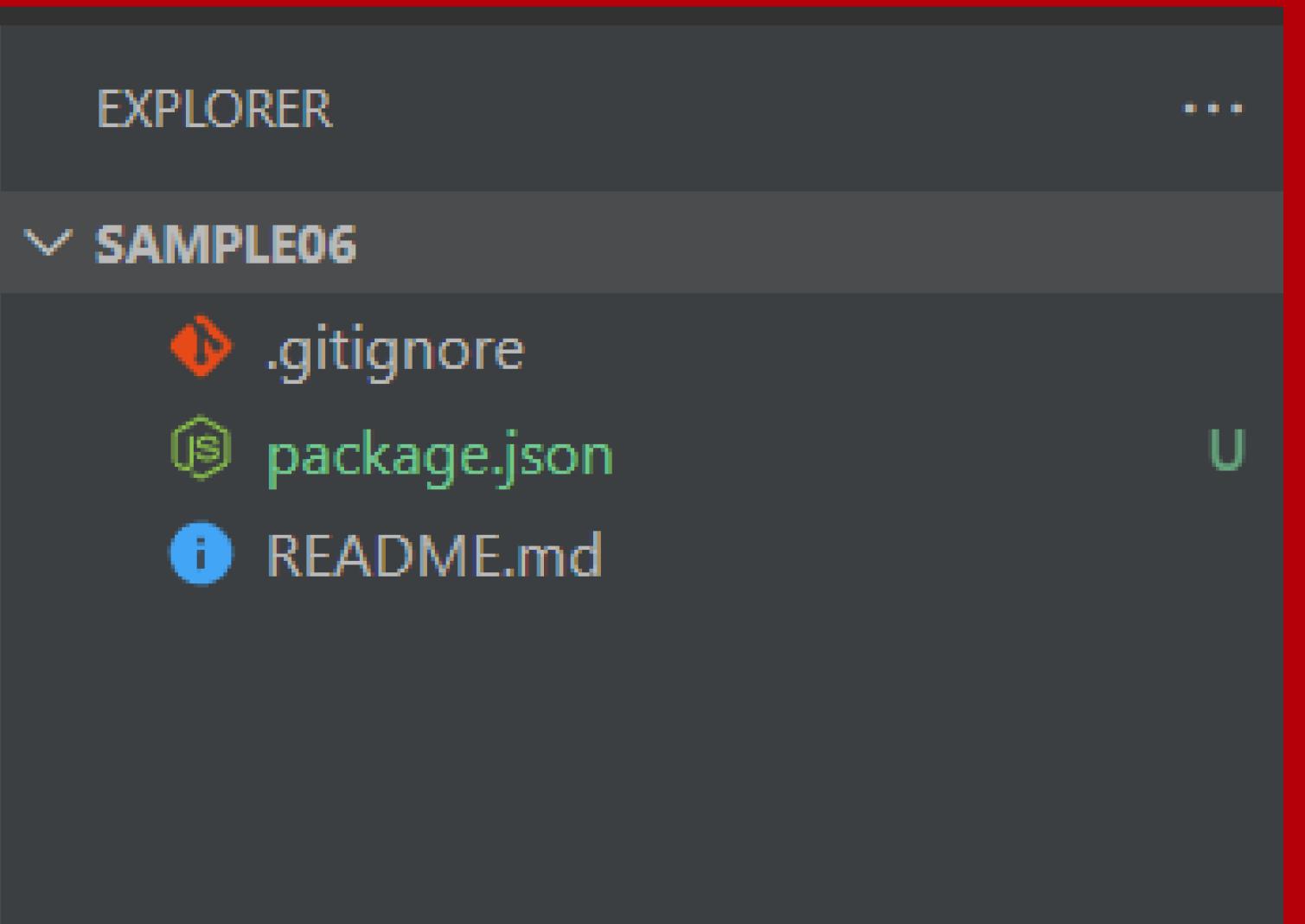
Neste momento precisamos ter instalado na máquina:

- Node.js
- Next.js
- React.js

Passo 3

No diretório da aplicação, no terminal do VS CODE, digite:

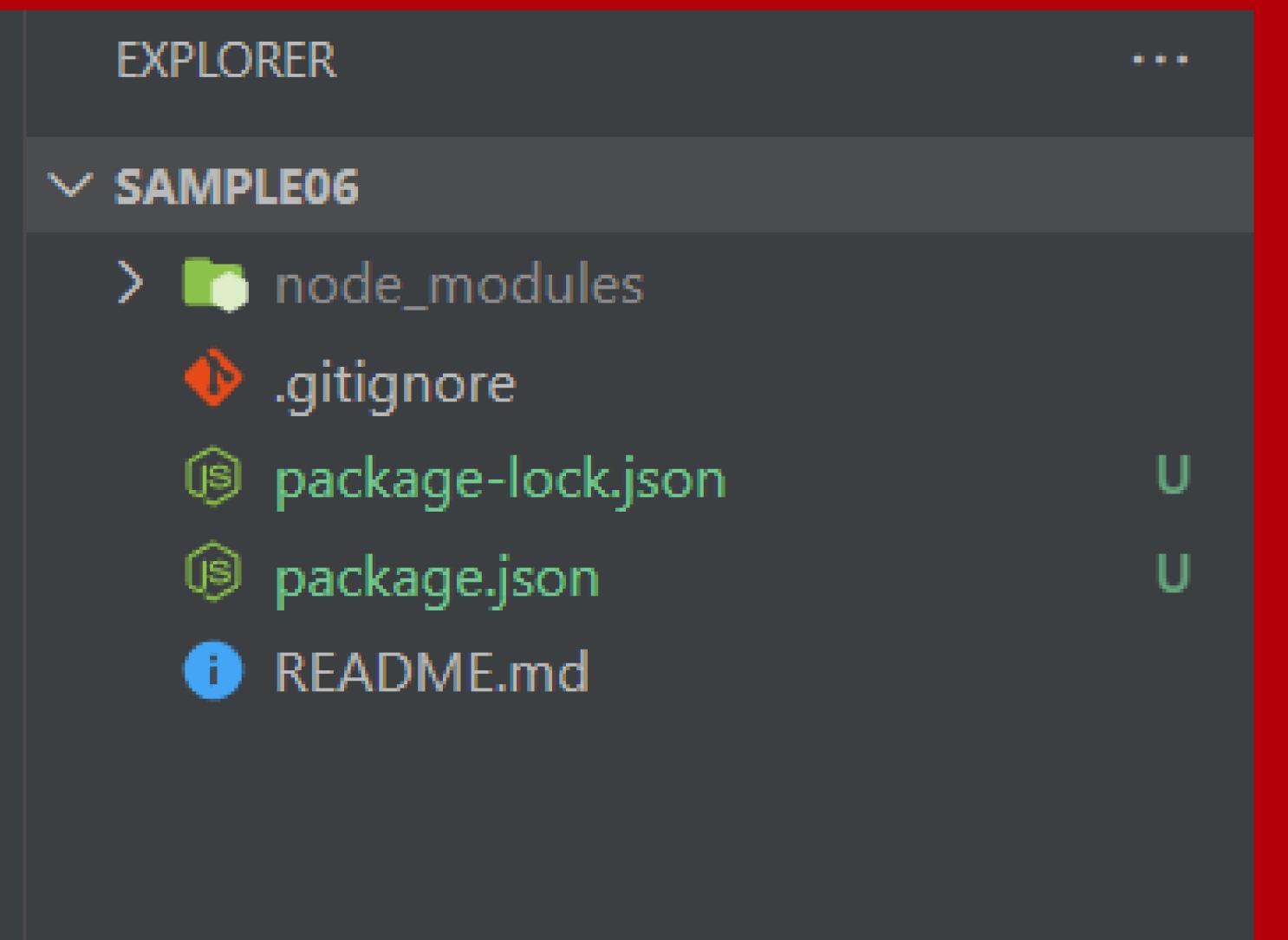
npm init -y



Passo 4

Para instalar a dependência do Next.js, digite o comando:

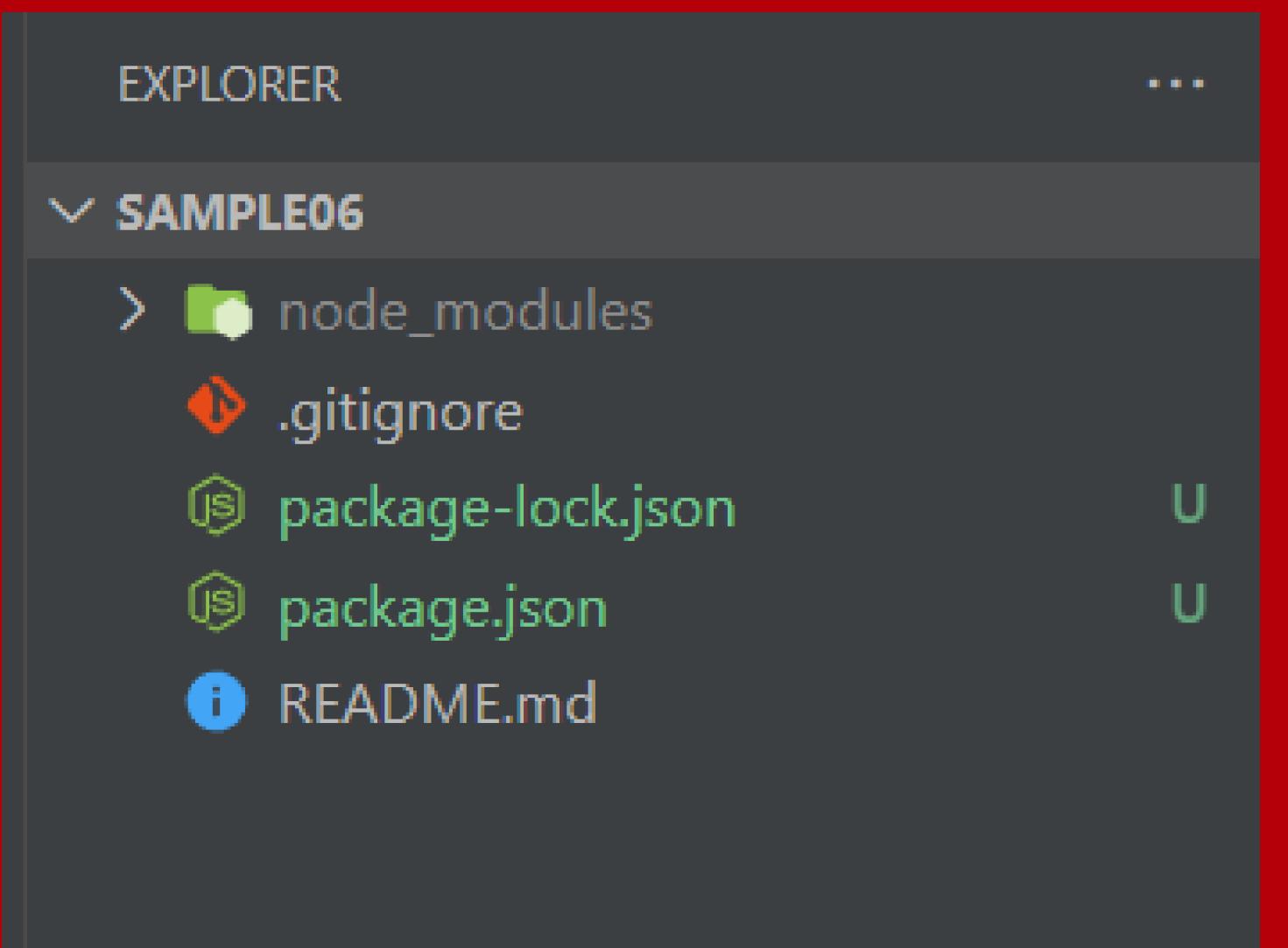
npm install next



Passo 5

Para instalar a dependência do React, digite o comando:

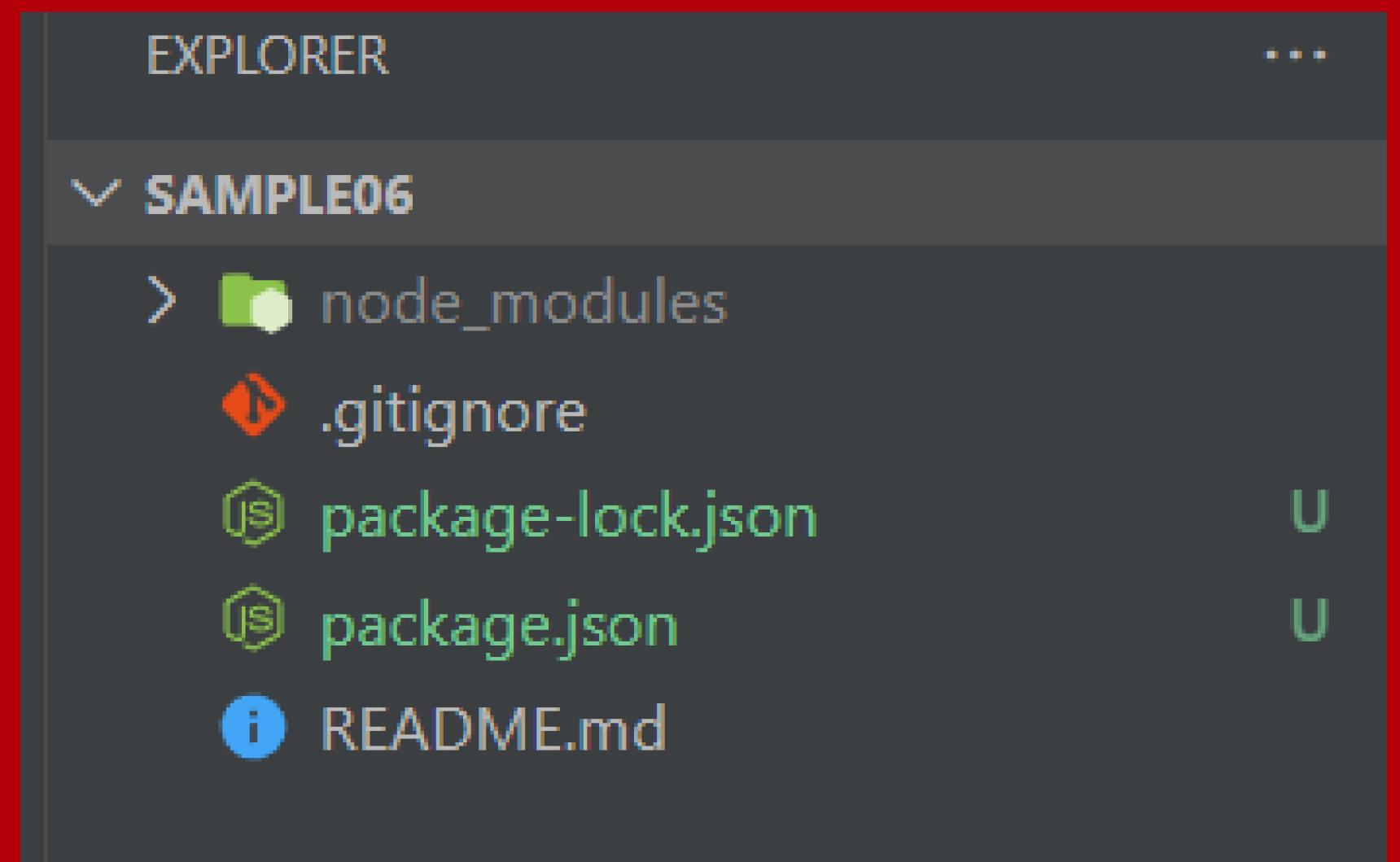
npm install react



Passo 6

Para instalar a dependência do React-dom, digite o comando:

npm install react-dom

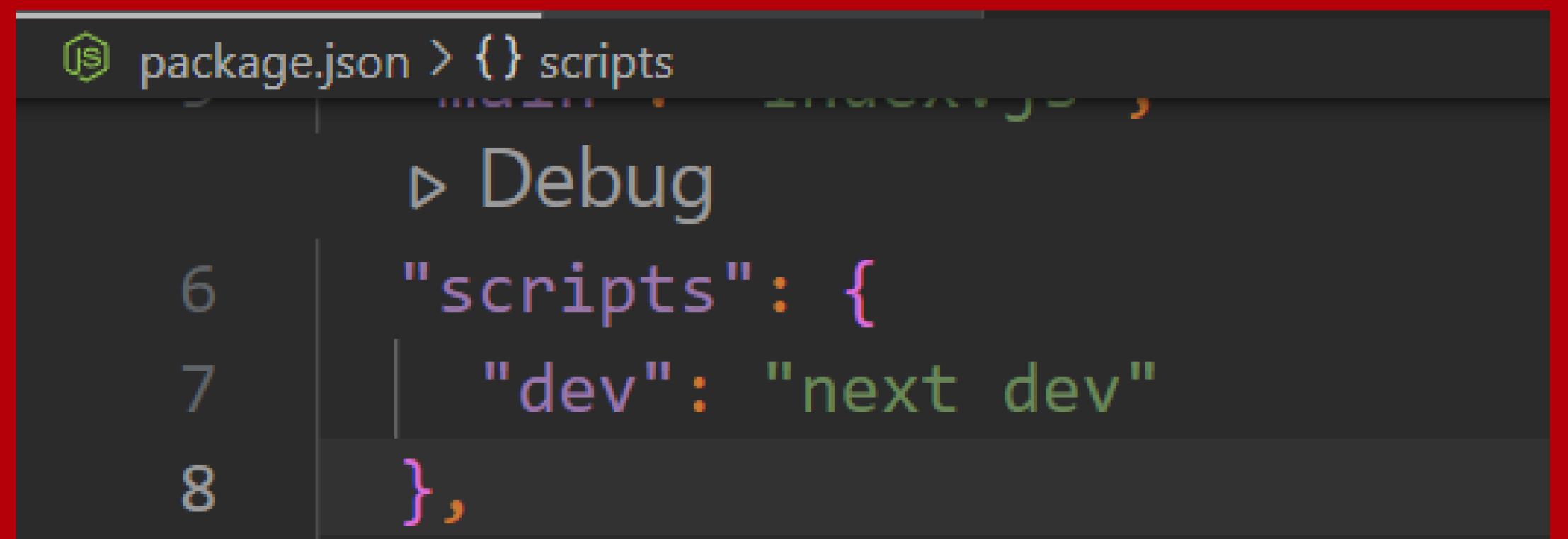


Passo 7 - Criando nossa página estática



No package.json:

- Alterando a linha de script, incluindo a **dev**, que simplesmente vai rodar o comando "next dev"



```
package.json > {} scripts
▶ Debug
6   "scripts": {
7     "dev": "next dev"
8   },
```

Nota



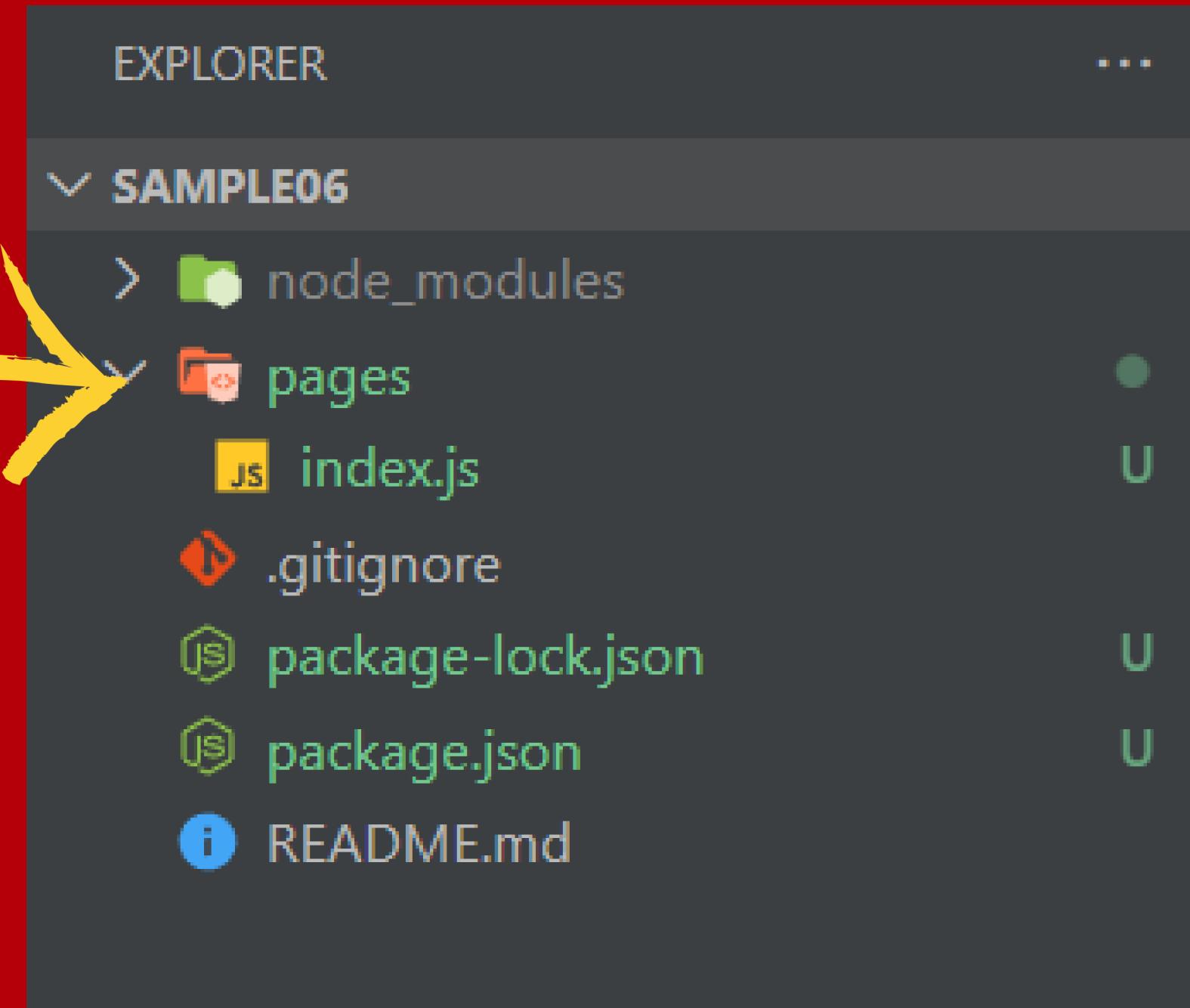
No Next.js

Você não precisa configurar as rotas do seu projeto, porque para cada arquivo dentro da pasta pages (que criaremos) virá automaticamente uma rota e uma página.



Passo 8 - Criando a pasta Pages

Criando a pasta pages e dentro dela o arquivo index.js.





Componente Home

Dentro do arquivo index.js, vamos criar nosso componente React que representa a página home da nossa aplicação. Como dentro de um arquivo Javascript podemos ter vários componentes React, várias funções tradicionais, então como o Next.js vai saber qual componente/função representa a nossa página ?

- Simples. Basta usar o novo padrão de módulo do Javascript e exportar por default o componente para que seja renderizado como página.



Passo 9: Componente: Home

```
pages > js index.js > [?] default
1 function Home(){
2   return <div>
3     |   Home
4   </div>
5 }
6 export default Home;
```

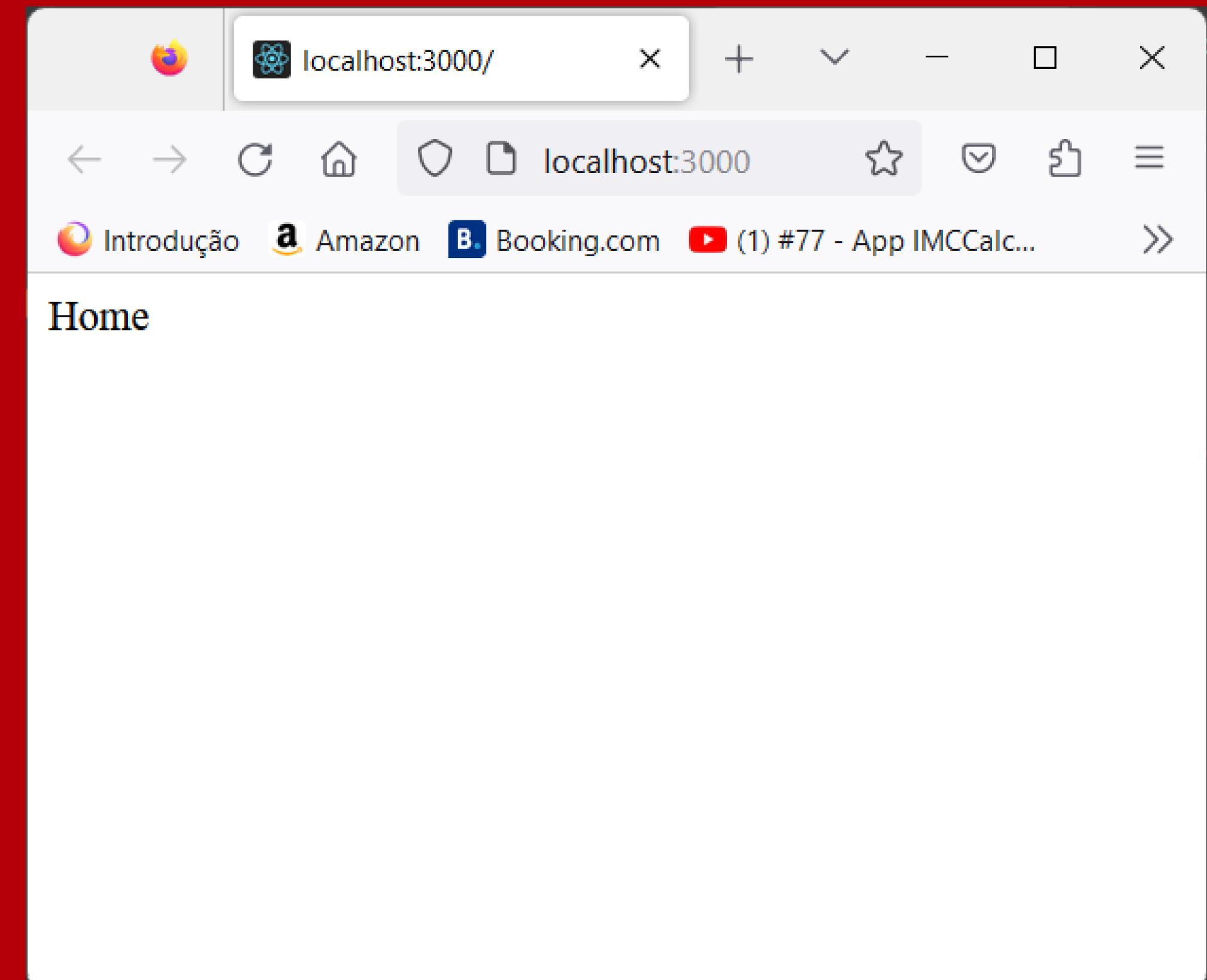
Rodando Localmente



Para isso, basta executar o comando:

npm run dev

No terminal de comandos.



Home

Vamos alterar o nosso componente **Home** para que ele funcione como um contador, que ao clicar em um botão chamado Adicionar, execute alteração no estado da aplicação.

Neste exemplo, sempre que se clicar no botão adicionar ele vai manter o estado do componente, de maneira que ao executar a função **adicionarContador** ele incrementa um na variável contador.

Você pode fazer ajustes no código e verificar as alterações sem perder o estado anterior.

Experimente acrescentar uma tag H3 para alterar o título da página e salve. Ele alterará a div que contém a palavra Home e mantém o valor do contador.

Home

```
1 import { useState } from "react";
2 
3 function Home(){
4     return <div>
5         Home
6         <Contador />
7     </div>
8 }
9 
10 function Contador(){
11     const [contador, setContador] = useState(1);
12     function adicionarContador(){
13         setContador(contador + 1);
14     }
15     return(
16         <div>
17             <div>
18                 {contador}
19             </div>
20             <button onClick={adicionarContador}>Adicionar</button>
21         </div>
22     )
23 }
```

Pronto!!!

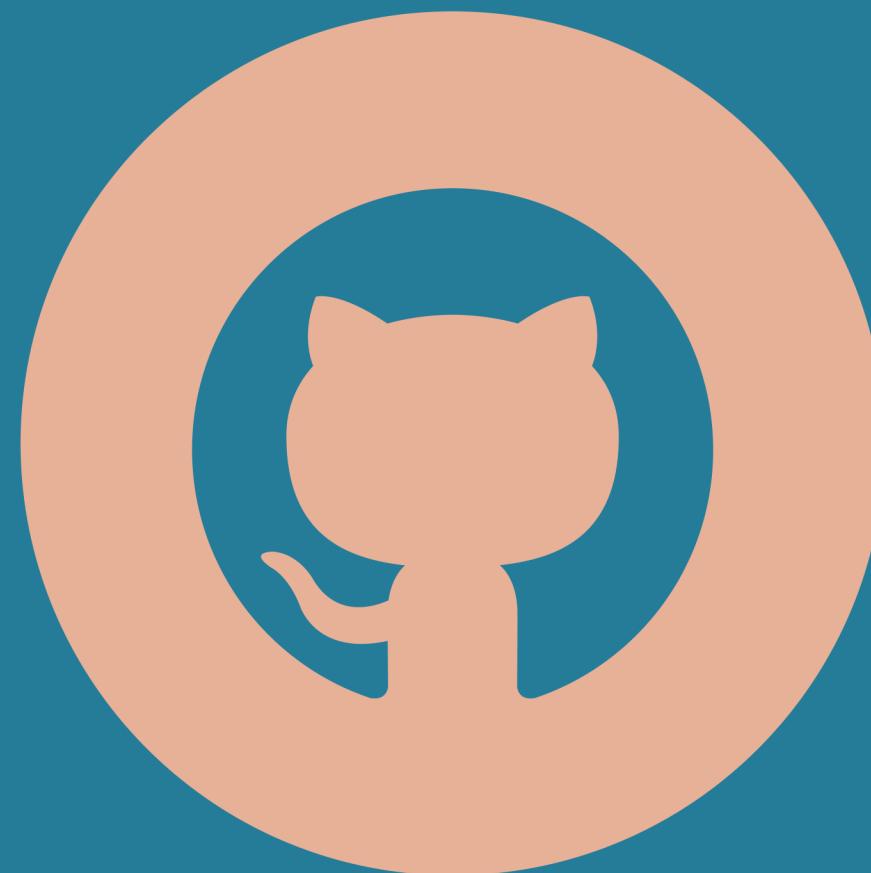
Com essa página estática criada, que tal colocarmos ela no ambiente de produção. Apesar de simples, podemos armazená-la na nuvem em um endereço de verdade na net que você poderá acessar de qualquer lugar.

A primeira coisa a se fazer é parar o servidor de desenvolvimento.
(CTRL + S)

Agora vamos colocar as alterações na branch principal do github.

*Vamos colocar na
nuvem*

GITHUB



Colocando na branch principal



Sequencia de comandos no terminal:

git add -A

git commit -m "Primeiro site na nuvem"

git push origin main



Repositório no Github

maromo71 / sample06 Public

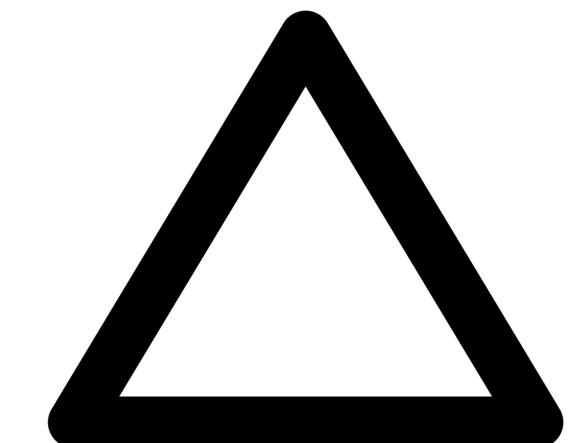
<> Code Issues Pull requests Actions Projects Security Insights

main 1 branch 0 tags

File	Description
maromo71 Primeiro site na nuvem	
pages	Primeiro site na nuvem
.gitignore	Initial commit
README.md	Initial commit
package-lock.json	Primeiro site na nuvem
package.json	Primeiro site na nuvem



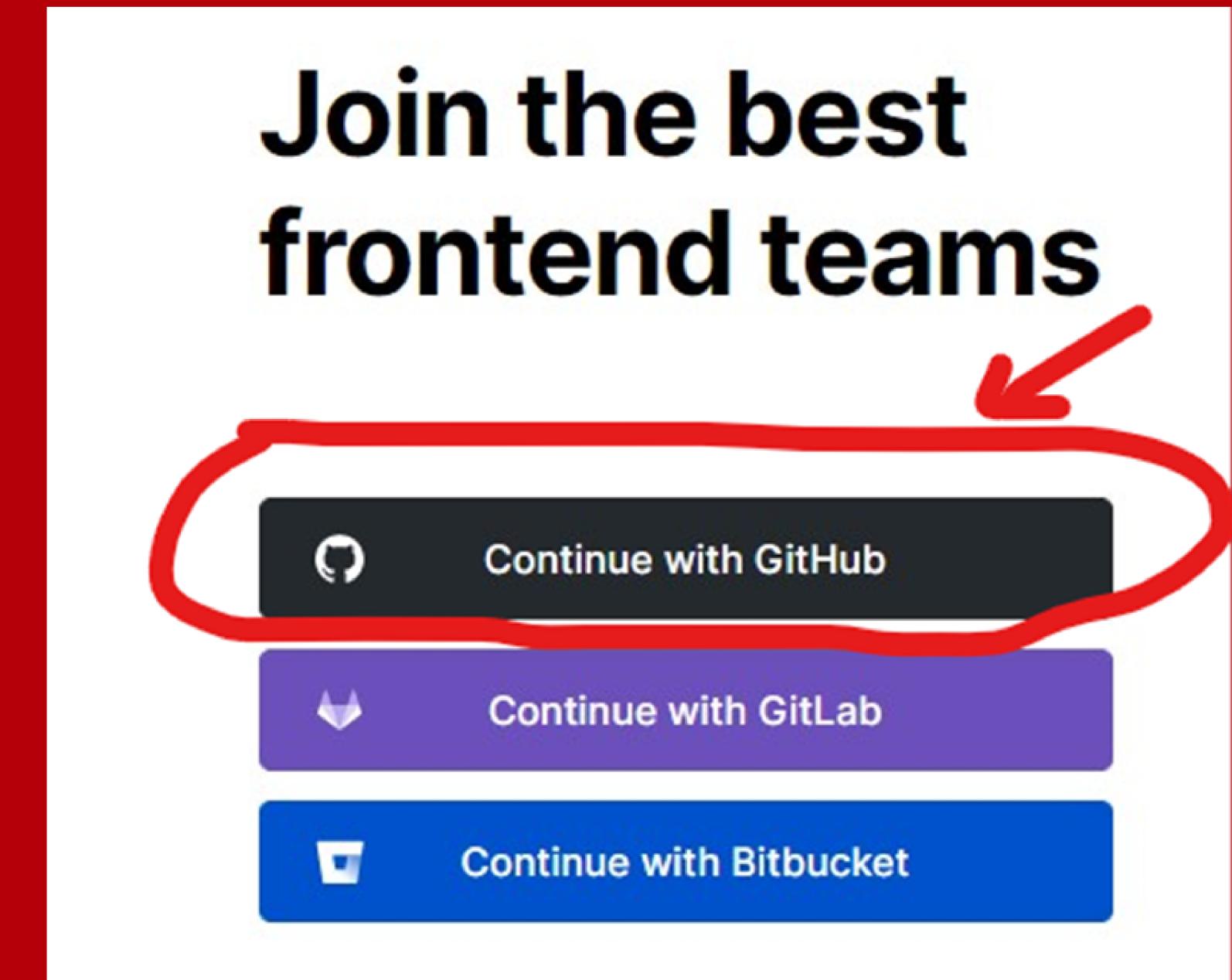
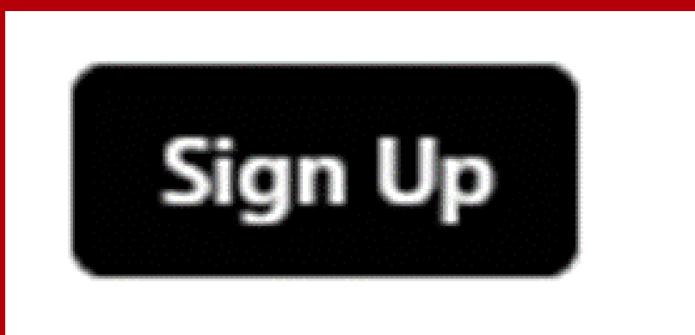
Develop.
Preview.
Ship.



Acesse o site

<https://vercel.com/>

E entre no Sing Up para se registrar:



▲ / maromo71 Hobby ^

Let's build something new.

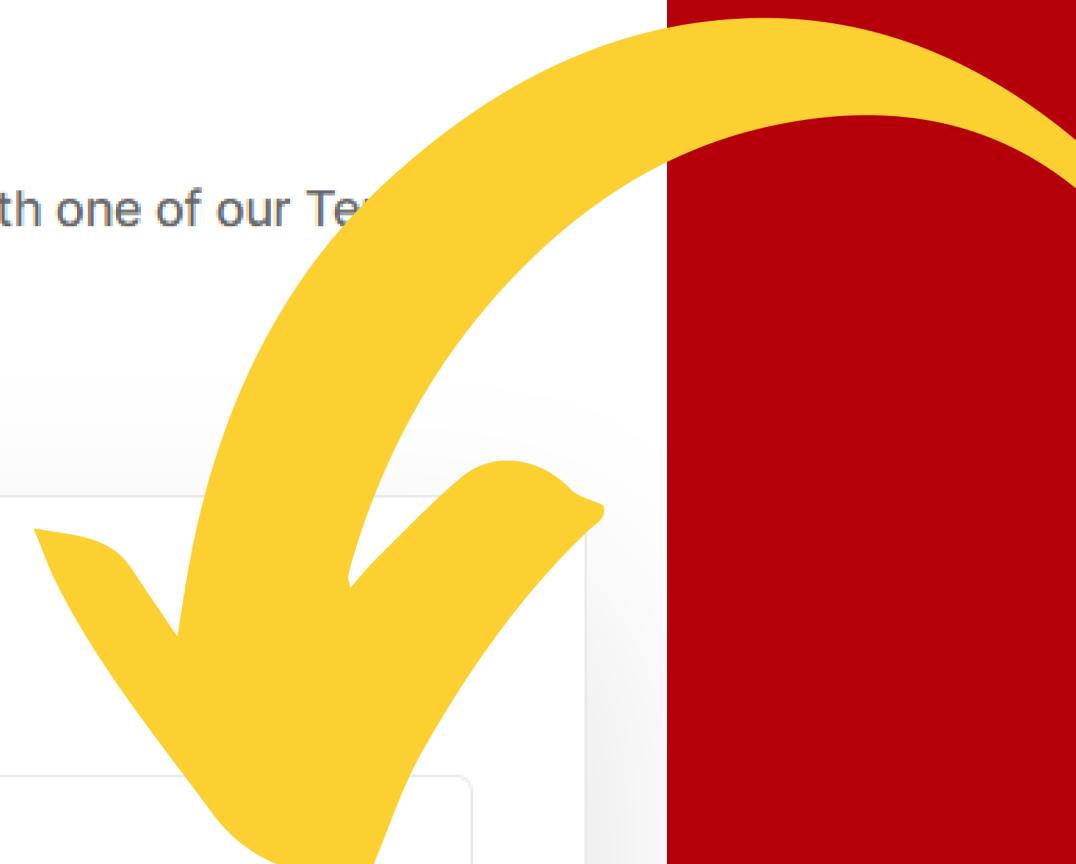
To deploy a new Project, import an existing Git Repository or get started with one of our Templates.

Import Git Repository

maromo71

sample06 · 10m ago

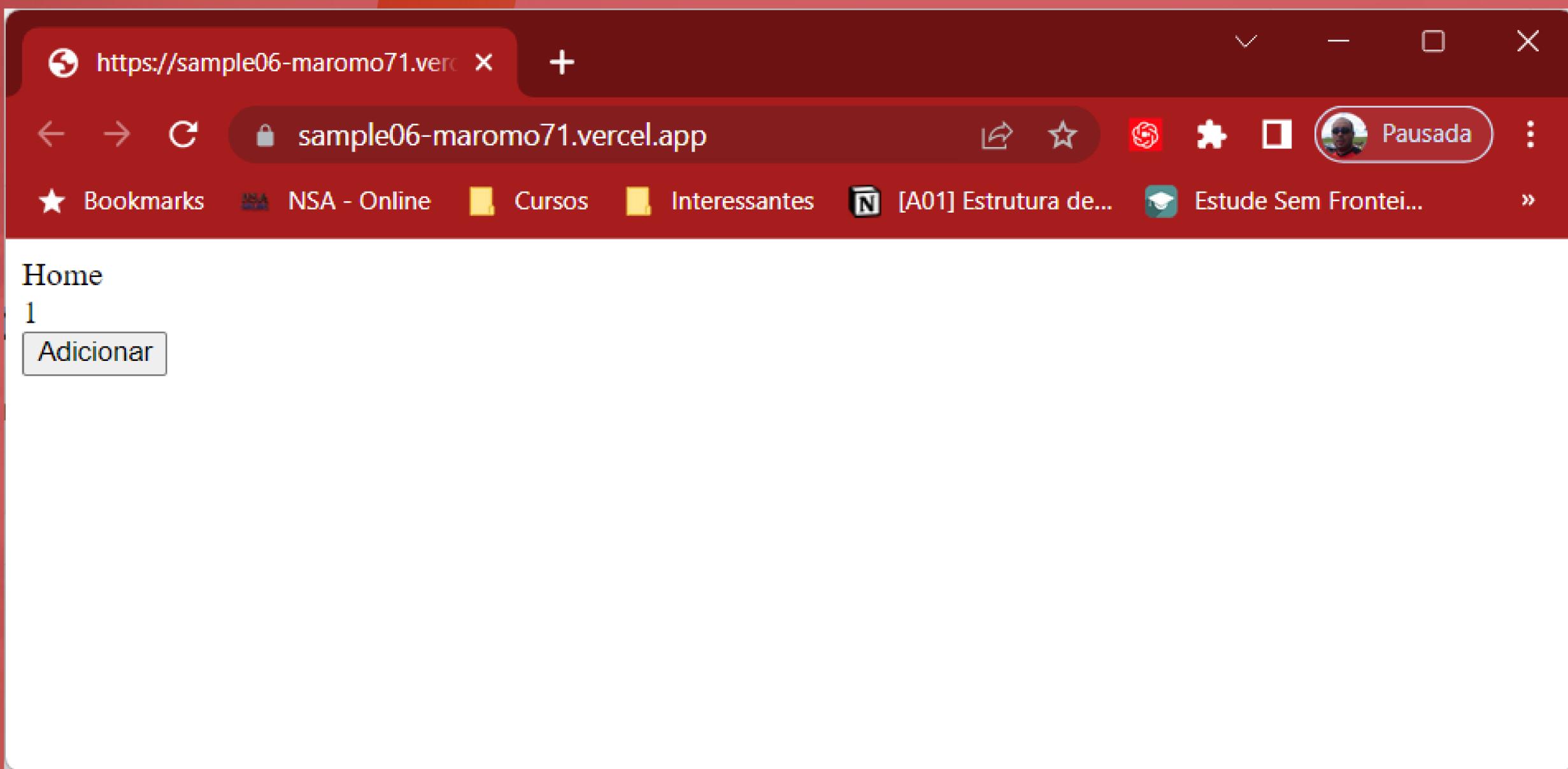
Import



Após este processo de importação,
escolha Deploy

Pronto

<https://sample06-maromo71.vercel.app/>



Resultado da Execução



Resumo

Com a hospedagem na Vercel, seu contador estará disponível para acesso através de um URL público, permitindo que as pessoas visualizem e interajam com ele.

É um exemplo simples. Mas qualquer aplicativo com React e Next.js pode ser hospedado na Vercel.

O sucesso é a soma de pequenos esforços repetidos diariamente.

