

Inicializando um projeto React

O React é uma biblioteca de JavaScript, e pode ser instalado através do npm, que é o gerenciador de pacotes do Node.js. Então é necessário ter o Node.js instalado na máquina.

O próprio time do React recomenda que um projeto seja inicializado utilizando frameworks mais completos, como NextJS, Gatsby ou Remix. Para criar utilizando Next JS:

```
npx create-next-app@latest  
# Irá perguntar o nome do projeto, e outras configurações
```

[Documentação Create Next App](#)

Após o termino da execução do comando, o projeto será criado, e será possível iniciar o servidor de desenvolvimento com o comando `npm run dev`. Que irá iniciar o servidor na porta 3000.

Podemos vincular o projeto a um repositório git, e fazer o deploy em plataformas como Vercel, Netlify, Github Pages, etc.

Para vincular com um repositório já existente, basta adicionar o repositório remoto e fazer o push.

Configuração do projeto

O NextJS já vem com uma série de configurações e funcionalidades prontas, como o suporte a TypeScript, CSS Modules, Sass, Less, Tailwind CSS, etc.

É comum remover as configurações iniciais de estilos e layout, para inicializar um projeto do zero.

Estrutura de pastas

Existem duas maneiras de criar um projeto next, Utilizando [App Router](#) e [Pages Router](#). Atualmente é recomendado a utilização do App Router, que é mais flexível e permite a criação de rotas dinâmicas.

Onde as pastas dentro de `/app` serão rotas, e irão renderizar o arquivo `page.tsx` por padrão, ex: o arquivo `/app/dashboard/pages.tsx` será acessível em `http://localhost:3000/dashboard`.

Rotas App Router

Criando componentes e estilos

É comum organizar componentes dentro de uma pasta `components`, e nomear pastas dentro da `components` com o nome do componente, e o arquivo principal do componente com o nome `index.tsx`. Ex: `/components/Header/index.tsx`.

Também é comum utilizar a extensão `.module.css` para arquivos de estilos, que são módulos de estilos, e são importados no componente.

Para estilizar com CSS Modules, basta importar o arquivo de estilos no componente, e utilizar as classes como propriedades do objeto.

```
.header {  
  background-color: lightblue;  
  padding: 10px;  
}
```

```
import styles from './styles.module.css'  
  
function Header() {  
  return <header className={styles.header}>Header</header>  
}
```

Deploy Github Pages

Antes de fazer o deploy, é necessário configurar o projeto para gerar um build estático, que será utilizado para servir o site no Github Pages.

Para fazer dessa configuração, basta acessar o arquivo `next.config.js` e adicionar a seguinte configuração:

```
module.exports = {  
  output: 'export',  
}
```

Deploy de sites estáticos

Para fazer o deploy de um projeto React no Github Pages, basta acessar a página do repositório, ir na aba **Settings**, e na seção **Github Pages**, selecionar a opção para utilizar Github Actions.

Após isso, podemos verificar os workflows disponíveis no github, e procurar o workflow **Next.js**, selecionar a opção 'configure', que irá criar um novo arquivo no seu projeto chamado **.github/workflows/next.yml**. Basta commitar o arquivo para inicializar um deploy.