

3 - Estrutura de um Projeto Next.js

E aí, programador! o/

Nessa aula vamos entender melhor a estrutura de um projeto Next.js, e não há jeito melhor de fazer isso do que criando um projeto na prática. E ao final nós ainda vamos ver como fazer o deploy do projeto na Vercel.

ATENÇÃO!

Recentemente o Next.js foi atualizado para a versão 13 e o processo de criação de novos projetos está um pouco diferente. Ainda é possível utilizar a mesma estrutura mostrada na aula, com o diretório “pages” e o Bootstrap, porém é preciso selecionar a opção “No” em duas perguntas realizadas durante a criação, sobre o uso do **TailwindCSS** e o sobre o uso do **App Router**. Ao final do módulo teremos aulas extras ensinando a utilizar a nova estrutura de projeto.

```
PS [Projects] npx create-next-app@latest black-nextjs --typescript
✓ Would you like to use ESLint? ... No / Yes
✓ Would you like to use Tailwind CSS? ... No / Yes
✓ Would you like to use 'src/' directory? ... No / Yes
✓ Would you like to use App Router? (recommended) ... No / Yes
✓ Would you like to customize the default import alias (@/*)? ... No / Yes
Creating a new Next.js app in [Projects]black-nextjs.
Using npm.
```

1. Para começar, vamos utilizar o comando abaixo para criar um projeto utilizando typescript:

```
npx create-next-app black-nextjs --typescript
```

2. Vamos abrir a pasta com o VS Code para conseguirmos ver melhor o conteúdo de um projeto recém-criado. Dentro do projeto vamos editar o componente da página inicial:

Obs.: Repare que já estamos trabalhando com o componente Head do Next.js.

```
import type { NextPage } from 'next'
import Head from 'next/head'

const Home: NextPage = () => {
  return (
    <>
      <Head>
        <title>Create Next App</title>
        <meta name="description" content="Generated by create next app" />
        <link rel="icon" href="/favicon.ico" />
      </Head>

      <main>
        <h1>
          Welcome to <a href="https://nextjs.org">Next.js!</a>
        </h1>

        <p>
          Lorem ipsum dolor, sit amet consectetur adipisicing elit. Sint error consectetur modi odio obcaecati tempore dignissimos
        </p>
      </main>
    </>
  )
}

export default Home
```

3. Vamos agora criar algumas novas páginas na pasta “pages” e adicionar o seguinte conteúdo nelas:

Obs.: Crie as páginas products.tsx e cart.tsx, trabalharemos mais a frente com elas.

```
// pages/products.tsx

import { NextPage } from "next";
import Head from "next/head";

const Products: NextPage = () => {
  return (
    <>
      <Head>
        <title>Nossos Produtos</title>
        <meta name="description" content="Conheça todos os nossos produtos" />
        <link rel="icon" href="/favicon.ico" />
      </Head>

      <h1>
        Nossos Produtos
      </h1>
    </>
  )
}

export default Products
```

```
// pages/cart.tsx

import { NextPage } from "next";
import Head from "next/head";

const Cart: NextPage = () => {
  return (
    <>
      <Head>
        <title>Carrinho</title>
        <meta name="description" content="Meu carrinho de compras" />
        <link rel="icon" href="/favicon.ico" />
      </Head>

      <h1>
        Carrinho
      </h1>
    </>
  )
}

export default Cart
```

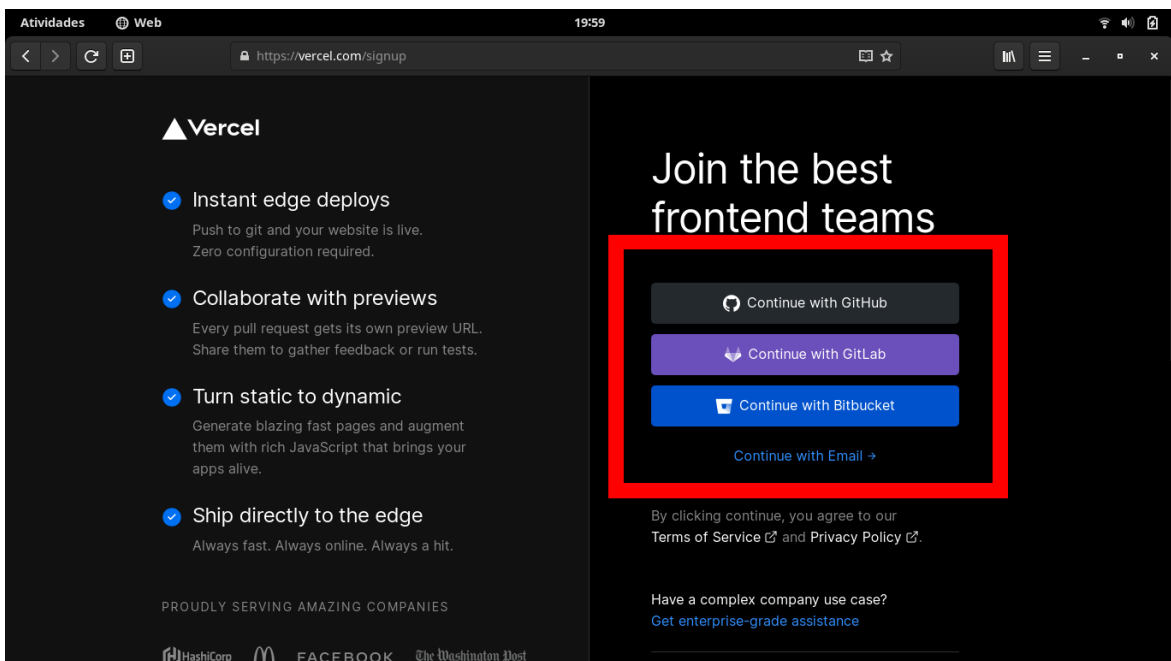
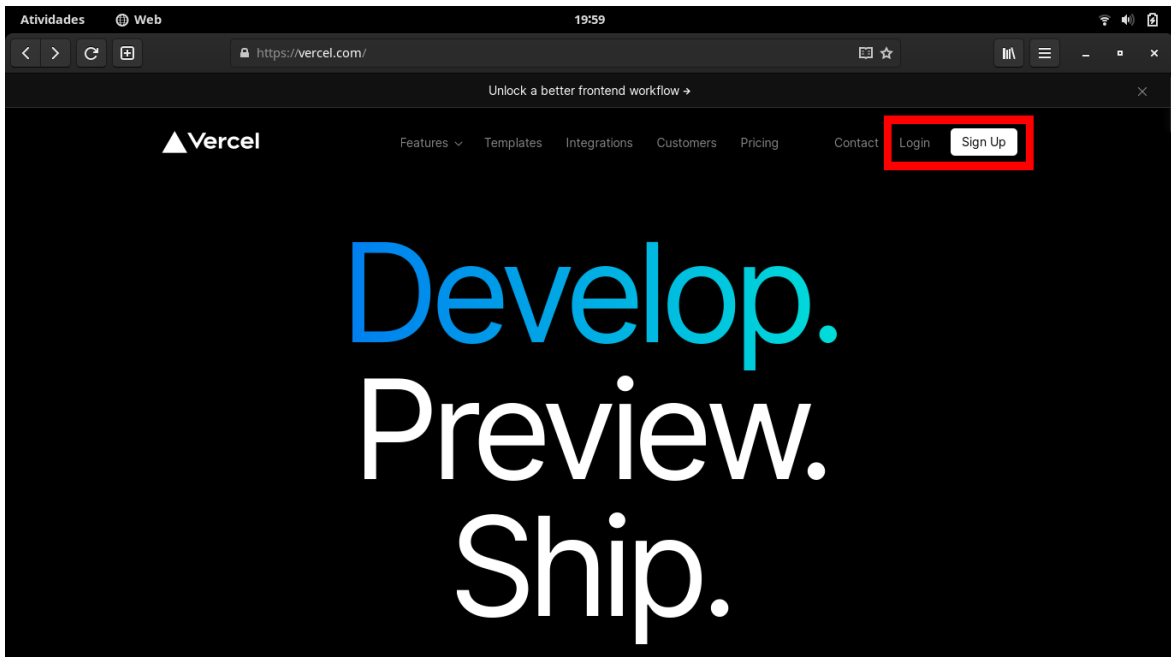
4. Se rodarmos o nosso app e formos ao navegador poderemos ver agora como estão as nossas páginas. Execute o app com o comando abaixo e acesse o caminho da aplicação em <http://localhost:3000/> <http://localhost:3000/products/> e <http://localhost:3000/cart/>

```
npm run dev
```

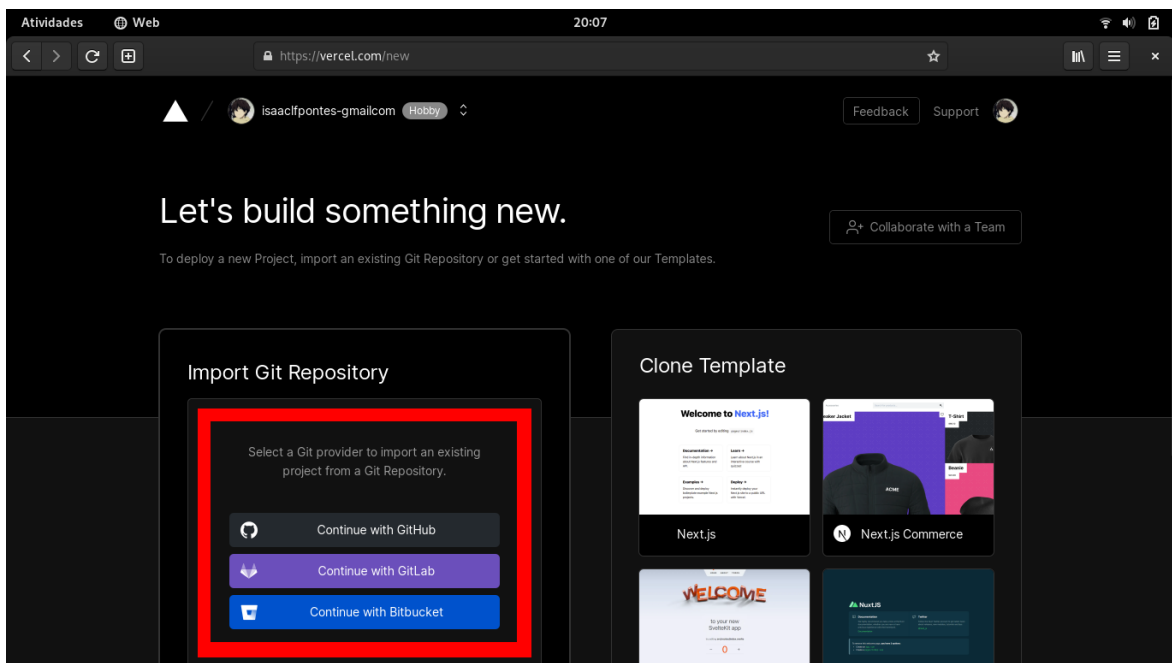
5. Agora vamos ver como funcionam as rotas de API. Repare que na pasta “api” já existe um arquivo “hello.ts”. Nesse arquivo já temos um exemplo de rota de API funcionando para testarmos. Acesse <http://localhost:3000/api/hello> e veja que um JSON é retornado.
6. Antes de passarmos para o deploy vamos subir nossa aplicação para o GitHub. Crie o repositório no GitHub e suba sua aplicação.

Obs.: Não se esqueça de fazer o commit e push das atualizações que fizemos.

7. Agora vamos fazer o deploy da nossa aplicação na Vercel. É muito simples, basta acessar o site <https://vercel.com/> e fazer login ou se registrar (caso ainda não tenha uma conta). Também é possível fazer login com o GitHub.



8. Agora basta ir no botão "New Project" e, caso ainda não tenha concedido o acesso aos seus repositórios, faça isso para que eles estejam disponíveis para serem importados.



9. Clique em "Import" para importar o repositório da aplicação que criamos. Na tela seguinte basta clicar em "Deploy" (as configurações padrão devem bastar por enquanto), aguardar enquanto a Vercel sobe a aplicação e pronto! Nosso app já vai estar online.

