

Primeira aplicação com React

Pablo Henrique

Desenvolvedor Full Stack na Fleye
Educador e Sócio fundador do Catapulta

@pablohdev

Objetivo Geral

Criar nossa primeira aplicação React e consumir dados externo.

Pré-requisitos

- Editor de código (Visual Studio Code)
- Conhecimento básico de React

Percurso

Etapa 1

Criando projeto com React CLI

Etapa 2

Organizando pastas/arquivos

Etapa 3

Criando interface da aplicação GitFind

Percurso

Etapa 4

Requisições com Fetch

Etapa 1

Criando projeto com React CLI

Percurso

~~Etapa 1~~

~~Criando projeto com React CLI~~

Etapa 2

Organizando pastas/arquivos

Etapa 3

Criando interface da aplicação GitFind

Etapa 2

Organizando pastas/arquivos

Percurso

~~Etapa 1~~

~~Criando projeto com React CLI~~

~~Etapa 2~~

~~Organizando pastas/arquivos~~

Etapa 3

Criando interface da aplicação GitFind

Etapa 3

Criando interface do GitFind

Percurso

~~Etapa 1~~

~~Criando projeto com React CLI~~

~~Etapa 2~~

~~Organizando pastas/arquivos~~

~~Etapa 3~~

~~Criando interface da aplicação GitFind~~

Percurso

Etapa 4

Requisições com Fetch

Etapa 4

Requisição com Fetch

Percurso

~~Etapa 4~~

~~Requisições com Fetch~~

Links Úteis

- **Referências:**

- https://developer.mozilla.org/pt-BR/docs/Web/API/Fetch_API/Using_Fetch

Dúvidas?

- > Contem comigo.
- > Comunidade Online (Discord)



Code Style.

Pablo Henrique

Desenvolvedor Full Stack na Fleye
Educador e Sócio fundador do Catapulta

@pablohdev

Objetivo Geral

Conhecer boas práticas em organização e qualidade de código.

Pré-requisitos

- Editor de código (Visual Studio Code)

Percorso

Etapa 1

Eslint

Etapa 2

Prettier

Etapa 3

Editor Config

Etapa 1

Eslint

Eslint

O ESLint é uma ferramenta de análise de código que, juntamente com a sua *extensão* de mesmo nome disponível no VSCode, permite identificar erros quanto ao padrão de escrita que definimos. Com ele você pode, por exemplo, definir que no seu código JavaScript as sentenças sempre terminarão com ponto e vírgula o que após o último elemento de um *array* sempre terá uma vírgula.

Hands On!

***“Falar é fácil.
Mostre-me o código!”***

Linus Torvalds

Percurso

~~Etapa 1~~

~~Eslint~~

Etapa 2

Prettier

Etapa 3

Editor Config

Etapa 2

Prettier

Prettier

É um formatador de código que visa ajudar os desenvolvedores a escrever aplicações que são mais fáceis de entender e mais uniformizadas entre as diversas formas de programar que existem.

Hands On!

*“Falar é fácil.
Mostre-me o código!”*

Linus Torvalds

Percorso

~~Etap~~a 1

~~E~~slint

~~Etap~~a 2

~~P~~rettier

Etap

3

Editor Config

Etapa 3

Editor Config

Editor Config

Nada mais é do que um plugin que obriga o editor de código a seguir de forma prescrita os padrões macro essenciais de formatação configurados pelo usuário.

Em outras palavras, o editorConfig ajuda a estabelecer um estilo de código entre diferentes editores, independente da tecnologia utilizada.

Hands On!

***“Falar é fácil.
Mostre-me o código!”***

Linus Torvalds

Percorso

~~Etapa 1~~

~~Eslint~~

~~Etapa 2~~

~~Prettier~~

~~Etapa 3~~

~~Editor Config~~

Links Úteis

- **Referências:**

- <https://editorconfig.org/>
- <https://prettier.io/>
- <https://eslint.org/>

Dúvidas?

- > Contem comigo.
- > Comunidade Online (Discord)



Conceitos fundamentais para React.

Pablo Henrique

Desenvolvedor Full Stack na Fleye
Educador e Sócio fundador do Catapulta

@pablohdev

Objetivo Geral

Entender os conceitos fundamentais de como o react funciona.

Pré-requisitos

- Editor de código (Visual Studio Code)

Percurso

Etapa 1

JSX

Etapa 2

Componentes de classes e funcionais no React

Etapa 3

Ciclo de vida dos componentes

Etapa 1

JSX

JSX

JSX é uma forma de criar elementos para serem utilizadas como templates de aplicações React. Basicamente, os elementos criados com o JSX são bem similares com código HTML e fornecem aos desenvolvedores uma forma mais simples e intuitiva de criar os componentes de uma aplicação.

Hands On!

***“Falar é fácil.
Mostre-me o código!”***

Linus Torvalds

Percurso

~~Etapa 1~~

~~JSX~~

Etapa 2

Componentes de classes e funcionais no React

Etapa 3

Ciclo de vida dos componentes

Etapa 2

Componentes de classes e funcionais no React

Hands On!

***“Falar é fácil.
Mostre-me o código!”***

Linus Torvalds

Percurso

~~Etapa 1~~

~~JSX~~

~~Etapa 2~~

~~Componentes de classes e funcionais no React~~

Etapa 3

Ciclo de vida dos componentes

Etapa 3

Ciclo de vida dos componentes

Hands On!

***“Falar é fácil.
Mostre-me o código!”***

Linus Torvalds

Percurso

~~Etapa 1~~

~~JSX~~

~~Etapa 2~~

~~Componentes de classes e funcionais no React~~

~~Etapa 3~~

~~Ciclo de vida dos componentes~~

Links Úteis

- **Referências:**

- <https://pt-br.reactjs.org/docs/introducing-jsx.html>

Dúvidas?

- > Contem comigo.
- > Comunidade Online (Discord)



Criando um projeto React do zero

Pablo Henrique

Desenvolvedor Full Stack na Fleye
Educador e Sócio fundador do Catapulta

@pablohdev

Objetivo Geral

Entender como o React funciona e criar um projeto do zero.

Pré-requisitos

- Editor de código (Visual Studio Code)

Percorso

Etapa 1

Babel

Etapa 2

Webpack

Etapa 3

Adicionar React em um projeto existente

Etapa 1

Babel

Babel

É um Transpilador JavaScript de código aberto e gratuito, o Babel possui como uma das principais características converter código JavaScript atual em uma versão que o navegador possa executar.

É amplamente utilizado por desenvolvedores que necessitarem converter recursos da linguagem JavaScript para uma versão que o navegador compreenda.

Hands On!

*“Falar é fácil.
Mostre-me o código!”*

Linus Torvalds

Percurso

~~Etapa 1~~ ~~Babel~~

Etapa 2 **Webpack**

Etapa 3 Adicionar React em um projeto existente

Etapa 2

Webpack

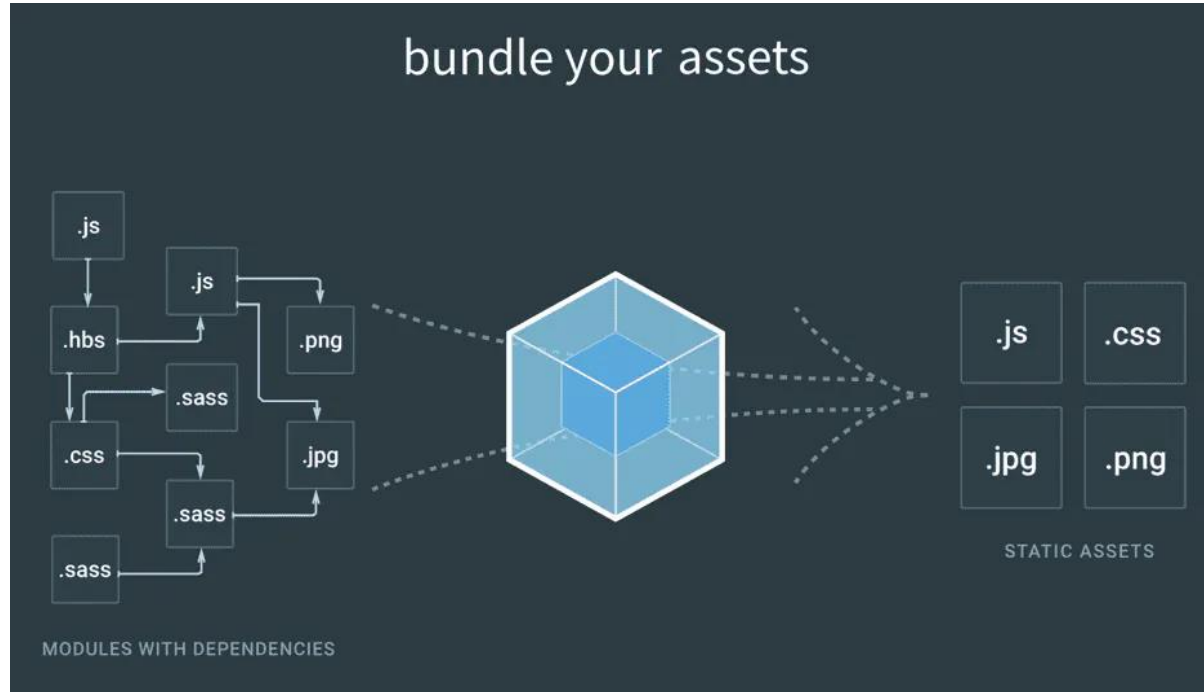
Webpack

webpack é um *module bundler* (empacotador de módulos) para JavaScript, em outras palavras, ele junta os arquivos JS (e também outros formatos) da sua aplicação (seja arquivos seus ou dependências externas) em um arquivo só (ou mais de um), de forma otimizada.

Webpack

Os arquivos são unificados na ordem certa, sem duplicação e podem ser minificados para reduzir o tamanho.

Webpack



Hands On!

***“Falar é fácil.
Mostre-me o código!”***

Linus Torvalds

Percurso

~~Etapa 1~~

~~Babel~~

~~Etapa 2~~

~~Webpack~~

Etapa 3

Adicionar React em um projeto existente

Etapa 3

Adicionar React em um projeto existente

Hands On!

*“Falar é fácil.
Mostre-me o código!”*

Linus Torvalds

Percurso

~~Etapa 1~~

~~Babel~~

~~Etapa 2~~

~~Webpack~~

~~Etapa 3~~

~~Adicionar React em um projeto existente~~

Links Úteis

- Referências:

- <https://pt-br.reactjs.org/docs/getting-started.html>

Dúvidas?

- > Contem comigo.
- > Comunidade Online (Discord)

