

**Aplicações criadas com
React.js e Vite.js**





@EuCarlos



/josecarlos98



carlosalves.now.sh



@eucarlos

Quem sou eu

- Carlos Alves
 - Cursando GTI na Fatec de Itaquá
 - Desenvolvedor Web e de Softwares
 - Freelancer
-
- Linguagens: TS, Ruby e Python
 - Ferramentas: Ubuntu, Git e React

Sumário do conteúdo

1. O que é React.js?
2. Empresas que usam React
3. Documentação do React
4. JSX e TSX
5. O que são componentes
6. Props e State
7. Exemplo de código React
8. Exemplo de estrutura de pasta
9. O que é Vite.js
10. Começando a usar o React.js com Vite.js
11. Código completo no Github
12. Break coffee
14. Exemplo de uso
15. Considerações finais e Trabalhos Futuros
16. Desafio Proposto

O que é React.js


React JS é uma biblioteca JavaScript para a criação de interfaces de usuário (UI).

- Criado em 2011 pelo time do Facebook (atualmente chamada de Meta)
- Os componentes são utilizados para reaproveitamento de código e padronização de interface
- Tem como uma de suas principais características a acessibilidade com que ele é compreendido.

Empresas que usam React



Documentação do React


 **React**

Documentação


Tutorial


Blog

Comunidade

 Buscar docs

v17.0.2

 Languages

GitHub 

Hello World

O menor exemplo de React é algo assim:

```
ReactDOM.render(  
  <h1>Hello, world!</h1>,  
  document.getElementById('root')  
)
```

Isso mostra um header dizendo "Hello, world!" na página.

Experimente no CodePen

Clique no link acima para abrir um editor online. Sinta-se livre para fazer algumas mudanças e

INSTALAÇÃO ▾

PRINCIPAIS CONCEITOS

^

1. Hello World

2. Introduzindo JSX

3. Renderizando Elementos

4. Componentes e Props

5. State e Ciclo de Vida

6. Manipulando eventos

7. Renderização condicional

8. Listas e Chaves

9. Forms

10. Elevando o State

11. Composição vs

Fonte: <https://pt-br.reactjs.org/docs/hello-world.html>

JSX e TSX

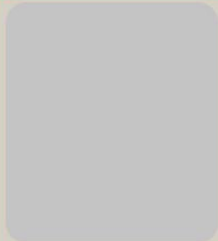

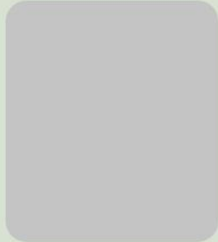



É uma extensão de sintaxe para JavaScript. Recomendamos usar JSX com o React para descrever como a UI deveria parecer. JSX pode lembrar uma linguagem de template, mas que vem com todo o poder do JavaScript.

- JSX produz “elementos” do React.
- TSX nada mais é que JSX aplicado ao TypeScript

O que são componentes

Tudo é um componente

Carrinho de compras

	<div>Nome do Produto RS 00,00 Em estoque Unidades: <input type="text" value="3"/></div> <div></div>
	<div>Nome do Produto RS 00,00 Disponível em 2 dias Unidades: <input type="text" value="--"/></div> <div></div>
	<div>Nome do Produto RS 00,00 Fora de Estoque Unidades: <input type="text" value="--"/></div> <div></div>

- combinável
- reutilizável
- sustentável
- testável

Props (properties)

São **entradas** que **não podem** ser alteradas dentro de um componente. Componentes devem, obrigatoriamente, apenas ler as props. Todos os componentes de React devem ser “puros” e não podem alterar nenhum valor das props.

```
function Welcome(props) {  
  return <h1>Olá, {props.name}</h1>;  
}  
function App() {  
  return (  
    <>  
      <Welcome name="Carlos" />  
    </>  
  );  
}
```

State (estado)

Assim como as propriedades, os states, também são objetos ou informações, mas ao invés de serem passadas para o componente, eles são criados dentro do componente.

- É onde guardamos os dados do nosso componente
- interno apenas
- mutável
- pode ser passado como props

Exemplo de código React

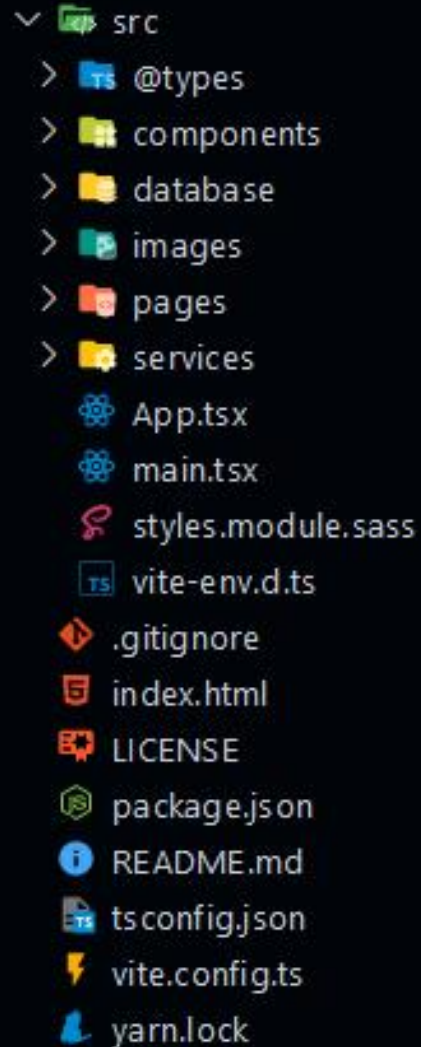
```
import '../styles/cart.min.css'

type Products = {
  "id": string,
  "product_name": string,
  "price": number,
  "image": string,
}

export function ShoppingCart (props: Products) {
  return (
    <>
      <div className="product_image">
        <img src={props.image} alt={props.product_name} />
      </div>

      <div className="product_info">
        <h2>{props.product_name}</h2>
        <div className="price">
          <p>R$ {props.price.toLocaleString("pt-BR")}</p>
        </div>
      </div>
    </>
  )
}
```

Exemplo de estrutura de pastas



A screenshot of a file explorer window showing a project structure. The 'src' directory is expanded, revealing subdirectories and files. The subdirectories are '@types', 'components', 'database', 'images', 'pages', and 'services'. The files are 'App.tsx', 'main.tsx', 'styles.module.sass', and 'vite-env.d.ts'. Below the 'src' directory, the following files are listed: '.gitignore', 'index.html', 'LICENSE', 'package.json', 'README.md', 'tsconfig.json', 'vite.config.ts', and 'yarn.lock'.

- src
 - @types
 - components
 - database
 - images
 - pages
 - services
 - App.tsx
 - main.tsx
 - styles.module.sass
 - vite-env.d.ts
- .gitignore
- index.html
- LICENSE
- package.json
- README.md
- tsconfig.json
- vite.config.ts
- yarn.lock

O que é Vite.js

É uma ferramenta de construção de front-end de próxima geração.

- Desenvolvido pelo do mesmo criador do Vue.js
- É um bundler e um dev-server que você consegue usar compiladores e frameworks direto da caixa, com zero configuração
- APIs programáticas flexíveis com digitação TypeScript completa

Começando a usar o React.js com Vite.js

yarn

```
yarn create vite name-project --template react-ts
```

npm 6.x

```
npm create vite@latest name-project --template react-ts
```

npm 7+, extra double-dash is needed:

```
npm create vite@latest name-project -- --template react-ts
```

pnpm

```
pnpm create vite name-project -- --template react-ts
```

```
cd my-project && yarn install
```

```
yarn dev || npm run dev
```

Código completo no Github

<https://github.com/EuCarlos/react-and-vitejs-presentation>

The background is a dark blue field filled with a repeating pattern of yellow-outlined food icons. These icons include burgers, hot dogs, french fries, soft drinks, ice cream cones, ice cream bars, donuts, and tacos, arranged in a grid-like fashion.

Break coffee

Exemplo Prático

Considerações finais e Trabalhos futuros

Estude mais

Canais no youtube

- Rocketseat
- FreeCodeCamp
- Otávio Miranda
- Roger Melo
- Willian Justen Cursos

Cursos

- Origamid
- Cod3r Cursos

Livros

- Primeiros Passos com React: Construindo Aplicações web
- React - Aprenda Praticando
- React Cookbook

Blogs

- Luiz Tools
- Carlos Alves
- Blog React

Duvidas?

Contatos



@eucarlos



carlosalves.now.sh



carlosalves369m@protonmail.com

Desafio Proposto