

Componentes React

Link Notion: <https://cherry-client-b8f.notion.site/Componentes-React-232911d84e0d80798d4bd8599cff8c74?pvs=73>

O que é um componente reutilizável?

Um componente reutilizável é uma parte independente da interface do usuário que pode ser usada em vários lugares da aplicação. A ideia é escrever o componente uma vez e usá-lo onde for necessário, em vez de duplicar o código.

Estrutura

Um componente React básico é, na maioria das vezes, apenas uma função exportada que retorna JSX (HTML + JS).

```
function MeuComponente() {
  return (
    <div>
      <h1>Olá, mundo!</h1>
      <p>Esse é meu primeiro componente React.</p>
    </div>
  );
}

export default MeuComponente;
```

Usando um componente

Se você criar **MeuComponente.jsx**, pode usá-lo em outro arquivo **fazendo o import** no início do arquivo e utilizando como se ele fosse uma tag html com auto fechamento(com a barra fechando ao final, como o , por exemplo):

```
import MeuComponente from "./MeuComponente";

function App() {
  return (
    <div>
      <MeuComponente />
    </div>
  );
}

export default App;
```

Import e Export

No JavaScript moderno (ES6), o **import** e **export** são usados para compartilhar código entre arquivos. Assim você consegue dividir sua aplicação em módulos mais organizados.

Export

Existem duas formas principais: **export nomeado** e **export default**.

Export nomeado

Você exporta várias coisas de um mesmo arquivo, cada uma com seu nome.

```
// utils.jsx
export function soma(a, b) {
  return a + b;
}

export const PI = 3.14159;
```

Export Default

Cada arquivo pode ter apenas um export default.

É usado quando aquele módulo tem uma “função principal”.

```
// calculadora.jsx
export default function multiplicar(a, b) {
  return a * b;
}
```

Import

No outro arquivo onde você irá utilizar os recursos exportados, você importa o que precisa no início do arquivo.

Import de export nomeado

É necessário usar o mesmo nome. Você colocará o(s) nome(s) **entre chaves** e após o **from**, o caminho do arquivo onde estão os exports.

```
// main.jsx
import { soma, PI } from './utils.jsx';

console.log(soma(2, 3)); // 5
console.log(PI);        // 3.14159
```

Também é possível renomear, usando a palavra-chave **as**.

```
import { soma as adicionar } from './utils.jsx';

console.log(adicionar(10, 5)); // 15
```

Import de export default

Você pode escolher o nome que quiser para referenciar o elemento exportado.

```
// main.js
import multiplica from './calculadora.jsx';

console.log(multiplica(4, 2)); // 8
```

Observações importantes

1. Para usar import e export no navegador, precisa usar type="module" no <script>:

```
<script type="module" src="main.jsx"></script>
```

2. No Node.js, você precisa ativar ESM Modules (**colocar "type": "module" no arquivo package.json**).

Boas práticas

No React existem alguns padrões de boas práticas para criar componentes, que ajudam a manter o código organizado, reutilizável e fácil de manter.

1. Nome dos componentes

- Sempre com letra maiúscula (Botao, CardUsuario).
- Use nomes descritivos, que expliquem o que o componente faz.

Não recomendado

```
function comp1() { ... }
```

Correto

```
function CardProduto() { ... }
```

2. Componentes pequenos e focados

- Cada componente deve fazer apenas uma coisa (Single Responsibility).
- Evite criar um componente gigante com toda a UI da página.

Não recomendado

```
function PaginaCompleta() {
  return (
    <header>...</header>
    <main>...</main>
    <footer>...</footer>
  );
}
```

Correto

```
function Header() { ... }
function Main() { ... }
function Footer() { ... }

function Pagina() {
  return (
    <>
      <Header />
      <Main />
      <Footer />
    </>
  );
}
```

```
);  
}
```

3. Props e composição

- Passe dados para componentes via props, não com variáveis globais.
- Use composição ao invés de tentar fazer tudo em um único componente.

```
function Card({ titulo, conteudo }) {  
  return (  
    <div className="card">  
      <h2>{titulo}</h2>  
      <p>{conteudo}</p>  
    </div>  
  );  
}  
  
// Uso  
<Card titulo="Notícia 1" conteudo="Detalhes da notícia" />
```

4. Pastas e organização

- Geralmente se organiza cada componente em sua própria pasta, sempre dentro da pasta assets:

```
/components  
/Card  
  Card.jsx  
  Card.css  
/Botao  
  Botao.jsx  
  Botao.css
```

Props e Children

Props

São as propriedades de um componente, que tem a função de transmitir informações entre componentes, criando uma característica **reutilizável** dos componentes.

```
//App.jsx  
import Welcome from './Welcome';  
  
function App() {  
  return (  
    <div className="App">  
      <Welcome name="John"/>  
      <Welcome name="Mary"/>  
      <Welcome name="Alex"/>  
    </div>  
  );  
}
```

```
//Welcome.jsx  
function Welcome(props) {  
  return <h1>Hello, {props.name}</h1>;  
}
```

Props por desestruturação

Normalmente as props são passadas por desestruturação, que é quando transformamos cada posição do array ou chave de um objeto em uma variável ao recebê-los nos parâmetros do componente.

```
function Welcome({name}) {  
  return <h1>Hello, {name}</h1>;  
}
```

O nome do parâmetro **deve ser o mesmo** que o nome da propriedade do objeto recebido.

Children

Children é uma prop especial que representa tudo que está dentro da tag do componente.

```
function Card({ children }) {  
  return <div className="card">{children}</div>;  
}  
  
// Uso  
<Card>  
  <h2>Notícia 1</h2>
```

```
<p>Conteúdo da notícia</p>
</Card>
```

Valores padrão

Quando falamos de valores padrão em um componente, estamos nos referindo a valores que o componente usa caso nenhuma prop seja passada. Eles funcionam como uma "reserva" para garantir que o componente sempre tenha algo para renderizar ou operar, evitando undefined ou erros. Se texto ou cor não forem passados, o valor após o = será usado.

```
function Botao({ texto = "Clique aqui", cor = "blue" }) {
  return <button style={{ backgroundColor: cor }}>{texto}</button>;
}
```

Referências

https://www.alura.com.br/artigos/componentes-do-react?utm_campaign=Newsletter-Dev-em-T&utm_medium=email&_hsenc=p2ANqtz-_jsTLbuhP8Kk7pXbmZ3DPGpEj_MUS60_N-vcW0GaoMnioH6jS_josALX_rdr0kVmHKC6jWRmxxaq0UDM0iZdiJoLQQB_PDFi1XVhxSNZ6lAKonLI&_hsmi=342850710&utm_content=342850710&utm_source=hs_email

<https://medium.com/@rashmipatil24/reusable-react-components-7b025f232ca4>

<https://react.dev/learn/your-first-component>