Atividade: Componentes em React

Olá! Com a chuva, é uma ótima ideia usarmos este tempo para entender melhor os **Componentes em React**, que são a base para construir interfaces de forma organizada e eficiente.

---

1. O que são componentes em React?

Componentes em React são **blocos de construção reutilizáveis da interface do usuário (UI)**.
Pense neles como pequenas partes da tela, cada uma com sua própria lógica, que juntas formam a aplicação inteira. Eles são criados usando código e retornam **JSX**, que é a forma do React de descrever a aparência da interface.

Para que servem?

- Dividir a aplicação em partes menores, facilitando o gerenciamento de cada uma.
- Organizar a interface e **reutilizar código**, evitando repetição.
- Tornar a manutenção e os testes muito mais fáceis.

Como usar um?

Para usar um componente, você primeiro precisa criá-lo, geralmente como uma **função** que recebe dados (chamados de **props**) e retorna o que será exibido na tela.
Por exemplo, um `CardNoticias` pode ser usado para exibir diferentes notícias, bastando passar o título, a imagem e o resumo de cada uma.

Por que utilizamos?

- Para tornar o código mais limpo e organizado.
- Para reutilização de código: se um elemento se repete, você só precisa criá-lo uma vez.
- Para facilitar a manutenção: alterações em um componente se refletem em todos os lugares onde ele é usado.
- Para facilitar o trabalho em equipe: várias pessoas podem trabalhar em diferentes componentes simultaneamente.

Qual a importância de organizar o código em componentes?

- Facilita o crescimento de um projeto.
- Garante **clareza** no código.
- Ajuda a evitar **bugs**.
- Cada componente cuida de uma parte específica da interface, tornando o projeto mais robusto e fácil de escalar.

---

2. Análise do site do Terra

Observando a página inicial do site do Terra, podemos identificar várias partes que seriam ótimos componentes React:

1. **Menu de navegação (Header):**
2. Reaproveitado em todas as páginas.
3. Um componente `Header` garante consistência e facilita alterações futuras.
4. **Card de notícia:**
5. A página possui vários cards com mesma estrutura (imagem, título e resumo).
6. Um componente `CardNoticia` permite reutilizar essa estrutura apenas passando dados diferentes para cada notícia.
7. **Rodapé (Footer):**
8. Presente em todas as páginas, contendo links institucionais e direitos autorais.
9. Um componente `Footer` garante que qualquer atualização seja feita em apenas um lugar.

Exemplo Prático: Usando o Componente CardNoticias

1. O Componente Reutilizável (`CardNoticias.jsx`)

Este é o nosso bloco de construção. Ele recebe as informações específicas de cada notícia (as **props**) e define a estrutura HTML (JSX) que será repetida.

```
// O componente é uma função que recebe as 'props' (propriedades)
function CardNoticias({ titulo, imagem, resumo, link }) {
  return (
    // O componente retorna o JSX (estrutura visual)
    <div className="card-noticia">
      <img src={imagem} alt={titulo} />
      <h3>{titulo}</h3>
      <p>{resumo}</p>
      <a href={link}>Leia mais</a>
    </div>
  );
}

export default CardNoticias;
```

---

2. Usando o Componente (O "Container" de Notícias)

Este é o componente "pai" que organiza a interface e usa `CardNoticias` várias vezes. Ele simula buscar uma lista de dados e iterar sobre ela.

```
// Imagine que esses dados vieram de uma API
const listaDeNoticias = [
  {
    id: 1,
    titulo: "React é como LEGO",
    resumo: "Componentes são como peças de LEGO, permitindo reuso e
organização.",
    imagem: "imagem-lego.jpg",
    link: "/artigo/lego"
  },
  {
    id: 2,
    titulo: "Organização no Código",
    resumo: "Um código bem organizado facilita a manutenção e evita bugs.",
    imagem: "imagem-clean-code.jpg",
    link: "/artigo/organizacao"
  },
  // Mais notícias poderiam ser listadas aqui...
];

function SecaoDestaques() {
  return (
    <section>
      <h2>Últimas Notícias</h2>
      <div className="lista-cards">
        {/* Usamos map para reutilizar o componente para cada item da lista
*/}
        {listaDeNoticias.map((noticia) => (
          <CardNoticias
            key={noticia.id} // "key" é importante para o React
            titulo={noticia.titulo}
            imagem={noticia.imagem}
            resumo={noticia.resumo}
            link={noticia.link}
          />
        ))}
      </div>
    </section>
  );
}

export default SecaoDestaques;
```

Conclusão

Com este método, você **não precisa copiar e colar** o `<div>...</div>` do Card de Notícias para cada item.
Você escreve a estrutura uma vez em `CardNoticias.jsx` e a reutiliza para montar toda a seção. Qualquer alteração na estrutura visual acontece em apenas **um lugar**.

Componentes em React - Guia Ilustrado

Este material mostra como funcionam os componentes em React, trazendo exemplos de código e também **imagens ilustrativas** de como ficaria a interface renderizada.

Exemplo 1: Componente Reutilizável - CardNoticias

```
function CardNoticias({ titulo, imagem, resumo, link }) {
  return (
    <div className="card-noticia">
      <img src={imagem} alt={titulo} />
      <h3>{titulo}</h3>
      <p>{resumo}</p>
      <a href={link}>Leia mais</a>
    </div>
  );
}

export default CardNoticias;
```

■ Visual aproximado do **CardNoticias**:

Exemplo 2: Componente Pai - SecaoDestaques

```
function SecaoDestaques() {
  return (
    <section>
      <h2>Últimas Notícias</h2>
      <div className="lista-cards">
        {listaDeNoticias.map((noticia) => (
```

```
          <CardNoticias
            key={noticia.id}
            titulo={noticia.titulo}
            imagem={noticia.imagem}
            resumo={noticia.resumo}
            link={noticia.link}
          />
        ))}
      </div>
    </section>
  );
}

export default SecaoDestaques;
```

■ Visual aproximado da **Seção de Notícias**:

Ao usar componentes em React, evitamos repetição de código, aumentamos a clareza e facilitamos manutenção. Alterar a estrutura de um componente afeta todos os lugares onde ele é usado automaticamente.