

Criando um Site React: Da Estrutura Básica às Páginas Completas

Por Escola Dnc

Introdução

Este ebook aborda o processo de criação de um site utilizando React, focando na estruturação de componentes reutilizáveis e na construção de múltiplas páginas. Vamos explorar como criar uma base sólida para o site e, em seguida, expandir essa estrutura para incluir páginas como Home, Projects, About e Contact. O objetivo é demonstrar como a componentização facilita o desenvolvimento e manutenção de um site moderno e responsivo.

Estrutura Básica e Componentização

Criando Componentes Reutilizáveis

A base de um site React bem estruturado está na criação de componentes reutilizáveis. Neste projeto, começamos desenvolvendo componentes fundamentais como:

- Header
- Footer
- Banner

Estes componentes serão utilizados em várias páginas do site, proporcionando consistência visual e facilitando a manutenção.

Vantagens da Componentização

- **Reutilização de código:** Componentes podem ser facilmente reutilizados em diferentes partes do site.
- **Manutenção simplificada:** Alterações em um componente refletem em todas as suas instâncias.
- **Organização do código:** Estrutura mais clara e modular.

Dica: Não tenha medo de copiar e colar componentes entre páginas. Esta prática é comum e eficiente no desenvolvimento React.

Estrutura de Arquivos

Para cada componente, criamos dois arquivos principais:

1. `NomeDoComponente.jsx` : Contém a lógica e estrutura do componente.
2. `NomeDoComponente.css` : Contém os estilos específicos do componente.

Esta separação ajuda a manter o código organizado e facilita a manutenção.

Desenvolvimento das Páginas

Página Home e Projects

As páginas Home e Projects foram as primeiras a serem desenvolvidas, utilizando os componentes base (Header, Footer, Banner) como estrutura inicial.

Página About

Estrutura da Página About

1. Importação dos componentes base (Header, Footer, Banner).
2. Criação de um novo componente `AboutText` para o conteúdo específico.

Componente AboutText

- **Estrutura:** Duas colunas de texto.
- **Arquivos:** `AboutText.jsx` e `AboutText.css`.
- **Conteúdo:**
 - Título (H2)
 - Parágrafos de texto

Estilização do AboutText

- Uso de classes CSS para estilização.

- Implementação de layout flexbox para as colunas.
- Responsividade com media queries para dispositivos móveis.

Dica de produtividade: Use atalhos como Alt (no Windows) para seleção múltipla ou Ctrl+F para substituições rápidas no VS Code.

Página de Contato

Estrutura da Página de Contato

1. Reutilização dos componentes base (Header, Footer, Banner).
2. Criação de um novo componente `ContactForm`.

Componente ContactForm

- **Arquivos:** `ContactForm.jsx` e `ContactForm.css`.
- **Estrutura inicial:**
 - Título (H2)
 - Espaço reservado para o formulário (a ser implementado posteriormente)

Estilização do ContactForm

- Uso de flexbox para centralização.
- Definição de margens e largura máxima.
- Estilização básica do título.

Técnicas e Boas Práticas

Reutilização de Componentes

A criação de componentes reutilizáveis como Header, Footer e Banner permite:

- Rápida construção de novas páginas.
- Consistência visual em todo o site.
- Facilidade de manutenção e atualização.

Responsividade

- Implementação de media queries para adaptar o layout em dispositivos móveis.
- Uso de unidades flexíveis (como porcentagem) para dimensões.

Organização de Estilos

- Criação de um arquivo `main.css` para estilos globais.
- Uso de classes específicas para cada componente.
- Implementação de variáveis CSS para cores e tamanhos recorrentes.

Planejamento Futuro

- Integração com API para conteúdos dinâmicos.
- Implementação completa do formulário de contato.
- Adição de interatividade e funcionalidades específicas.

Conclusão

Neste ebook, exploramos o processo de criação de um site React, desde a estruturação básica até o desenvolvimento de múltiplas páginas. Aprendemos a importância da componentização, reutilização de código e organização de estilos.

O site agora possui uma estrutura sólida com páginas Home, Projects, About e Contact, cada uma utilizando componentes compartilhados e específicos. Esta abordagem não apenas acelera o desenvolvimento, mas também facilita futuras expansões e manutenções.

Lembre-se, o desenvolvimento web é um processo contínuo. As bases estabelecidas aqui permitem fácil adição de novas funcionalidades e melhorias no futuro. Continue explorando e aprimorando suas habilidades em React para criar sites ainda mais dinâmicos e interativos.

