

Pensamento componentizado

Componentes de Interface

Vimos diversos tipos de componentes e a cada aula outros novos vão surgindo. Componentes de Interface são aqueles que representam apenas uma **parte visual** da aplicação. Em geral podemos deixar os componentes mais organizados e a nossa aplicação como um todo mais escalável.

Componentes de Interface de Contexto

São aqueles componentes visuais que servem apenas para algum contexto específico da aplicação, dificilmente são reutilizados pois o real objetivo desses componentes é organizar nosso componente principal e extrair pequenas partes dele em novos componentes.

Componentes de Interface Utilitários

Parecido com os componentes de contexto, esses devem ser bem mais reutilizados. Devem conter as possíveis variações e geralmente possuem bastante `props` para manter a alta reutilização.

Organização do projeto

Pastas importantes

- **node_modules** → Pasta com todas as dependências da aplicação. Não deve ser alterada manualmente, apenas por outras ferramentas como NPM e Yarn.
- **src** → Toda a implementação da nossa aplicação fica nessa pasta, a partir dela devemos organizar os arquivos em sub-pastas:
 - **assets** → Imagens, ícones, videos, fontes e etc. Pode ser organizado com sub-pastas para cada necessidade também.
 - **components** → Componentes que fazem parte da nossa aplicação, cada contexto de componente deve ter a sua própria pasta. Dentro de cada componente também podem ter novas subpastas para organizar componentes de Contexto
 - **utils** → Componentes utilitários de alta reutilização.

- **contexts** → Pasta para separar e organizar componentes com Context API (vemos isso mais adiante no curso)
- **pages** → Organizar componentes referente ao layout e disposição de páginas diferentes
- **utils** → Pasta para organizar funções de alta reutilização como manipuladores de datas, números, *strings* e etc.

Arquivos importantes

- **App.js** → Arquivo principal que renderiza nossa aplicação inteira.
- **.env** → Arquivo para configuração de variáveis de ambiente, cada variável DEVE ter o prefixo `REACT_APP`
- **package.json** → Arquivo de declaração de configurações da aplicação, dependências, versão e etc.
- **README.md** → Arquivo para ter uma documentação breve e guiada de o que é a aplicação, como utilizar e os processos para rodar o projeto localmente

© Curso Online de React do Zero ao Pro
Desenvolvido por Gustavo Vasconcellos e EBAC Online