1) Criar uma pasta no seu projeto chamada "componentes"; 2) Criar uma pasta no seu projeto chamada "imagens"; Discord Alura Fórum do curso 3) E, finalmente, baixar as imagens do projeto e colocar na pasta Voltar para Dashboard imagens. vitrine dev Conheça o Vitrine.Dev Criando componente de Logo **MODO NOTURNO** 1) Criar uma pasta chamada "Logo" dentro da pasta de componentes; Henrique Galvão Prezotti 9 2) Dentro da pasta criada, crie dois arquivos: "estilo.css" para estilização e 25.2k xp "index.js" para a criação do componente. Componente 1) No arquivo index.js, escreva o código do componente de Logo começando por uma "carcaça" básica, conforme a seguir: function Logo() { return ( **COPIAR CÓDIGO** 2) Agora, adicione os imports do que será usado no componente (neste caso é o CSS e a imagem do logo). Confira abaixo: import './estilo.css' import logo from '.../.../imagens/logo.svg' **COPIAR CÓDIGO** 3) Depois de fazer os imports, podemos fazer o código do componente, que será uma imagem e um texto com suas respectivas classes e atributos: function Logo() { return ( <div className='logo'> <img src={logo} alt='logo' className='logo-img' ></img> <strong>Alura</strong>Books </div> **COPIAR CÓDIGO** 4) Após esta etapa, resta apenas a etapa de exportação (export default) Export Default Logo. 5) Então, após concluir essas etapas, o resultado final do componente fica assim: import './estilo.css' import logo from '../../imagens/logo.svg' function Logo() { return ( <div className='logo'> <img src={logo} alt='logo' className='logo-img' ></img> <strong>Alura</strong>Books </div> export default Logo **COPIAR CÓDIGO Estilo** 1) No arquivo CSS da pasta Logo você deve adicionar os seguintes estilos: .logo { display: flex; font-size: 30px; .logo-img { margin-right: 10px; **COPIAR CÓDIGO Componente OpcoesHeader** Bora criar mais um componente?! Componente 1) Repetiremos os mesmos primeiros passos do componente feito anteriormente. Para isso, crie uma nova pasta chamada OpcoesHeader dentro da pasta "componentes"; 2) Dentro da pasta OpcoesHeader, crie o estilo.css; 3) Em seguida, crie a carcaça e faça o import do css; 4) Neste caso, iremos usar uma constante com valores que serpa útil para a criação do componente: import './estilo.css' const textoOpcoes = ['CATEGORIAS', 'FAVORITOS', 'MINHA ES] function OpcoesHeader() { return ( export default OpcoesHeader **COPIAR CÓDIGO** 5) Agora iremos aplicar a função nativa do JavaScript(map) para varrer esta lista de valores criando elementos HTML para cada item da lista. 6) o resultado final é: import './estilo.css' const textoOpcoes = ['CATEGORIAS', 'FAVORITOS', 'MINHA ES1 function OpcoesHeader() { return ( { textoOpcoes.map( (texto) => ( {texto} ) ) } export default OpcoesHeader **COPIAR CÓDIGO Estilo** 1) No arquivo estilo.css da pasta do componente, devemos escrever os seguintes estilos: .opcoes { display: flex; .opcao { font-size: 16px; display: flex; justify-content: center; align-items: center; text-align: center; height: 100%; padding: 0 5px; cursor: pointer; min-width: 120px; **COPIAR CÓDIGO Componente IconesHeader** Te convido para criar um componente dos ícones do header, vamos lá? Componente 1) Repetir os mesmos passos de criação de componente, e criar a pasta dentro de "componentes", chamada IconesHeader. Dentro da pasta, crie o *index.js* e o *estilo.css*. 2) No index.js, vamos repetir os mesmos passos do componente anterior; porém ao invés de usarmos uma lista de textos, será uma lista de imagens importadas: import perfil from '../../imagens/perfil.svg' import sacola from '../../imagens/sacola.svg' const icones = [perfil, sacola] **COPIAR CÓDIGO** 3) Então, o resultado final fica assim: import './estilo.css' import perfil from '../../imagens/perfil.svg' import sacola from '../../imagens/sacola.svg' const icones = [perfil, sacola] function IconesHeader() { return ( { icones.map( (icone) => ( <img src={icone}></img> )) } export default IconesHeader **COPIAR CÓDIGO Estilo** 1) No arquivo estilo.css da pasta do componente, vamos escrever os seguintes estilos: .icones { display: flex; align-items: center; .icone { margin-right: 40px; width: 25px; **COPIAR CÓDIGO Componente Header** Componente 1) Vamos repetir os mesmos passos de criação de componente, e gerar a pasta dentro de "componentes" chamada Header. Dentro da pasta, crie index.js e estilo.css; 2) Esse componente servirá apenas como um agregador de todos os anteriores, então só precisamos importar os outros componentes e usálos dentro de uma tag header com sua respectiva classe: import './estilo.css' import Logo from '../Logo' import OpcoesHeader from '../OpcoesHeader' import IconesHeader from '../IconesHeader' function Header() { return ( <header className='App-header'> <Logo/> <OpcoesHeader/> <IconesHeader/> </header> export default Header **COPIAR CÓDIGO Estilo** 1) No arquivo estilo.css, da pasta do componente, vamos escrever os seguintes estilos: .App-header { background-color: #FFF; display: flex; justify-content: center; **COPIAR CÓDIGO Utilizando Header** 1) Agora que criamos o componente de Header, precisamos importá-lo e usá-lo no App.js para vermos na nossa tela. import './App.css'; import Header from './componentes/Header' function App() { return ( <div className='App'> <Header /> </div> ); export default App **COPIAR CÓDIGO** DISCUTIR NO FORUM

07 Importante: atualizar o CSS

Criando uma aplicação React

Para saber mais: entendendo o "Export

Simplificando o código

O que aprendemos?

20min

23min

8 Usando Loops

Default"

12 Mão na massa!

sequência:

**componentes React!** 

Preparação para criar componentes

Chegou a hora de praticar e criar seus próprios

Nesta aula, criamos nossos primeiros componentes do Alura Books.

Vamos praticar o passo a passo que eu fiz durante a aula? Para isso, siga a

PRÓXIMA ATIVIDADE