

FIAP

GRADUAÇÃO

45697056



TDS

FRONT-END DESIGN ENGINEERING

Prof. Alexandre Carlos profalexandre.jesus@fiap.com.br

Prof. Luís Carlos lsilva@fiap.com.br



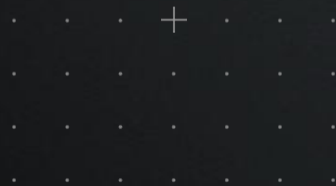
45697056



INTRODUÇÃO



React





O React é uma biblioteca Javascript criada pelo Facebook, utilizada para criar interfaces para usuários, com uma particularidade de renderizar somente a parte da tela que é necessária, aumentando assim em muito a performance da página.

Seu foco principal é transformar a experiencia do usuário mais eficiente, tornando a aplicação mais leve e performática, permitindo a reusabilidade de componentes.

No React tudo é Javascript, até os elementos HTML são criados por ele através do JSX, uma extensão de sintaxe que nos permite trazer a criação dos elementos HTML para dentro do Javascript.

Para que as mudanças dos componentes da tela sejam harmoniosas, ele utiliza o Virtual DOM (VDOM) que gerencia os componentes em memória e sincroniza com o DOM real, utilizando a biblioteca do ReactDOM. Isso aumenta a performance, melhorando até a classificação nos motores de busca.



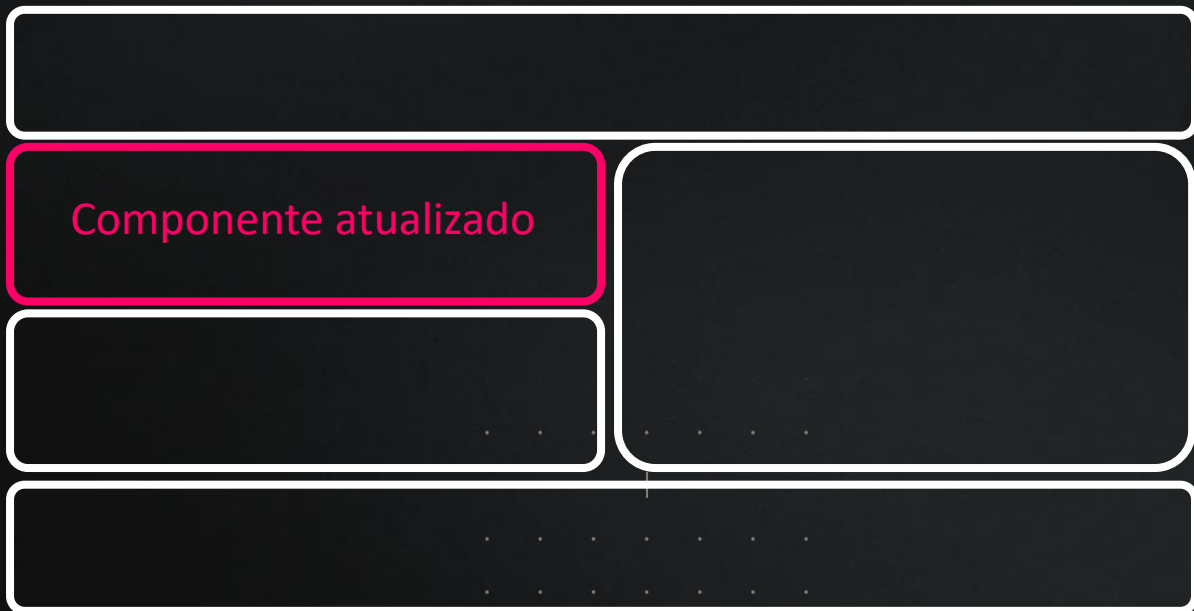


REACT

45697056



O React, por trabalhar com componentes, só precisa carregar a parte da página que foi alterada, mantendo as demais partes, deixando o trabalho mais rápido.



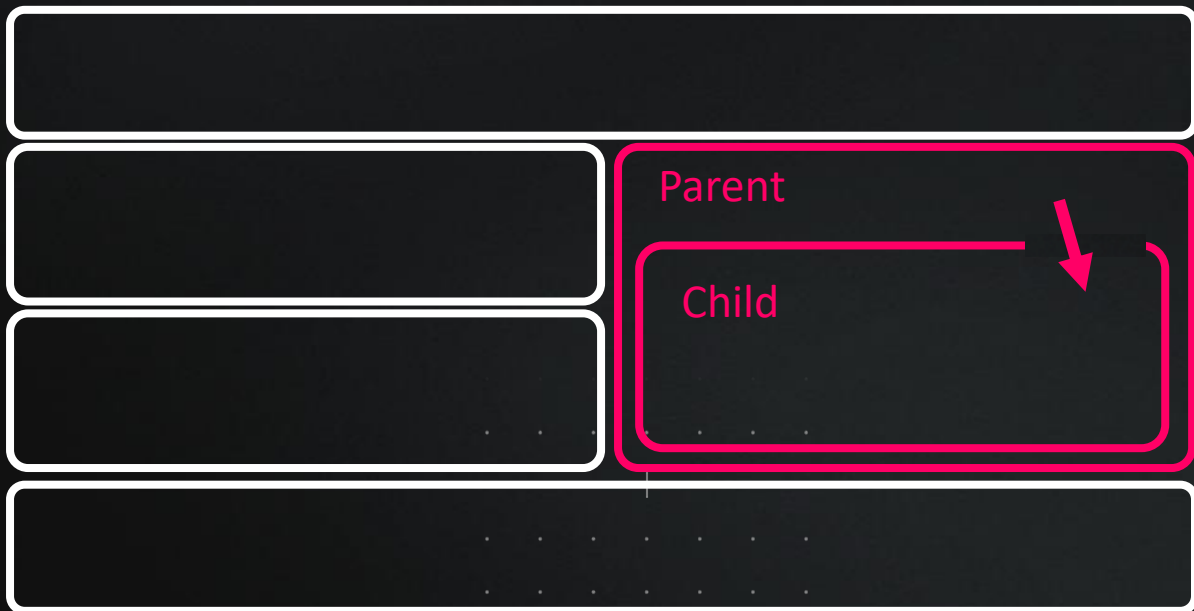


REACT

45697056



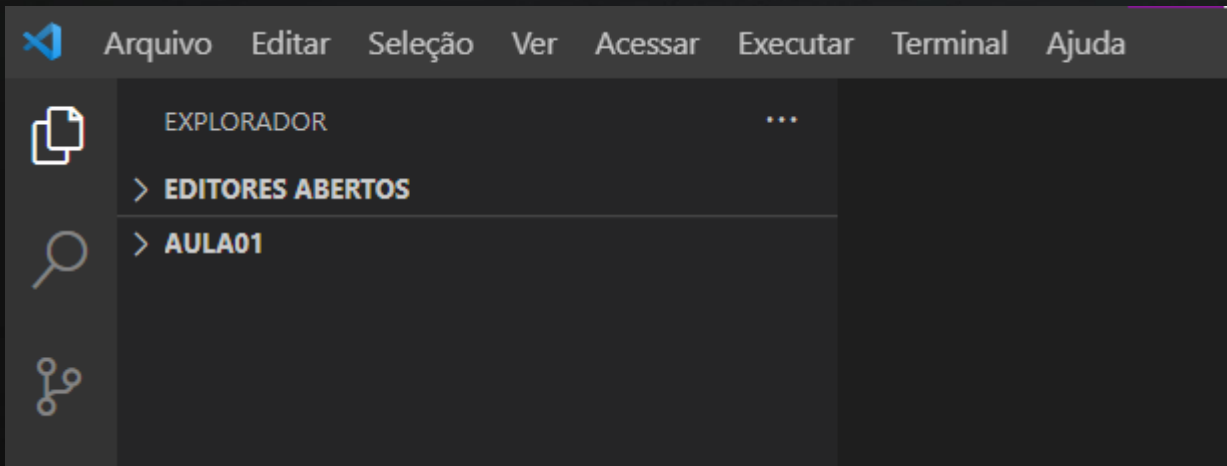
Ele trabalha com um fluxo unidirecional, em um único sentido, ou “One-way data flow”. As informações devem sempre vir do elemento pai para o elemento filho.





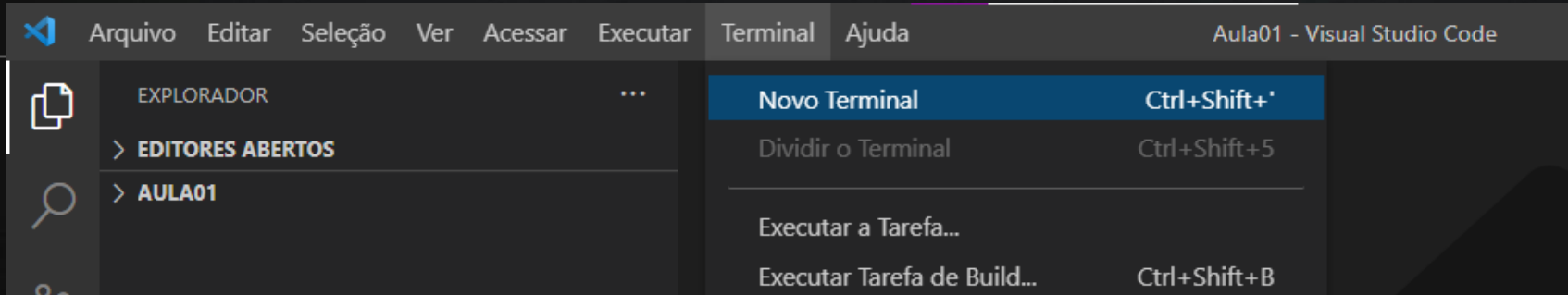
Na apostila passada, instalamos o Node.js, vamos usar ele para criar nossas aplicações diretamente do computador, sem estruturas online, ok?

- Vamos abrir o VS Code e criar uma pasta na área de trabalho, chame de Aula01.

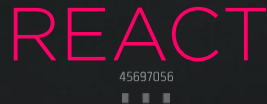




- Abra o terminal do VS Code, pode usar o atalho: **ctrl + shift + `** ou Menu Terminal -> Novo terminal.



- Obs.** Se quiser também pode usar o prompt de comando, basta acessar a nossa pasta Aula01.



- Agora vamos usar o npx, que é um executor de pacotes para criar nossa primeira aplicação. Digite o seguinte comando no prompt do terminal:

npx create vite@latest .

Se for a primeira vez, ele pode pedir permissão para fazer a instalação de pacotes.

```
PS C:\Workspace\intro-react> npx create vite@latest .  
Need to install the following packages:  
create-vite@5.5.2  
Ok to proceed? (y) ☐
```

É só por Y + enter e dar continuidade. Nas próximas ele não pedirá mais.



- Em seguida ele vai perguntar qual framework vamos utilizar.

```
? Select a framework: » - Use arrow-keys. Return to submit.  
  Vanilla  
  Vue  
> React  
  Preact  
  Lit  
  Svelte  
  Solid  
  Qwik  
  Others
```

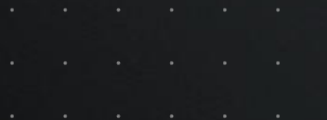
- Escolha REACT





- Agora vamos escolher JavaScript para usar em nosso projeto .

```
PS C:\WorkSpace\intro-react> npm create vite@latest .
Need to install the following packages:
create-vite@5.5.2
Ok to proceed? (y) y
✓ Select a framework: » React
? Select a variant: » - Use arrow-keys. Return to submit.
  TypeScript
  TypeScript + SWC
> JavaScript
  JavaScript + SWC
  Remix ↗
```





REACT

45697056



- Ao término da criação da estrutura, se for concluída com sucesso, devemos receber uma mensagem sugerindo que instalemos o pacote das bibliotecas, o `node_modules`.

```
Scaffolding project in C:\Workspace\intro-react...
```

```
Done. Now run:
```

```
npm install
```

```
npm run dev
```

```
npm notice
```

```
npm notice New minor version of npm available! 10.2.4 -> 10.8.2
```

```
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.8.2
```

```
npm notice Run npm install -g npm@10.8.2 to update!
```

```
npm notice
```

```
PS C:\Workspace\intro-react> 
```

- Digite no terminal a instrução **npm install**

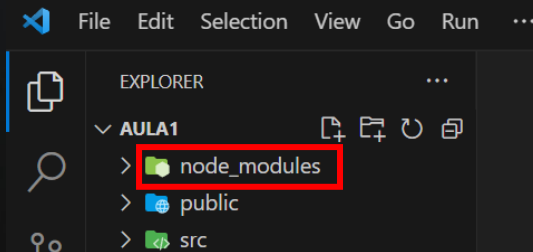


REACT

45697056



- Repare que agora temos uma nova pasta no projeto a `node_modules`, ela irá guardar todas as bibliotecas que iremos utilizar em nossos projetos.





REACT

45697056

■ ■ ■

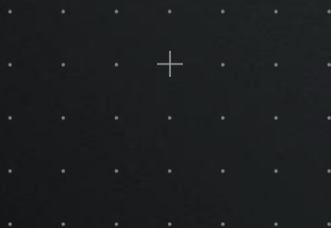
- Agora vamos subir nossa aplicação, digite `npm run dev` [enter] para subir nossa aplicação e em seguida ela irá gerar um endereço interno e uma porta, **segure a tecla Ctrl e clique no endereço**.

VITE v4.4.9 ready in **969 ms**

→ **Local:** `http://127.0.0.1:5173/`

→ **Network:** use `--host` to expose

→ press **h** to show help



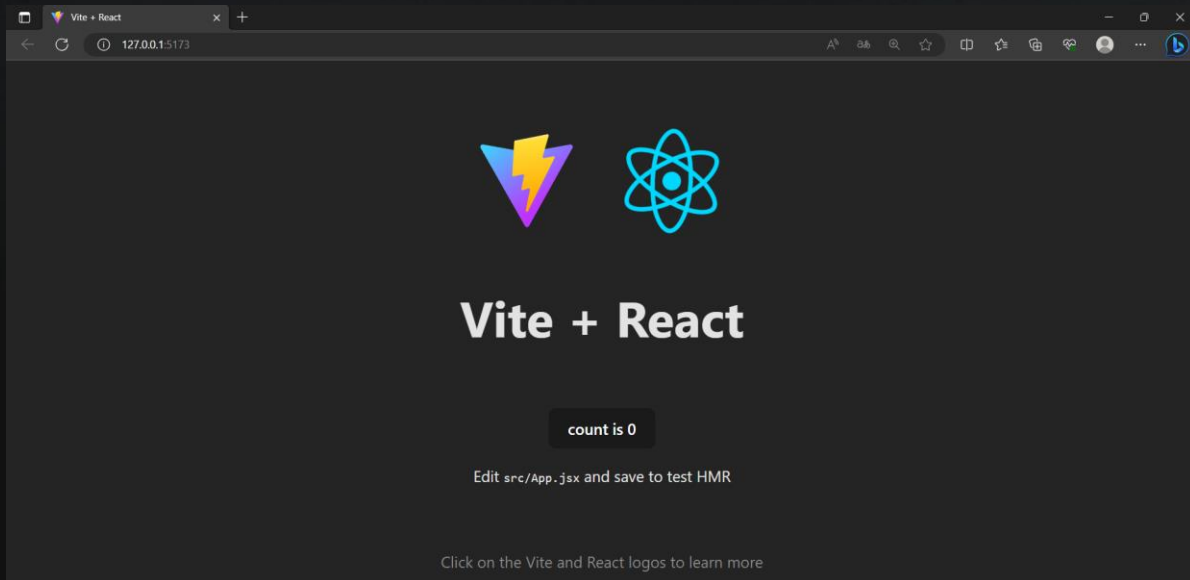


REACT

45697056



- Ele vai abrir a aplicação no seu navegador padrão.



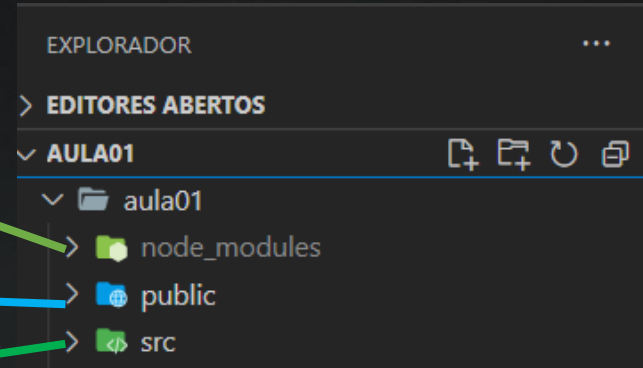


- Na pasta do nosso projeto temos 3 subpastas que foram criadas com o projeto:

Node Modules: são bibliotecas do Node que estão disponível para uso na aplicação

Public: contém os arquivos que são carregados no lado do cliente

Src: contém os arquivos que estão do lado do servidor





- O React é carregado na página index.html, dentro da div com id “root” e é a partir daí que o javascript carrega todos os componentes da aplicação.

.eslintrc.js

.gitignore

index.html

package-lock.json

package.json

README.md

```
8 </head>
9 <body>
10   <div id="root"></div>
11   <script type="module" src="/src/main.jsx"></script>
12 </body>
13 </html>
```

Arquivo index.html na pasta public

Div “root”, onde o javascript carrega toda a aplicação



REACT

45697056



- Já na pasta SRC, o main.js é o responsável por carregar toda a nossa aplicação na div de id “root” que está no index.html. Repare que ele está carregando o app.js nele.

```
ReactDOM.createRoot(document.getElementById('root')).render(  
  <React.StrictMode>  
    <App />  
  </React.StrictMode>,  
)
```





REACT

45697056

■ ■ ■

- O arquivo `app.js` é nosso arquivo que está dando origem ao componente principal da aplicação, repare que todos os componentes da tela estão sendo criados nele dentro de uma função que recebe o seu nome. Esta função retorna todos os elementos da tela que serão criados no html.
- Isso é possível por causa do jsx.



```
import { useState } from 'react'
import reactLogo from './assets/react.svg'
import viteLogo from '/vite.svg'
import './App.css'

function App() {
  const [count, setCount] = useState(0)

  return (
    <>
      <div>
        <a href="https://vitejs.dev" target="_blank">
          <img src={viteLogo} className="logo" alt="Vite logo" />
        </a>
        <a href="https://react.dev" target="_blank">
          <img src={reactLogo} className="logo react" alt="React logo" />
        </a>
      </div>
      <h1>Vite + React</h1>
      <div className="card">
        <button onClick={() => setCount((count) => count + 1)}>
          count is {count}
        </button>
        <p>
          Edit <code>src/App.jsx</code> and save to test HMR
        </p>
      </div>
      <p className="read-the-docs">
        Click on the Vite and React logos to learn more
      </p>
    </>
  )
}

export default App
```



REACT

45697056
■ ■ ■

Vamos limpar o conteúdo do arquivo app.js e vamos criar um novo conteúdo do zero.

```
function App() { // função com o nome do arquivo

  return ( // a função deve retornar o conteúdo que será exibido
    <>
    |   <h1>"Olá Mundo!"</h1>
    |
    </>
  )
}

export default App // instrução para chamar o componente
```

```
▼ <body>
  ▼ <div id="root"> == $0
    <h1>"Olá Mundo!"</h1>
  </div>
  <script type="module" src="/src/main.js" </script>
  </body>
</html>
```

"Olá Mundo!"



REACT

45697056



Podemos ter tantos componentes quanto necessário, mas todos eles devem estar dentro de um componente principal, senão teremos erro no código. Este elemento principal pode ser uma tag como uma DIV, por exemplo, ou uma tag vazia simplesmente para marcar o conteúdo, como temos abaixo:

```
export default function App(){  
  return(  
    <>  
    <h1>TDS FIAP - RWD </h1>  
    <a href="#">FIAP</a>  
    <br/>  
    <a href="#">GOOGLE</a>  
  </>  
  )  
}
```

Podemos fazer a exportação da função na mesma linha

Tags self close precisam de barra de fechamento "obrigatório"





REACT

45697056



Todas as expressões javascript realizadas dentro do jsx devem ser feitas dentro de chaves “{}”, para serem aceitas

```
export default function App(){
```

```
  const aluno = 'Matheus Ramalho'
```

```
  const curso = 'ADS'
```

```
  return(
```

```
    <>
```

```
    <h1>TDS FIAP - RWD </h1>
```

```
    <p>Alunos: {aluno}</p>
```

```
    <p>Curso: {curso}</p>
```

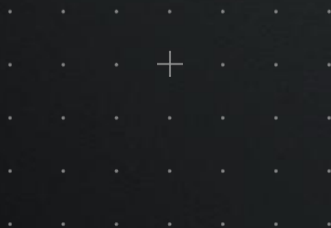
```
  </>
```

```
)
```

```
}
```

Código javascript da função antes do return

Valores expressos pelas constantes entre chaves





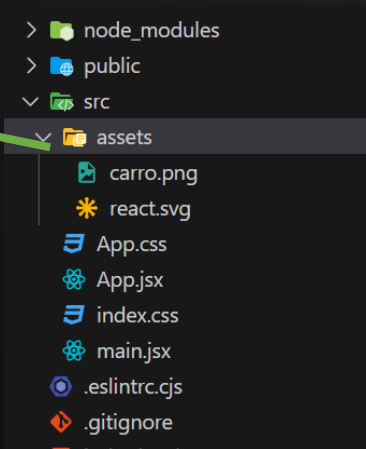
REACT

45697056

■ ■ ■

Para inserir elementos como imagens devemos importar antes de inserir no componente, lembrando sempre se usar as chaves:

Imagem salva na pasta src/assets



```
2 import carro from './assets/carro.png'
3
4
5 export default function App() {
6
7   const aluno = "Matheus Ramalho"
8   const curso = 'ADS'
9
10  return (
11    <>
12      <h1>TDS FIAP - RDW</h1>
13      <img src={carro} alt="um carro" />
14      <p>Aluno: {aluno}</p>
15      <p>Curso: {curso}</p>
```

Import da imagem

Inserindo a imagem na tag img

OBS. Se a imagem estiver na pasta public, o mapeamento será do modo convencional



REACT

45697056

■ ■ ■

Como falamos a pouco, o React trabalha com componentes, então vamos criar alguns para ver como funciona. Dentro da pasta src crie uma pasta chamada components e dentro dela um arquivo chamado ListaAluno.jsx. Nela insira o seguinte código:

```
export default function ListaAlunos(){  
  
  return(  
    <ul>  
      <li>Huguinho</li>  
      <li>Zézinho</li>  
      <li>Luizinho</li>  
    </ul>  
  )  
}
```




REACT

45697056



Agora para conseguirmos utilizar o arquivo ListaAlunos.js como um componente devemos importa-lo dentro de App.js e inserir ele no componente principal, conforme abaixo:

```
import carro from './assets/carro.png'  
import ListaAlunos from './components/ListaAlunos'
```

Importando o
arquivo
ListaAlunos.jsx

```
export default function App() {  
  
  const aluno = "Matheus Ramalho"  
  const curso = 'ADS'  
  
  return (  
    <>  
      <h1>TDS FIAP - RDW</h1>  
      <img src={carro} alt="um carro" />  
      <p>Aluno: {aluno}</p>  
      <p>Curso: {curso}</p>  
      <ListaAlunos/>  
    </>  
  )  
}
```

Inserindo no componente
principal, criando uma tag com
seu nome



Exercício

45697056



1. Crie uma nova aplicação chamada exercício;
2. Limpe o conteúdo do arquivo App.jsx
3. Crie um componente chamado Cabecalho.jsx e insira uma tag header com um h1 e um parágrafo;
4. Crie um componente chamado Carros.jsx e insira uma imagem de carro e uma lista com 5 modelos de carro.
5. Crie um componente chamado Parceiros.jsx e insira um h2 e 4 links.





REACT

Como comentamos no início os dados, se necessário, devem fluir do pai para seus filhos, este é o fluxo natural. Para conseguirmos passar estes dados vamos utilizar 'props'. Ele é passado do pai para seus filhos como um objeto e dentro do filho podemos acessar seus valores.

Para enviar os valores para o filho devemos criar atributos em sua tag que está no componente pai. Vamos enviar os nomes dos alunos daquela lista que criamos a pouco:

```
export default function App() {  
  
  const aluno = "Matheus Ramalho"  
  const curso = 'ADS'  
  const alunos = ['Luis', 'Alexandre', 'Wellington']  
  
  return (  
    <>  
      <h1>TDS FIAP - RDW</h1>  
      <img src={carro} alt="um carro" />  
      <p>Aluno: {aluno}</p>  
      <p>Curso: {curso}</p>  
      <ListaAlunos alunos={alunos}/>  
    </>  
  )  
}
```

Dados contidos no pai

Enviando os dados para o filho



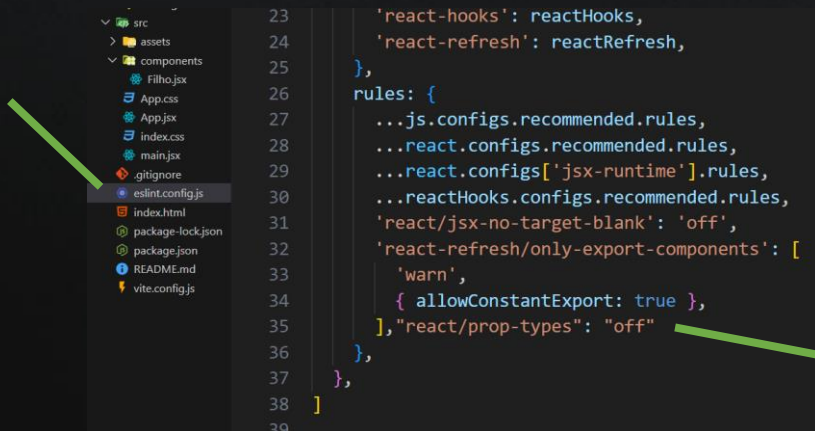
REACT

45697056



Como ainda não estamos utilizando o Typescript e o Eslint tem a função de indicar a falta de padronização e erros no código, teremos uma indicação de erro no props. Para resolvermos esta indicação de erro enquanto usamos o Javascript, coloque a instrução abaixo, assim não aparecerá mais o erro.

Abra o arquivo
eslint.config.js



```
23   'react-hooks': reactHooks,  
24   'react-refresh': reactRefresh,  
25 },  
26 rules: {  
27   ...js.configs.recommended.rules,  
28   ...react.configs.recommended.rules,  
29   ...react.configs['jsx-runtime'].rules,  
30   ...reactHooks.configs.recommended.rules,  
31   'react/jsx-no-target-blank': 'off',  
32   'react-refresh/only-export-components': [  
33     'warn',  
34     { allowConstantExport: true },  
35   ],  
36   'react/prop-types': 'off'  
37 },  
38 ],  
39 }
```

Insira a instrução:
"react/prop-types": "off" no
final do rules



REACT

45697056

■ ■ ■

Para o filho poder utilizar os dados enviados pelo pai devemos colocar o parâmetro props na função ListaAlunos e, como um objeto, chamar seus valores.

```
export default function ListaAlunos(props){  
  return(  
    <ul>  
      <li>{props.alunos[0]}</li>  
      <li>{props.alunos[1]}</li>  
      <li>{props.alunos[2]}</li>  
    </ul>  
  )  
}
```

Parâmetro props

Utilização dos valores
enviados pelo pai



REACT

45697056



Assim como passamos dados do pai para o filho, também podemos passar funções, vamos enviar uma função para ver as diferenças. Crie a função no componente pai conforme abaixo:

```
export default function App() {  
  
  const aluno = "Matheus Ramalho"  
  const curso = 'ADS'  
  const alunos = ['Luis', 'Alexandre', 'Wellington']  
  
  const novoAluno = () => 'Rafael'  
  
  return (  
    <>  
      <h1>TDS FIAP - RDW</h1>  
      <img src={carro} alt="um carro" />  
      <p>Aluno: {aluno}</p>  
      <p>Curso: {curso}</p>  
      <ListaAlunos alunos={alunos} novo={novoAluno}/>  
    </>  
  )  
}
```

Criação de uma função

Enviando a função
para o filho



REACT

45697056

■ ■ ■

Para a utilização da função herdada do pai só não devemos esquecer dos parênteses, o resto é igual as variáveis:

```
export default function ListaAlunos(props){  
  return(  
    <ul>  
      <li>{props.alunos[0]}</li>  
      <li>{props.alunos[1]}</li>  
      <li>{props.alunos[2]}</li>  
      <li>{props.novo()}</li>  
    </ul>  
  )  
}
```

Utilizando a função



Exercício

45697056



1. Volte para a aplicação exercício que criamos a pouco;
2. Passe os modelos dos carros do componente pai para o componente filho.
3. Crie uma função multiplicando dois números no componente pai e passe para o componente Parceiros, executando em uma span logo abaixo do h2;



DUVIDAS





Copyright © 2015 - 2021 Prof. Luís Carlos S. Silva
Prof. Alexandre Carlos de Jesus

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).