



FIAP

GRADUAÇÃO

45697056



TDS

Responsive Web Development

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

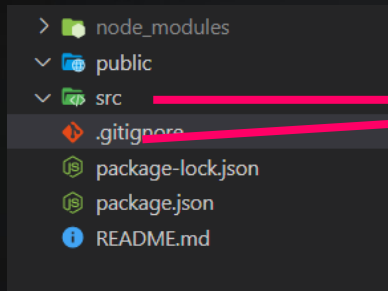
Prof. Luís Carlos lsilva@fiap.com.br



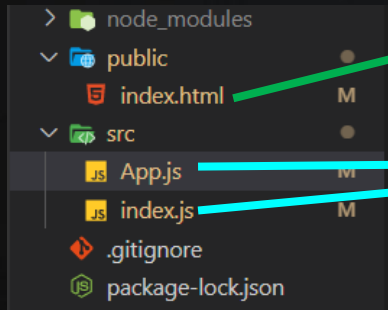


Novo projeto c/ pasta src e public do zero

Para iniciarmos nossos projeto da aula de hoje, vamos criar um novo projeto chamado **react-aula7** e vamos **apagar** todos os arquivos das pastas **src** e **public**.



Apagamos todos os arquivos da pasta src e public



Logo após, na pasta public, recrie os arquivos index.html contendo a div c/ id='root'

E na pasta src os arquivos index.js com o ReactDOM.render() e o arquivo App.js como nosso componente principal.



Instalando bibliotecas

45697056

■ ■ ■

Para esta aula vamos precisar de duas bibliotecas, a do **styled-components** e **react-router-dom** que já conhecemos na aula passada e a do **react-icons** que vamos conhecer hoje. Vamos instalar as 3 em nosso projeto, se você já inicializou ele com o npm start, abra um outro terminal para executar as instalações (ctrl +shift + ') no VSCode.

- * Primeiro vamos instalar o styled-components:

```
npm install --save styled-components
```

- * Em segundo, vamos instalar o react-router-dom:

```
npm install --save react-router-dom
```

- * Em segundo, vamos instalar o react-icons:

```
npm install react-icons --save
```



Projeto com algumas novidades

45697056
■ ■ ■

Na aula de hoje vamos consolidar os conhecimentos já adquiridos e conhecer algumas novas características do React.

A primeira coisa que vamos nos ater é que vamos começar a dar nome para as funções dos componentes, para o React não dar mais mensagens no console. Outra mudança é que só vamos importar o react no index.js, veja o exemplo de início do App.js



```
src > .js App.js > [e] default
1 | const App = ()=>{
2 |
3 |     return(
4 |         <>
5 |         <h1>Olá</h1>
6 |         </>
7 |     )
8 | }
9 | export default App;
```



Projeto com algumas novidades

45697056
■ ■ ■

Um bom lugar para dar um reset no css é no index, todos os nossos componentes passam por lá.
Crie um arquivo index.css.

```
src >  index.css > ...  
1 |  *{  
2 |     margin: 0;  
3 |     padding: 0;  
4 |     box-sizing: border-box;  
5 |     font-family: Arial, Helvetica, sans-serif;  
6 | }
```






Projeto com algumas novidades

45697056
■ ■ ■

A ideia é criar uma aplicação para criar pequenas anotações de tarefas. Vamos começar criando dentro da pasta **components** um Componente chamado **ListaTarefas.jsx**. Por enquanto ela vai ficar só com um parágrafo de teste.

```
src > components >  ListaTarefas.jsx > ...  
1  ∨ const ListaTarefas = ()=>{  
2  
3      return(  
4  ∨      <div>  
5          <p>Lista de Tarefas</p>  
6      </div>  
7  )  
8  }  
9  export default ListaTarefas;
```

.
.



React

Teremos componentes que servirão como post-it de lembretes de tarefa. Crie um componente chamado `Tarefa.jsx`.

Projeto com algumas novidades

```
src > components > Tarefa.jsx > ...
1  import styled from 'styled-components'
2
3  const DivTarefa = styled.div`
4    background-color: #ffb;
5    border: 2px solid #333;
6    box-shadow: 5px 5px 5px #333;
7    padding: 20px;
8    text-align:center;
9    width: 350px;
10   height:200px;
11   margin:10px;
12   h2,p{ padding-bottom: 10px; }
13 `
14 const Tarefa = ()=>{
15   return(
16     <DivTarefa>
17       <h2>Relatório de vendas</h2>
18       <p>Para: Departamento de Vendas</p>
19       <p>Levantar os valores das vendas deste mês.</p>
20     </DivTarefa>
21   )
22 }
23 export default Tarefa;
```




React

Voltando para o `ListaTarefas.jsx`, vamos organizar para receber várias tarefas. Vamos testar com 3 tarefas para começar.

Projeto com algumas novidades

```
src > components > ListaTarefas.jsx > ...  
1   import styled from 'styled-components'  
2   import Tarefa from "../Tarefa";  
3  
4   const DivLista = styled.div`  
5     width: 100%;  
6     min-height: 85vh;  
7     padding: 20px;  
8     background-color: #ccc;  
9     display: flex; flex-wrap: wrap;  
10    justify-content: space-around;  
11  `;  
12  const ListaTarefas = ()=>{  
13  
14    return(  
15      <DivLista>  
16        <Tarefa/>  
17        <Tarefa/>  
18        <Tarefa/>  
19      </DivLista>  
20    )  
21  }  
22  export default ListaTarefas;  
23
```



React

A ideia é criarmos novas tarefas, então vamos criar um state em `ListaTarefas` para guardar os dados das tarefas. Vamos criar o state e um map para carregar a tela.

Projeto com algumas novidades

45897056

src > components > ListaTarefas.jsx > ...

```
13  const ListaTarefas = ()=>{
14
15    const [tarefa,setTarefa]= useState([
16      {
17        titulo:'Lista de Pagamento',
18        setor: 'Dep. Vendas',
19        descricao: 'Levantar os valores das vendas',
20      },
21      {
22        titulo:'Planilha de Salários',
23        setor: 'Dep. Pessoal',
24        descricao: 'Gerar planilhas',
25      },
26    ])
27    return(
28      <DivLista>
29        {tarefa.map((tar,i)=>{
30          <Tarefa
31            key={i}
32            titulo={tar.titulo}
33            setor={tar.setor}
34            descricao={tar.descricao}
35          />
36        ))}
37      </DivLista>
38    )
39  }
```



React

Projeto com algumas novidades

Para funcionar precisamos receber os props em `Tarefa` para carregar os dados na tela.

```
13
14  ✓ const Tarefa = props=>{
15      return(
16  ✓      <DivTarefa>
17          <h2>{props.titulo}</h2>
18          <p>Para: {props.setor}</p>
19          <p>{props.descricao}</p>
20      </DivTarefa>
21  )
22  }
23  export default Tarefa;
```



React

Começaremos criando uma função em `ListaTarefas`, para inserir uma tarefa nova, por enquanto só para validar a função.

Projeto com algumas novidades

```
45697056
■ ■ ■

27  ✓   const addTarefa= ()=>{
28  ✓       const novaTarefa = {
29           titulo:'Planilha de Salários',
30           setor: 'Dep. Pessoal',
31           descricao: 'Gerar planilhas',
32       }
33
34       setTarefa([...tarefas, novaTarefa])
35   }
36   return(
37   ✓       <DivLista>
38   ✓           {tarefas.map((tar,i)=>{
39               <Tarefa
40   ✓               key={i}
41                   titulo={tar.titulo}
42                   setor={tar.setor}
43                   descricao={tar.descricao}
44               />
45           )}}
46       <button onClick={addTarefa}>Adicionar</button>
47   </DivLista>
48   )
```



React

Criaremos um formulário para inserir novas tarefas, crie um componente chamado `FormTarefas.jsx`.

Projeto com algumas novidades

```
src > components > FormTarefas.jsx > ...  
1  const FormTarefas= props=>{  
2  
3      return(  
4          <div>  
5              <form method="post" onSubmit={props.addTarefa}>  
6                  <div>  
7                      <input name="titulo" placeholder='Título'  
8                          value={props.tarefa.titulo}  
9                          onChange={props.digit}/>  
10                 </div>  
11                 <div>  
12                     <input name="setor" placeholder='Setor'  
13                         value={props.tarefa.setor}  
14                         onChange={props.digit}/>  
15                 </div>  
16                 <div>  
17                     <textarea name="descricao" placeholder='Descrição'  
18                         value={props.tarefa.descricao}  
19                         onChange={props.digit}/>  
20                 </div>  
21                 <button type="submit">Adicionar</button>  
22             </form>  
23         </div>  
24     )  
25 }  
26 export default FormTarefas;
```



React

Para o formulário funcionar temos que criar um state para nova tarefa, mudar nossa função de adicionar tarefa, criar um para captura das informações do formulário e passar tudo por props.

Projeto com algumas novidades

```
27
28 const [nTarefa, setNTarefa] = useState({
29   titulo:'', setor:'', descricao: '',})
30
31 const addTarefa= e=>{
32   e.preventDefault()
33   setTarefa([...tarefa, nTarefa])
34   setNTarefa({titulo:'', setor:'', descricao: '',})
35 }
36
37 const captura= e=>{
38   const {value, name} = e.target
39
40   if(name === 'titulo'){
41     setNTarefa({'titulo':value, 'setor':nTarefa.setor, 'descricao':nTarefa.descricao})
42   }else if(name === 'setor'){
43     setNTarefa({'titulo':nTarefa.titulo, 'setor':value, 'descricao':nTarefa.descricao})
44   }else if(name === 'descricao'){
45     setNTarefa({'titulo':nTarefa.titulo, 'setor':nTarefa.setor, 'descricao':value})
46   }
47 }
48
49 return(
50   <DivLista>
51     <FormTarefas addTarefa={addTarefa} tarefa={nTarefa}
52       digit={captura}/>
```



Projeto com algumas novidades

Agora um pouco de estilização no formulário, só para o vermos melhor.

```
src > components > FormTarefas.jsx > FormTarefas
1  import styled from 'styled-components'
2
3  const DivForm = styled.div`
4    width:100%;
5    padding:20px;
6    background-color:#993;
7
8    form{width:50%; margin:auto;}
9    form input, form textarea{padding:5px; width:80%;}
10   form textarea{height: 80px;}
11   form button{padding:10px; background: #3c3; color:#fff;}
12
13
14  const FormTarefas= props=>{
```



React

Vamos fazer uma função para remover a tarefa. Em `ListaTarefas`, crie a função `remover tarefa` e passe como propriedade pelo props.

Estamos usando o método `Bind` para ele conseguir referenciar o componente a ser excluído.

Projeto com algumas novidades

45697056

```
49   const removerTarefa = tar=>{
50     let lista = tarefa
51     lista = lista.filter((t)=> t!== tar)
52     setTarefa(lista)
53   }
54
55   return(
56     <DivLista>
57       <FormTarefas addTarefa={addTarefa} tarefa={nTarefa}
58         digit={captura}/>
59       {tarefa.map((tar,i)=>(
60         <Tarefa
61           key={i}
62           titulo={tar.titulo}
63           setor={tar.setor}
64           descricao={tar.descricao}
65           remove={removerTarefa.bind(this,tar)}
66         />
67       ))}
68     </DivLista>
69   )
```




Projeto com algumas novidades

45697056
■ ■ ■

Temos que criar um botão para usar a função no componente **Tarefa**.

```
const Tarefa = props=>{  
  return(  
    <DivTarefa>  
      <h2>{props.titulo}</h2>  
      <p>Para: {props.setor}</p>  
      <p>{props.descricao}</p>  
      <button onClick={props.remove}>X</button>  
    </DivTarefa>  
  )  
}  
export default Tarefa;
```

.
.
.



Projeto com algumas novidades

Vamos usar agora a biblioteca de ícones que instalamos no início. O react-icons junta várias libs de ícones e nos deixa usá-los como componentes. Escolha os ícones no endereço:

<https://react-icons.github.io/react-icons>

Escolha o ícone e insira conforme ao lado.

```
src > components > Tarefa.jsx > DivTarefa
1 import styled from 'styled-components'
2 import { FaTrashAlt as X } from "react-icons/fa";
3
4 const DivTarefa = styled.div`
5   background-color: #ffb;
6   border: 2px solid #333;
7   box-shadow: 5px 5px 5px #333;
8   padding: 20px;
9   text-align:center;
10  width: 350px;
11  height:200px;
12  margin:10px;
13  position:relative;
14  h2,p{ padding-bottom: 10px; }
15  button{position:absolute; top:10px; right:10px}
16
17 const Tarefa = props=>{
18   return(
19     <DivTarefa>
20       <h2>{props.titulo}</h2>
21       <p>Para: {props.setor}</p>
22       <p>{props.descricao}</p>
23       <button onClick={props.remove}><X></button>
24     </DivTarefa>
```



Projeto com algumas novidades

Agora vamos criar uma página principal e um menu para nosso projeto. Vamos começar com a página inicial, crie um componente chamado `Home.jsx`.

```
import styled from "styled-components"
```

```
const DivHome = styled.div`  
  width:100%;  
  height:85vh;  
  background: #bbf;  
  h1, p{padding:20px;}
```

```
const Home = ()=>{  
  
  return(  
    <div>  
      <h1>Página principal do projeto</h1>  
      <p>Projeto para controle de Tarefas.</p>  
    </div>  
  )  
}
```



React

Crie um componente chamado
Menu.jsx.

Projeto com algumas novidades

```
src > components > Menu.jsx > Menu
1  import { Link } from "react-router-dom"
2  import styled from "styled-components"
3
4  const DivMenu = styled.nav`
5    background: #999; height:15vh;
6    display:flex; justify-content: center;
7
8    ul{display:flex;}
9
10   ul li{list-style: none;}
11   ul li a{ display: inline-block; text-decoration: none;
12     padding: 20px; color: #fff; font-size:1.5em; }
13
14  const Menu = ()=>{
15
16    return(
17      <DivMenu>
18        <ul>
19          <li><Link to='/'>Home</Link></li>
20          <li><Link to='/tarefa'>Home</Link></li>
21        </ul>
22      </DivMenu>
23    )
24  }
```



React

Agora vamos montar as rotas
no `App.js`.

Projeto com algumas novidades

45697056

rc > `JS` App.js > ...

```
1  import { BrowserRouter, Switch, Route } from "react-router-dom";
2  import ListaTarefas from "../components/ListaTarefas";
3  import Menu from "../components/Menu";
4  import Home from "../components/Home";
5
6
7  const App = () => {
8
9      return(
10         <BrowserRouter>
11           <Menu/>
12           <Switch>
13             <Route path="/" exact component={Home}/>
14             <Route path="/tarefa" component={ListaTarefas}/>
15           </Switch>
16         </BrowserRouter>
17       )
18     }
19     export default App;
```



Exercício

45697056



Crie um projeto React chamado `exercicio7`, limpe todos os arquivos das pastas `public` e `src`, após isso recrie o arquivo `index.html` na pasta `public` e os arquivos `index.js` e `App.js` na pasta `src`.

Faça uma aplicação para uma lanchonete, onde você tenha uma tela e possa criar os pedidos das mesas, contendo Número, Garçom e Itens.

Ao atender cada mesa deve ser possível excluir a anotação do quadro.

Deve ser utilizado `Styled-Components` e `React-Icons` para estilização do exercício.



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).