

Aula 07 - Oi Devs

1. Sugestões de temas (30, 15, 5)
 - a. Curiosidade do frontend
 - b. Mulher na tecnologia
 - c. Cultura pop
 - d. Ecossistema OI

1. Ao digitar um usuário e senha inválidos troque a cor da label do input para vermelho.

```
useState()  
// inputColor, setInputColor  
// setInputColor("red")
```

2. Exiba uma mensagem utilizando o nosso `<Subtitle />` para informar ao usuário que ele digitou suas credenciais inválidas => Renderização Condicional

Aula 06 - Oi Devs

- ✓ ~~Stack: React, JavaScript (Componentes Funcionais com ES6)~~
- ✓ ~~CRA~~
- ✓ ~~Modelagem baseada em componentes~~
- ✓ ~~JSX (doc oficial)~~
- ✓ ~~ES6~~
- ✓ ~~React Router~~
- ✓ ~~Props~~
- ✓ ~~Arrow Function~~
- ✓ ~~Proptypes~~
- ✓ ~~React State~~
- ✓ ~~State Lifting~~
- ✓ ~~Renderização Condicional~~
- ✓ ~~Binds (class components)~~

Renderização de Listas

Função de Callback

Em JavaScript, o jeito de criar uma função de callback é passá-la como um parâmetro para outra função, chamando-a novamente em seguida, logo depois

que algo aconteça ou que alguma tarefa seja concluída.

Em JavaScript, funções são objetos.

Podemos passar objetos para funções como parâmetros? Sim.

Então, nós podemos também passar funções como parâmetros para outras funções e chamá-las dentro das funções externas. Parece complicado? Vamos ver isso no exemplo abaixo:

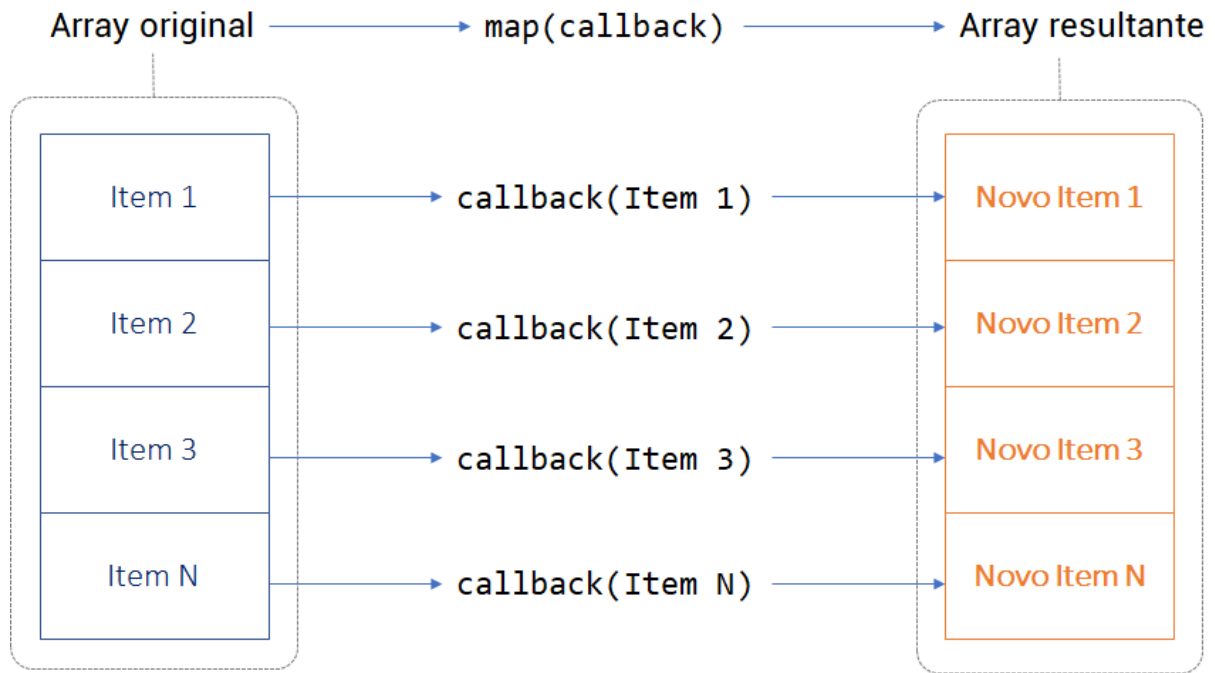
```
1 const Input = (onClick) => {  
2   onClick();  
3 }
```

map()

Dado o código abaixo, nós usamos a função `map()` para receber um array de números e dobrar o valor de cada um deles. Atribuímos o novo array retornado pela função `map()` para a variável `doubled` e imprime no console:

```
1 const numbers = [1, 2, 3, 4, 5];  
2 const doubled = numbers.map((number) => number * 2);  
3 console.log(doubled);
```

O **método map()** nos permite percorrer um vetor e obter um novo array cujos itens são o resultado de uma função de callback que recebe como parâmetro cada item original.



```

1  const alunos = [{
2      nomeDoAluno: "Adriana",
3      cpf: "13456713890",
4      nota: 0
5  }, {
6      nomeDoAluno: "Jady",
7      cpf: "13456813890",
8      nota: 0
9  }
10 ]
11
12 const ListaDeAlunos = ({ alunos }) => {
13     const atribuirNota = (cpf, nota) => {
14         const alunoNotaAlterada = alunos.find(
15             (aluno) => aluno.cpf === cpf && aluno.nota = nota
16         )
17     }
18
19     return (
20         <ul>

```

```
21     {alunos.map((aluno) => (  
22         <li key={aluno.cpf} onClick={() => atribuirNota(aluno.  
cpf, 100)}>  
23             {aluno.nomeDoAluno}  
24         </li>  
25     ))}  
26 </ul>  
27 );  
28 }
```

```
1  const NumberList = ({ numbers }) => {  
2      const listItems = numbers.map((number) => (  
3          <ListItem key={number.toString()} value={number} />  
4      ));  
5      return <ul>{listItems}</ul>;  
6  };
```

```
1  const NumberList = ({ numbers }) => {  
2      return (  
3          <ul>  
4              {numbers.map((number) => (  
5                  <ListItem key={number.toString()} value={number} />  
6              ))}  
7          </ul>  
8      );  
9  };
```

```
const users = [
  { id: "1", firstName: "Robin", lastName: "Wieruch" },
  { id: "2", firstName: "Dennis", lastName: "Wieruch" },
];

const App = () => {
  return (
    <div>
      <h1>Hello Conditional Rendering</h1>
      <List list={users} />
    </div>
  );
};

const List = ({ list }) => {
  if (!list) {
    return null;
  }

  return (
    <ul>
      {list.map((item) => (
        <Item key={item.id} item={item} />
      ))}
    </ul>
  );
};

const Item = ({ item }) => {
  return (
    <li>
      {item.firstName} {item.lastName}
    </li>
  );
};

export default App;
```

☐ Eventos

Bem como os eventos HTML da DOM, React pode realizar ações baseadas em eventos do usuário

React tem os mesmos eventos do HTML: click, change, mouseover, etc.

Os manipuladores de eventos determinam qual ação deve ser executada sempre que um evento é acionado.

Isso pode ser um clique num botão ou uma alteração no input de um texto.

Essencialmente, os manipuladores de eventos fazem com que os usuários interajam a aplicação React.

O tratamento de eventos com elementos React é semelhante ao tratamento de eventos em elementos DOM.

1. Renderize a nossa lista de usuários dentro da página de login.

- ☐ Formulários
- ☐ Composição de Componentes (children)
- ☐ Métodos do Ciclo de Vida