

```

import React from 'react';

class Produtos extends React.Component {
  constructor(props) {
    super(props);

    this.state = { produtos: [] };
  }

  handleAddClick() {
    const { produtos } = this.state;
    const novoProduto = { id: 0, descricao: 'Banana' };

    produtos.push(novoProduto);

    this.setState({ produtos });
  }

  render() {
    const { produtos } = this.state;

    return (
      <div>
        <ul>
          {produtos.map(p => <li>{p.descricao}</li>)}
        </ul>
        <button onClick={this.handleAddClick}></button>
      </div>
    );
  }
}

```

Foram encontrados dois problemas no código, o primeiro problema foi identificado na linha vermelha, **handleAddClick** é uma função que está sendo chamada no onClick do botão, o que causa o erro nessa função é que ela utiliza o `this.state`, mas dessa forma que ela está sendo declarada o `this` não é reconhecido, a função não faz parte do contexto do construtor, então para ela o `this.state` não foi declarado.

Para solucionar esse problema existe duas soluções que eu conheço, a primeira é referenciar essa função no construtor utilizando o `bind`:

```

this.handleAddClick = this.handleAddClick.bind(this)

```

O segundo modo, e mais utilizado atualmente, na minha opinião, é utilizar a arrow function na declaração da função, assim ela é declarada sem sair do contexto de `this`, substituindo a linha vermelha por:

```

handleAddClick = () => {

```

O segundo problema foi encontrado na linha amarela, ela não causa impacto a ponto da solução não rodar, mas para melhorar a performance do react é bom utilizar as `Keys` dentro de uma lista, quando é utilizado o `map` em react ele sempre espera um `key` única para cada registro da lista, para resolver esse aviso que o react vai mandar, basta editar a linha amarela adicionando a `key`:

```

produtos.map( p => <li key={p.id}> { p.descricao } </li>)}

```

Utilizando o `id` na `key`, pois é necessária uma chave única de cada registro.