

```
import React, { Component } from "react";
import { StyleSheet, Text, TextInput, View } from "react-native";

/*  PASSAGEM DE ESTADO POR ATRAVÉS DE COMPONENTE FEITO COM CLASSE:

    Quando trabalhamos com Componentes feitos com Classe, o método para guardar estados é diferente.
    Podemos usar duas formas para colher parâmetros e guardar estados, veja...

*/

// FORMA MAIS ELABORADA:
// Nessa forma, nós chamamos pelo construtor da classe, por através dele podemos acessar o atributo state, que é um
// atributo já pré-existente na classe e o responsável por guardar estados.
export class EstadoSemInput extends Component {

    //Veja que chamamos o construtor e pegamos os parâmetros à partir dele...
    constructor(props){
        super(props)

        //Note que chamamos pelo atributo "state", esse atributo sempre deve receber um objeto, que conterá uma
        //chave responsável por receber o parâmetro advindo da props. Note que automaticamente já fizemos uma mudança no
        //parâmetro logo de cara assim que chegou, o multiplicando por 7. Essa não é a forma convencional de mudar um valor
        //por estado, pois deveríamos usar o método setState, mas usaremos aqui só para exemplificar.
        this.state = {
            number: props.number * 7
        }
    }

    render() {
        return (
```

```
    <View style={styles.fundo}>
      <Text style={styles.texto}>
        Qualquer número passado via props será multiplicado por 7.
      </Text>
      <Text style={styles.texto}>
      </Text>
      <Text style={styles.texto}>
        {/* Note que usamos o this para referenciar a classe, o props para pegar o valor passado
        diretamente. Mas também usamos o state, para referenciar ao valor alterado por estado. */}
        O número passado foi {this.props.number} que vezes 7 é = {this.state.number}
      </Text>
    </View>
  )
}
}
```

// FORMA MAIS FÁCIL:

// Nessa forma, note que abolimos a chamada do construtor, simplesmente chamamos o state diretamente e atribuímos sobre ele o objeto state que vai receber o valor do props, veja abaixo...

```
export class EstadoComInput extends Component {
```

//Note que só chamamos o "state" atribuímos um objeto a ele onde a chave vai receber o parâmetro props referenciando a classe por através do this, muito mais simples...

```
    state = {
      numero: this.props.number
    }
```

//Aqui temos também um exemplo de como utilizar o setState.

//Note que o setState tem que ser referenciado pelo "this" e que para que ele seja invocado corretamente temos que criar uma arrow function para chamar ele.

//Mas por que uma arrow function?

//Acontece que o this quando usado diretamente sobre uma classe acaba não referenciando à ela diretamente, isso acontece por causa dos códigos internos que regem o React Native. Por isso, é necessário usar uma arrow function, visto que as arrow function sempre referenciam ao local onde a função é criada. Como a função neste caso está sendo criada dentro de uma classe, a arrow function faz com que o setState referencie a classe de fato.

```
    alteraValor = (valor) => {
      this.setState({numero: valor })
    }

    render() {
      return (
        <View style={styles.fundo}>
          <Text style={styles.texto}>
            Mude o número do valor atual, recebemos o valor de {this.state.numero}
          </Text>
          <Text style={styles.texto}>
            </Text>
          <TextInput
            keyboardType={"numeric"}
            style={{borderWidth: 1}}
            placeholder="Digite um número"
            value={this.state.numero}
            // Outra solução possível para o problema do problema do this, é usar uma arrow function
            // diretamente sobre o onChangeText também dá certo. Como no exemplo da linha abaixo...
            // onChangeText={valor => this.alteraValor(valor)}
            onChangeText={this.alteraValor}
          />
        </View>
      )
    }
  }
}
```

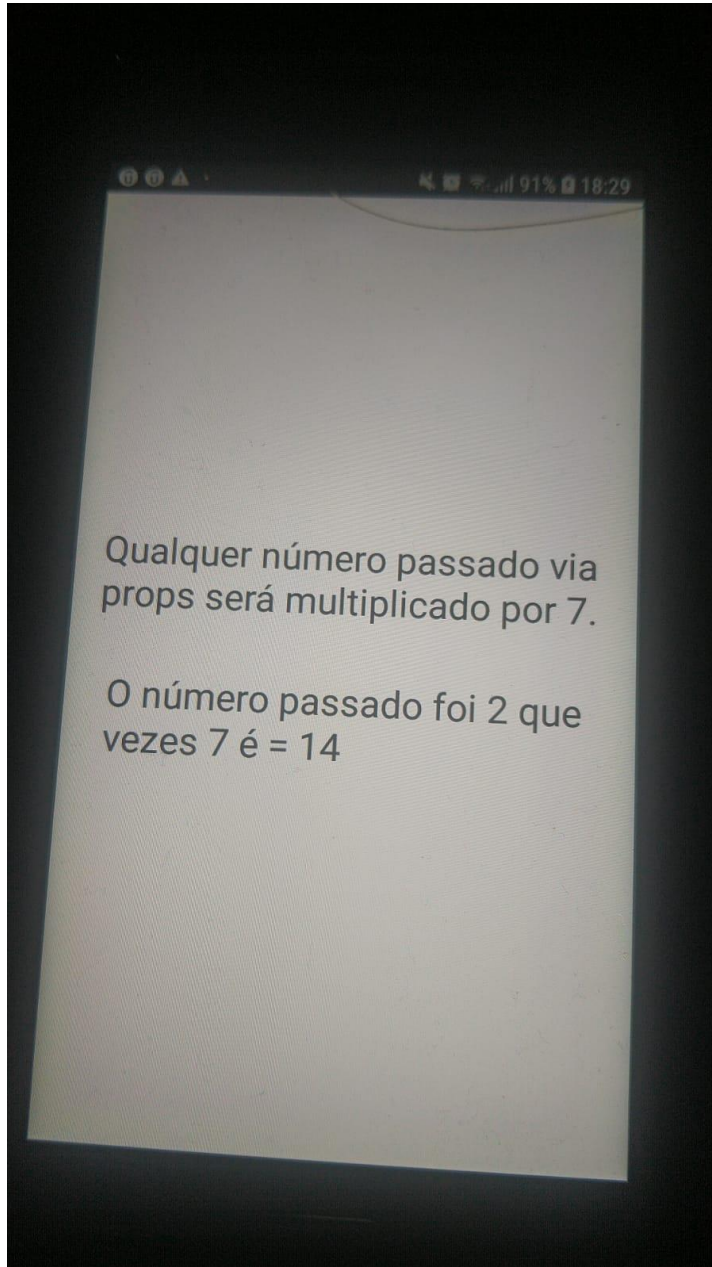
```
    }  
  }  
  
  const styles = StyleSheet.create({  
  
    fundo: {  
      flex: 1,  
      justifyContent: 'center',  
      padding: 15,  
    },  
    texto: {  
      color: 'black',  
      fontSize: 24,  
    },  
  })  
})
```

NO APP.JS...

```
import React from 'react'  
import { EstadoComInput } from '../src/classes/EstadosNaClasse'  
  
export default () => {  
  return (  
    /* Coloque seus componentes abaixo...*/  
    //Veja que não muda a forma como passamos parâmetros por através da instância do Componente...  
    <EstadoComInput number={2} />  
  )  
}
```

}

RESULTADO NO CELULAR SEM INPUT...



RESULTADO NO CELULAR COM INPUT...

