

Usabilidade, Desenvolvimento Web, Mobile e Jogos

React Native
APP Calculadora



Prof. Juliano Gaspar





Prof. Dr. Juliano Gaspar

Email: julianogaspar@gmail.com

Formação

- **Cientista da Computação** pela UNIVALI (SC)
- **Mestre em Informática Médica** pela UP (Portugal)
- **Doutor em Saúde Digital** pela UFMG
- **Pós-doutor em Tec. para Educação em Saúde** pela UFMG

Educação

Professor Convidado do Departamento de GOB da UFMG

- o Introdução à Pesquisa Científica II
- o Informática Médica
- o Informação em Decisão em Saúde

Professor Grupo Ânima: Una, Unibh e Faseh

- o Projetos e Processos em TI
- o Usabilidade, desenvolvimento Web, Mobile e Jogos
- o Pós-graduação em Saúde Digital e Telemedicina

Professor da Facisa BH

- o Linguagens de Programação (Linguagem C)
- o Programação orientada à objetos (Python)
- o Programação para dispositivos móveis
- o Programação para Web

Professor da IESLA

- o Gestão das Tecnologias da Informação

Inovação, Pesquisa, Desenvolvimento e Extensão

- o Vice-coordenador do núcleo de pesquisa em Informática Aplicada à Saúde da UFMG
- o Membro do CINTESIS - Universidade Porto - Portugal
- o Membro da SBIS - Sociedade Brasileira de Informática em Saúde
- o Presidente do Comitê Científico do CBIS 2022
- o Revisor de revistas científicas

Linhas de pesquisa e projetos

- o Detecção da prematuridade através da interação entre a luz e a pele neonatal
- o Sistema de Informação para Atenção Materno Infantil
- o Eliminando a morte materna: uma resposta a esse desafio no seu bolso.

Programas e projetos de extensão

- o Informática e Saúde
- o Prevenção da COVID-19 em APP
- o Meu Pré-natal (APP)
- o Projeto Educação Continuada em Informática



Usabilidade, Desenvolvimento Web, Mobile e Jogos

Prof. Juliano Gaspar

Contatos

Email

julianogaspar@gmail.com

Instagram

[@prof.julianogaspar](https://www.instagram.com/prof.julianogaspar)

Twitter

<https://twitter.com/JulianoGaspar21>

Currículo Lattes

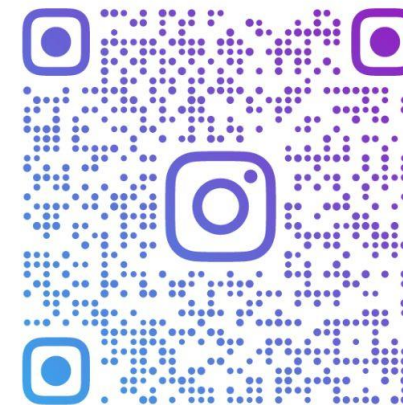
<http://lattes.cnpq.br/3926707936198077>

Orcid ID

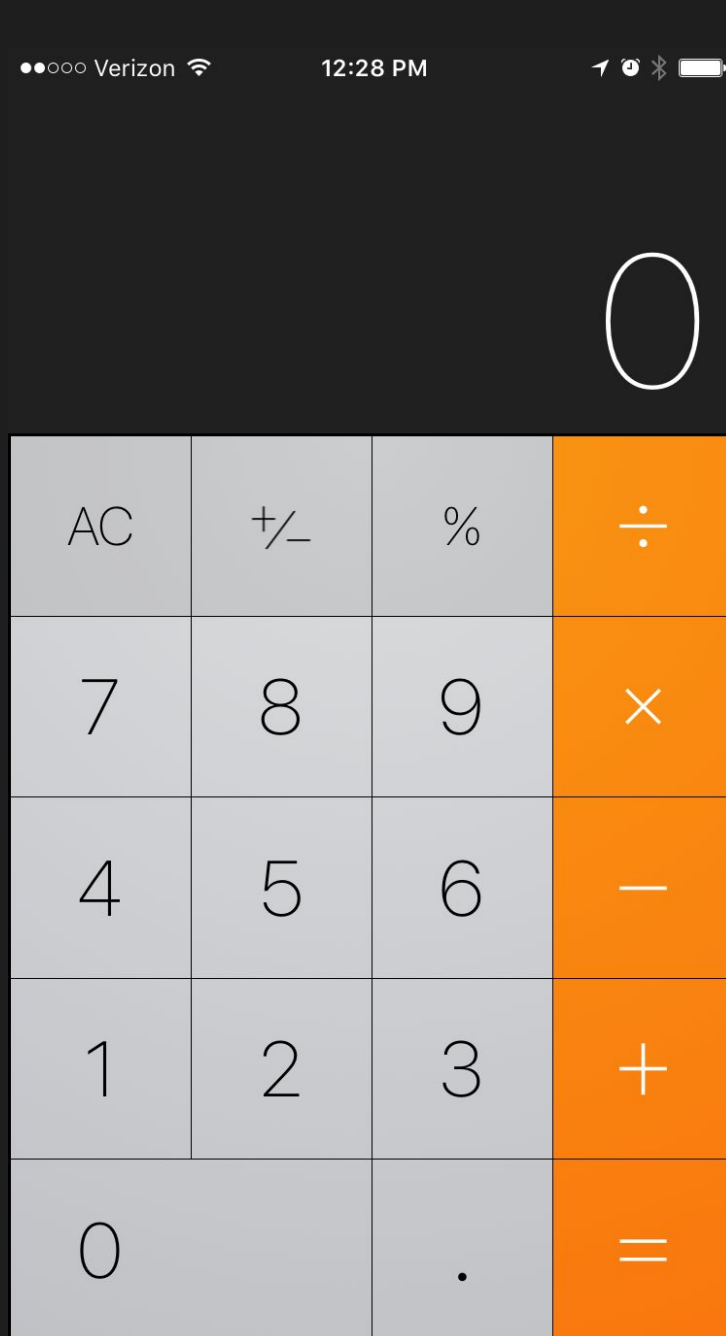
<https://orcid.org/0000-0003-0670-9021>

Research Gate

https://www.researchgate.net/profile/Juliano_Gaspar



PROF.JULIANOGASPAR



Nesta aula vamos criar uma calculadora com o paradigma de componentes funcionais.

Tanto para parte da camada de visualização, quanto todas as lógicas de cálculos.



```
Windows PowerShell
PS C:\Users\Juliano\Documents\ProjetosReact\Aulas_ReactNative> expo init AppCalculadora

There is a new version of expo-cli available (5.4.6).
You are currently using expo-cli 5.2.0
Install expo-cli globally using the package manager of your choice;
for example: `npm install -g expo-cli` to get the latest version

✓ Choose a template: » blank a minimal app as clean as an empty canvas
✓ Downloaded template.
⊠ Using Yarn to install packages. Pass --npm to use npm instead.
✓ Installed JavaScript dependencies.

⊠ Your project is ready!

To run your project, navigate to the directory and run one of the following yarn commands.

- cd AppCalculadora
- yarn start # you can open iOS, Android, or web from here, or run them directly with the commands below.
- yarn android
- yarn ios # requires an iOS device or macOS for access to an iOS simulator
- yarn web
PS C:\Users\Juliano\Documents\ProjetosReact\Aulas_ReactNative>
```



```
Windows PowerShell
PS C:\Users\Juliano\Documents\ProjetosReact\Aulas_ReactNative> cd .\AppCalculadora\
PS C:\Users\Juliano\Documents\ProjetosReact\Aulas_ReactNative\AppData\Calculadora> expo start

There is a new version of expo-cli available (5.4.6).
You are currently using expo-cli 5.2.0
Install expo-cli globally using the package manager of your choice;
for example: `npm install -g expo-cli` to get the latest version

Starting project at C:\Users\Juliano\Documents\ProjetosReact\Aulas_ReactNative\AppData\Calculadora
Developer tools running on http://localhost:19002
Starting Metro Bundler



> Metro waiting on exp://192.168.0.14:19000
> Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

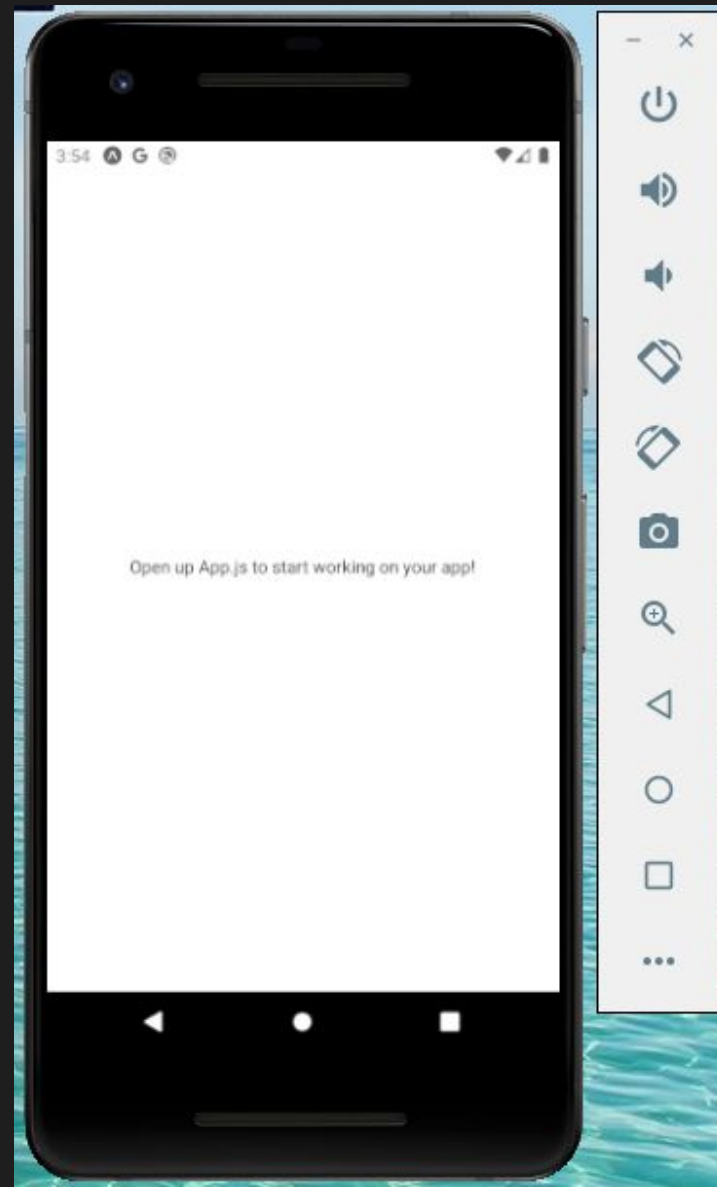
> Press a | open Android
> Press w | open web
```

Se tiver um emulador instalado, pode rodar visualizando no Android.
Ou, ver no Browser web.



Usabilidade, Desenvolvimento Web, Mobile e Jogos

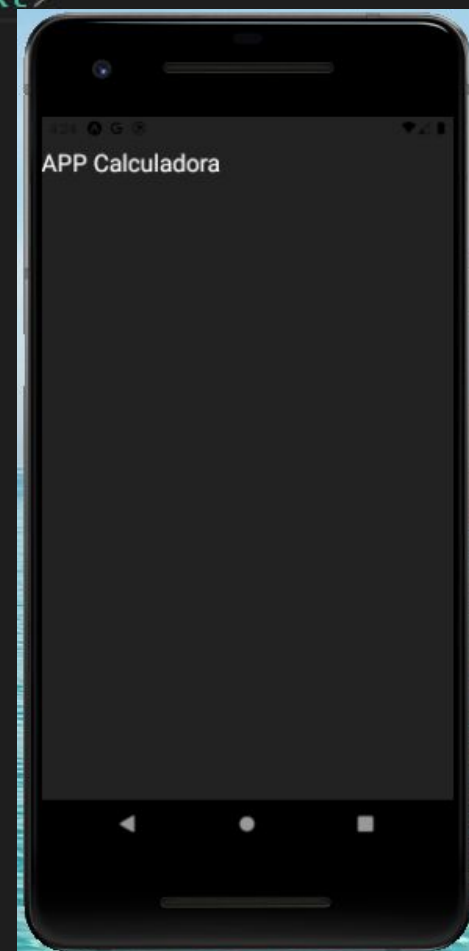
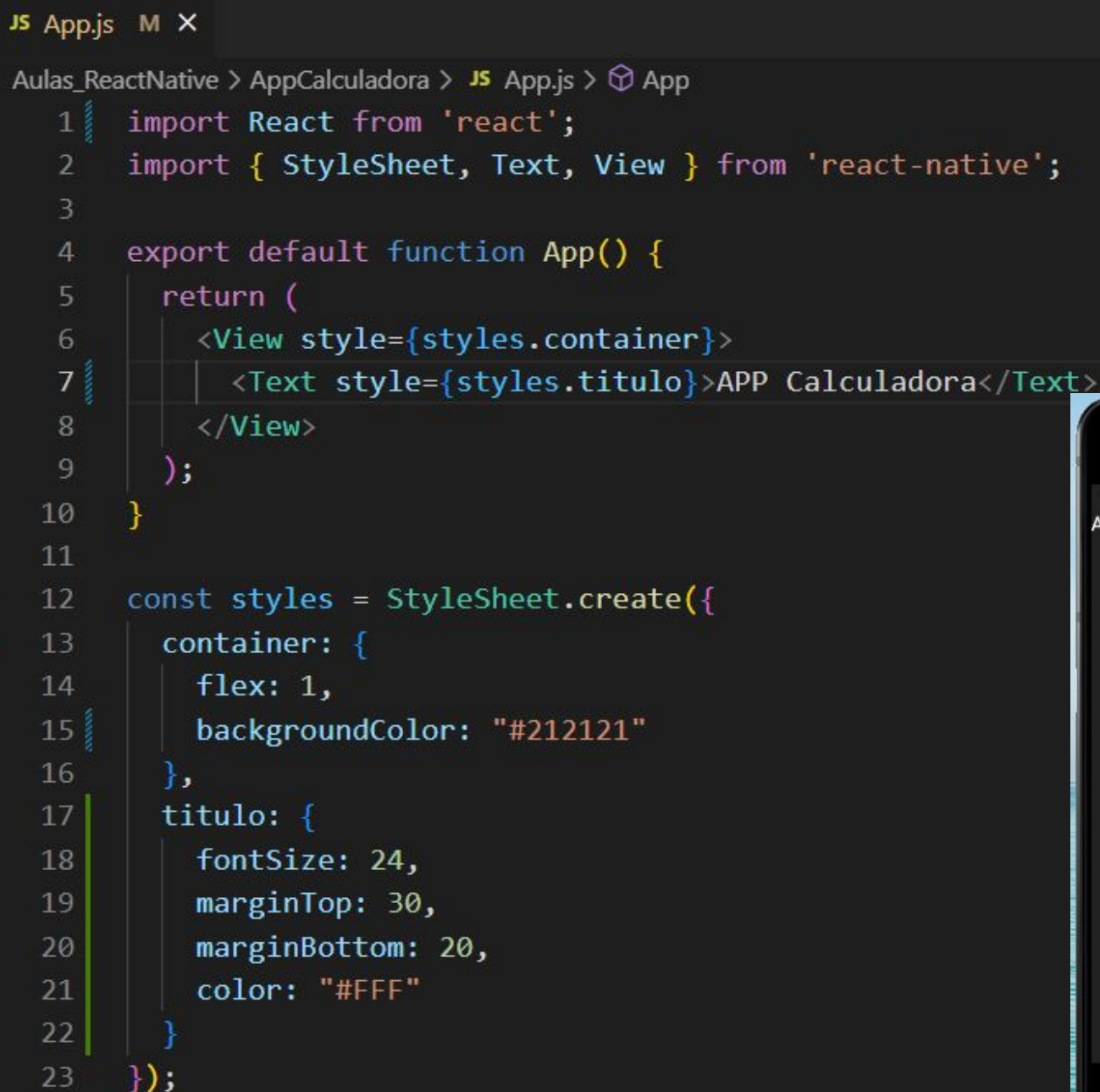
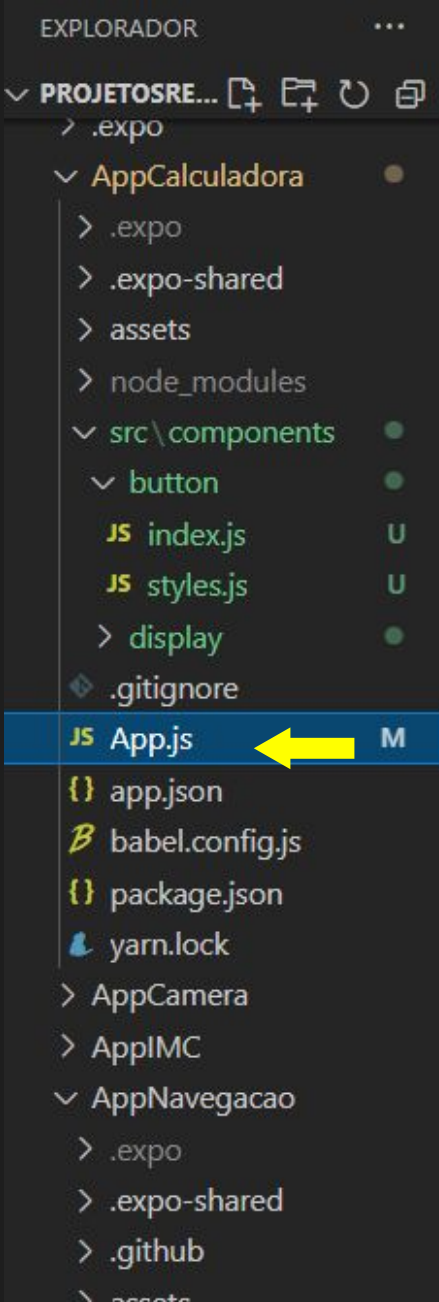
APP Calculadora



App.js



Prof. Juliano Gaspar









- Criar a pasta 'src'
- Dentro da pasta 'src' criar a pasta 'components'
- Dentro da pasta 'components' criar
 - pasta button com index.js e style.js
 - pasta display com index.js e style.js



EXPLORADOR

PROJETOSRE...    

> .expo

AppCalculadora

> .expo

> .expo-shared

> assets

> node_modules

src\components

button

JS index.js U

JS styles.js U

> display

.gitignore

JS App.js M

JS App.js M JS index.js U X

Aulas_ReactNative > AppCalculadora > src > components > button > JS index.js > ...

```
1  import React from 'react';
2  import { Text, TouchableHighlight } from 'react-native';
3  import { styles } from './styles';
4
5  export default function Button() {
6
7      return (
8          <TouchableHighlight >
9              <Text style={styles.button}>AC</Text>
10          </TouchableHighlight>
11      );
12  }
```



EXPLORADOR ... JS App.js M JS styles.js U X

PROJETOSRE...
 .expo
 AppCalculadora
 .expo
 .expo-shared
 assets
 node_modules
 src\components
 button
 JS index.js U
 JS styles.js ←
 display
 .gitignore
 JS App.js M
 {} app.json
 {} babel.config.js
 {} package.json
 yarn.lock
 AppCamera
 AppIMC
 AppNavegacao
 .expo
 expo-shared

Aulas_ReactNative > AppCalculadora > src > components > button > JS styles.js > ...

```
1  import { Dimensions, StyleSheet } from "react-native";
2
3  exports.styles = StyleSheet.create({
4    button: {
5      fontSize: 40,
6      height: Dimensions.get('window').width / 4,
7      width: Dimensions.get('window').width / 4,
8      padding: 20,
9      backgroundColor: '#F0F0F0',
10     textAlign: 'center',
11     borderWidth: 1,
12     borderColor: '#888',
13   },
14   operationButton: {
15     color: '#fff',
16     backgroundColor: '#FA8231'
17   },
18   buttonDouble: {
19     width: Dimensions.get('window').width / 2,
20   }
21 });
```


Usabilidade, Desenvolvimento Web, Mobile e Jogos

APP Calculadora



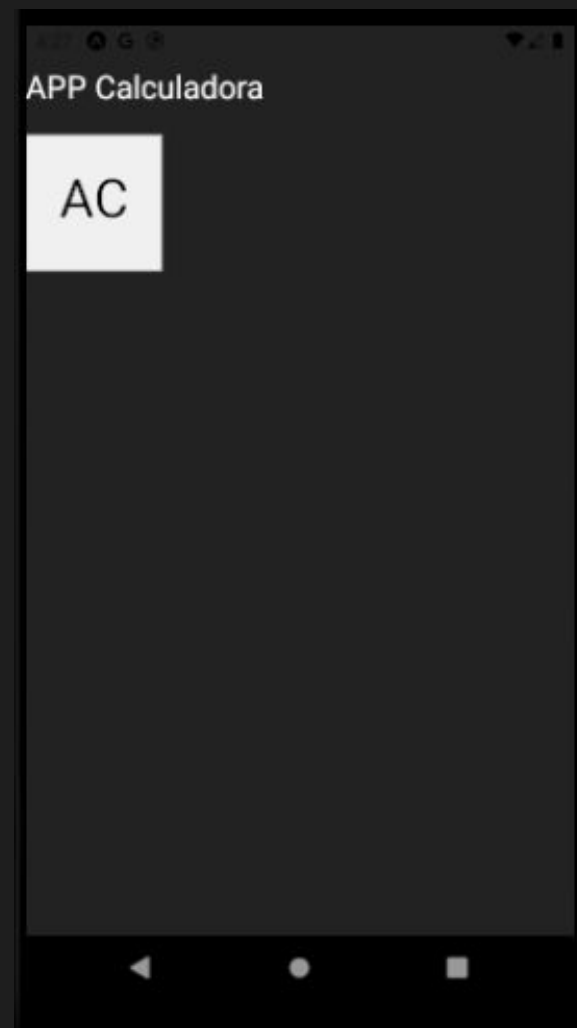
EXPLORADOR

PROJETOS REACT

- Aula_UDWMJ
- Aulas_ReactNative
 - .expo
 - AppCalculadora
 - .expo
 - .expo-shared
 - assets
 - node_modules
 - src\components
 - button
 - JS index.js
 - JS styles.js
 - display
 - .gitignore
 - JS App.js
 - app.json
 - babel.config.js
 - package.json
 - yarn.lock
 - AppCamera
 - AppIMC
 - AppNavegacao
 - .expo
 - .expo-shared
 - .github
 - assets
 - node_modules
 - src
 - .gitignore

ESTRUTURA DO CÓDIGO

```
JS App.js M X JS index.js U JS button.js
Aulas_ReactNative > AppCalculadora > JS App.js > App
1 import React from 'react';
2 import { StyleSheet, Text, View } from 'react-native';
3 import Button from './src/components/button'
4
5 export default function App() {
6   return (
7     <View style={styles.container}>
8       <Text style={styles.titulo}>APP Calculadora</Text>
9
10      <View style={styles.buttons}>
11        <Button />
12      </View>
13    </View>
14  );
15 }
16
17 const styles = StyleSheet.create({
18   container: {
19     flex: 1,
20     backgroundColor: "#212121"
21   },
22   titulo: {
23     fontSize: 24,
24     marginTop: 30,
25     marginBottom: 20,
26     color: "#FFF"
27   },
28   buttons: {
29     flexDirection: 'row',
30     flexWrap: 'wrap'
31   }
```





EXPLORADOR ... JS App.js M JS index.js U X

PROJETOSRE...
package-lock.json
package.json
Calculadora - arquivos ...
Calculadora2
TesteAula
Aula_UDWMJ
Aulas_ReactNative
 .expo
 AppCalculadora
 .expo
 .expo-shared
 assets
 node_modules
 src\components
 button
 JS index.js U
 JS styles.js U
 display
 .gitignore
 JS App.js M

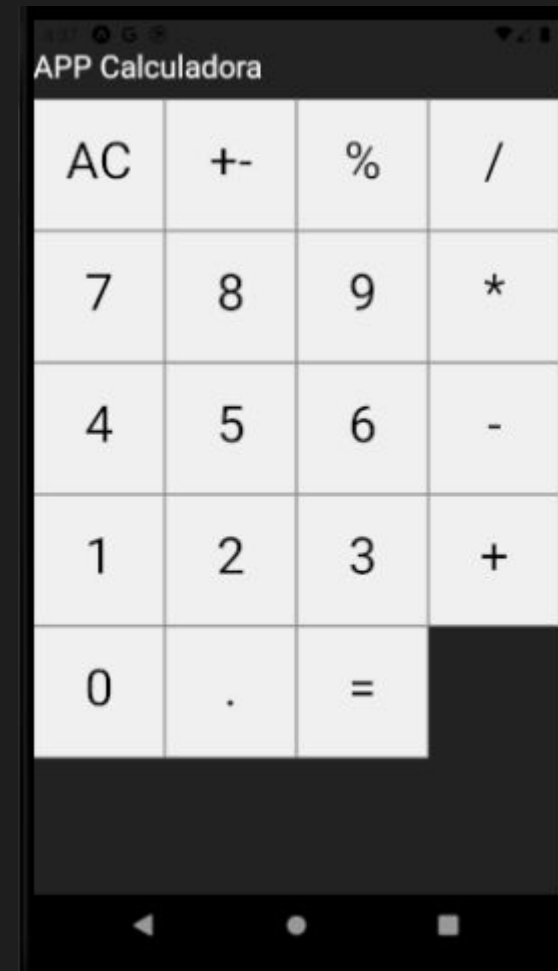
Aulas_ReactNative > AppCalculadora > src > components > button > JS index.js > Button

```
1  import React from 'react';
2  import { Text, TouchableHighlight } from 'react-native';
3  import { styles } from './styles';
4
5  export default function Button(props) {
6
7      return (
8          <TouchableHighlight >
9              <Text style={styles.button}>{props.label}</Text>
10          </TouchableHighlight>
11      );
12  }
```

Two yellow arrows point to the function definition and the JSX element in the code.



```
JS App.js M X JS styles.js U JS index.js U
Aulas_ReactNative > AppCalculadora > JS App.js > styles > titulo > marginTop
export default function App() {
  6   return (
  7     <View style={styles.container}>
  8       <Text style={styles.titulo}>APP Calculadora</Text>
  9
 10       <View style={styles.buttons}>
 11         <Button label='AC' />
 12         <Button label='+-' />
 13         <Button label='%' />
 14         <Button label='/' />
 15         <Button label='7' />
 16         <Button label='8' />
 17         <Button label='9' />
 18         <Button label='*' />
 19         <Button label='4' />
 20         <Button label='5' />
 21         <Button label='6' />
 22         <Button label='-' />
 23         <Button label='1' />
 24         <Button label='2' />
 25         <Button label='3' />
 26         <Button label='+' />
 27         <Button label='0' />
 28         <Button label='.' />
 29         <Button label='=' />
 30       </View>
 31     </View>
 32   );
 33 }
```





JS App.js M

JS index.js U X

JS button.js

Aulas_ReactNative > AppCalculadora > src > components > button > JS index.js > Button

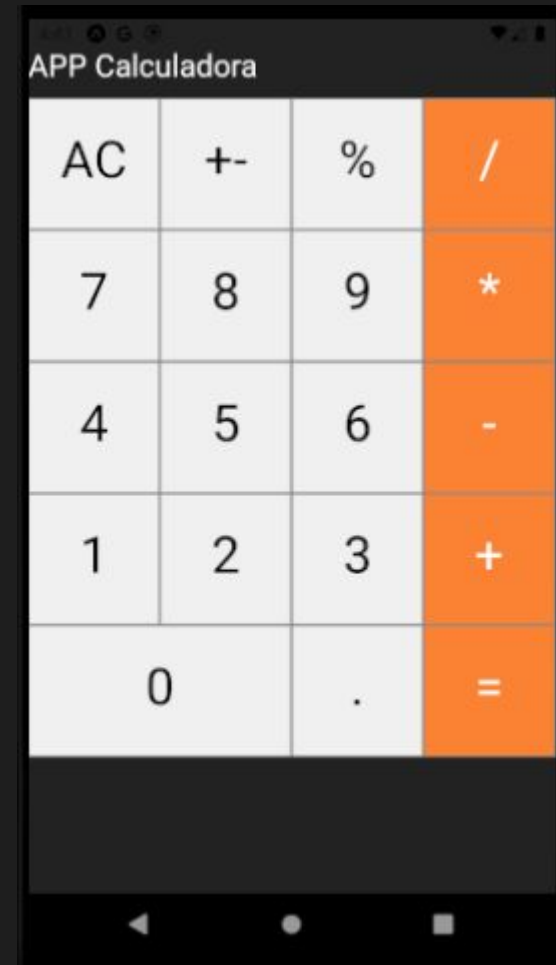
```
1  import React from 'react';
2  import { Text, TouchableHighlight } from 'react-native';
3  import { styles } from './styles';
4
5  export default function Button(props) {
6    // o estilos podem ser agrupados, sobrepostos, complementados,
7    // usando array com cada estilo que deseja aplicar
8    // criar uma constante para conter todos os estilos que o botão irá ter
9    const stylesButton = [styles.button]
10
11    if (props.double) stylesButton.push(styles.buttonDouble)
12    if (props.operation) stylesButton.push(styles.operationButton)
13
14    return (
15      <TouchableHighlight >
16        <Text style={stylesButton} props.label </Text>
17      </TouchableHighlight>
18    );
19  }
```




JS App.js M X JS index.js U JS button.js

Aulas_ReactNative > AppCalculadora > JS App.js > App

```
5 export default function App() {
6   return (
7     <View style={styles.container}>
8       <Text style={styles.titulo}>APP Calculadora</Text>
9
10      <View style={styles.buttons}>
11        <Button label='AC' />
12        <Button label='+-' />
13        <Button label='%' />
14        <Button label='/' operation />
15        <Button label='7' />
16        <Button label='8' />
17        <Button label='9' />
18        <Button label='*' operation />
19        <Button label='4' />
20        <Button label='5' />
21        <Button label='6' />
22        <Button label='- ' operation />
23        <Button label='1' />
24        <Button label='2' />
25        <Button label='3' />
26        <Button label='+ ' operation />
27        <Button label='0' double />
28        <Button label='.' />
29        <Button label='=' operation />
30      </View>
31    </View>
  )
}
```





EXPLORADOR ... JS App.js M JS index.js U JS styles.js U X

✓ PROJETOSRE... [ícone] [ícone] [ícone] [ícone]





- ✓ AppCalculadora
 - > .expo
 - > .expo-shared
 - > assets
 - > node_modules
 - ✓ src\components
 - ✓ button
 - JS index.js U
 - JS styles.js U
 - ✓ display
 - JS index.js U
 - JS styles.js U** ←
 - ◇ .gitignore
 - JS App.js M
 - { } app.json

Aulas_ReactNative > AppCalculadora > src > components > display > JS styles.js > ...

```
1  import { StyleSheet } from "react-native";
2
3  exports.styles = StyleSheet.create({
4    display: {
5      flex: 1,
6      padding: 20,
7      justifyContent: 'center',
8      backgroundColor: 'rgba(0,0,0,0.6)',
9      alignItems: 'flex-end'
10   },
11   displayValue: {
12     fontSize: 60,
13     color: '#fff'
14   }
15  });
```



EXPLORADOR

PROJETOSRE...    

- AppCalculadora
 - .expo
 - .expo-shared
 - assets
 - node_modules
 - src\components
 - button
 - JS index.js U
 - JS styles.js U
 - display
 - JS index.js U
 - JS styles.js U
 - .gitignore
 - JS App.js M
 - { } app.json
 - Babel.config.js

JS App.js M JS index.js U X JS styles.js U

Aulas_ReactNative > AppCalculadora > src > components > display > JS index.js > Display

```
1  import React from 'react';
2  import { Text, View } from 'react-native';
3  import { styles } from './styles';
4
5  export default function Display(props) {
6    return (
7      <View style={styles.display}>
8        <Text style={styles.displayValue} numberOfLines={1}>
9          {props.value}
10        </Text>
11      </View>
12    );
13  }
```

App.js



Prof. Juliano Gaspar



EXPLORADOR

PROJETOSRE...

- > .expo
- ▼ AppCalculadora
 - > .expo
 - > .expo-shared
 - > assets
 - > node_modules
 - ▼ src\components
 - ▼ button
 - JS index.js U
 - JS styles.js U
 - ▼ display
 - JS index.js U
 - JS styles.js U
 - ◆ .gitignore
 - JS App.js ← M
 - { } app.json

Aulas_ReactNative > AppCalculadora > JS App.js > App

```
1 import React from 'react';
2 import { StyleSheet, Text, View } from 'react-native';
3 import Button from './src/components/button'
4 import Display from './src/components/display'
5
6 export default function App() {
7   return (
8     <View style={styles.container}>
9       <Text style={styles.titulo}>APP Cal
10      <Display value='0' />
11
12     <View style={styles.buttons}>
13       <Button label='AC' />
14       <Button label='+-' />
15       <Button label='%' />
16       <Button label='/' operation />
17
```

APP Calculadora

0

AC	+ -	%	/
7	8	9	*
4	5	6	-
1	2	3	+
0	.	=	



```
JS App.js ...\AppCalculadora M • ← JS index.js U JS App.js ...\Calculadora JS App.js ...\Calculadora2
Aulas_ReactNative > AppCalculadora > JS App.js > App

4 import Display from './src/components/display'
5 import { useState } from 'react'; ←
6
7 export default function App() {
8   const [displayValue, setDisplayValue] = useState('0')
9   const [clearDisplay, setClearDisplay] = useState(false) // se precisar limpar o display no
10  const [operation, setOperation] = useState(null) // qual operação foi clicada
11  const [values, setValues] = useState([0, 0]) // Valor digitado anterior e atual ←
12  const [current, setCurrent] = useState(0) // qual valor (anterior ou atual)
13
14  clearMemory = () => { ←
15    setDisplayValue('0')
16    setClearDisplay(false)
17    setOperation(null)
18    setValues([0, 0])
19    setCurrent(0)
20  }
21
22  addDigit = n => { ←
23    console.warn('chegou aqui', n)
24    setDisplayValue(n)
25  }
```

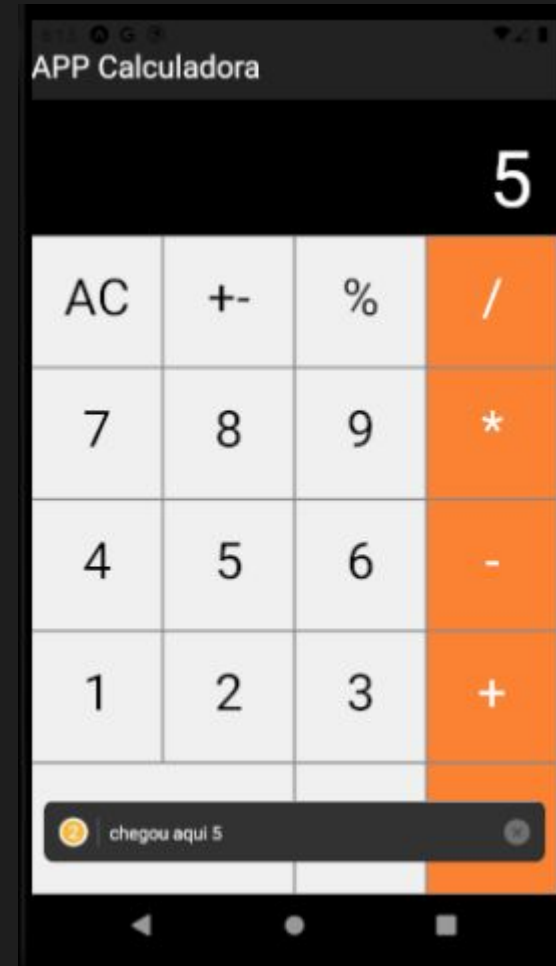
App.js

Hooks -> useState
clearMemory
addDigit



App.js

```
51   return (  
52     <View style={styles.container}>  
53       <Text style={styles.titulo}>APP Calculadora</Text>  
54       <Display value={displayValue} />  
55  
56       <View style={styles.buttons}>  
57         <Button label='AC' onClick={clearMemory} />   
58         <Button label='+-' />  
59         <Button label='%' />  
60         <Button label='/' operation />  
61         <Button label='7' onClick={addDigit} />  
62         <Button label='8' onClick={addDigit} />   
63         <Button label='9' onClick={addDigit} />  
64         <Button label='*' operation />  
65         <Button label='4' onClick={addDigit} />  
66         <Button label='5' onClick={addDigit} />   
67         <Button label='6' onClick={addDigit} />  
68         <Button label='-' operation />  
69         <Button label='1' onClick={addDigit} />  
70         <Button label='2' onClick={addDigit} />   
71         <Button label='3' onClick={addDigit} />  
72         <Button label='+' operation />  
73         <Button label='0' double onClick={addDigit} />  
74         <Button label='.' onClick={addDigit} />   
75         <Button label='=' operation />  
76       </View>  
77     </View>  
  )
```





EXPLORADOR

PROJETOSRE... [+] [] [↺] [↻]

- Aulas_ReactNative
 - > .expo
 - ▼ AppCalculadora
 - > .expo
 - > .expo-shared
 - > assets
 - > node_modules
 - ▼ src\components
 - ▼ button
 - JS index.js** ←
 - JS styles.js U
 - ▼ display
 - JS index.js U
 - JS styles.js U
 - ◆ .gitignore
 - JS App.js M
 - { } app.json
 - 📄 babel.config.js
 - { } package.json
 - 👤 yarn.lock

Aulas_ReactNative > AppCalculadora > src > components > button > JS index.js > Button

```
1 import React from 'react';
2 import { Text, TouchableHighlight } from 'react-native';
3 import { styles } from './styles';
4
5 export default function Button(props) {
6   // o estilos podem ser agrupados, sobrepostos, complementados,
7   // usando array com cada estilo que deseja aplicar
8   // criar uma constante para conter todos os estilos que o botão irá
9   const stylesButton = [styles.button]
10
11   if (props.double) stylesButton.push(styles.buttonDouble)
12   if (props.operation) stylesButton.push(styles.operationButton)
13
14   return (
15     <TouchableHighlight onPress={() => props.onClick(props.label)}>
16       <Text style={stylesButton}>{props.label}</Text>
17     </TouchableHighlight>
18   );
19 }
```




```
21
22 addDigit = n => {
23   // console.warn('chegou aqui', n)
24   // setDisplayValue(n)
25
26   const clear = (displayValue === '0' || clearDisplay === true)
27
28   if (n === '.' && displayValue.includes('.') && !clear) {
29     return
30   }
31
32   let currentValue
33   if (clear === true) {
34     currentValue = ''
35   } else {
36     currentValue = displayValue
37   }
38   const display = currentValue + n
39
40   setDisplayValue(display)
41   setClearDisplay(false)
42
43   if (n !== '.') {
44     const newValue = parseFloat(display)
45     const valuesTemp = [...values]
46     valuesTemp[current] = newValue
47     setValues(valuesTemp)
48   }
49 }
```

App.js





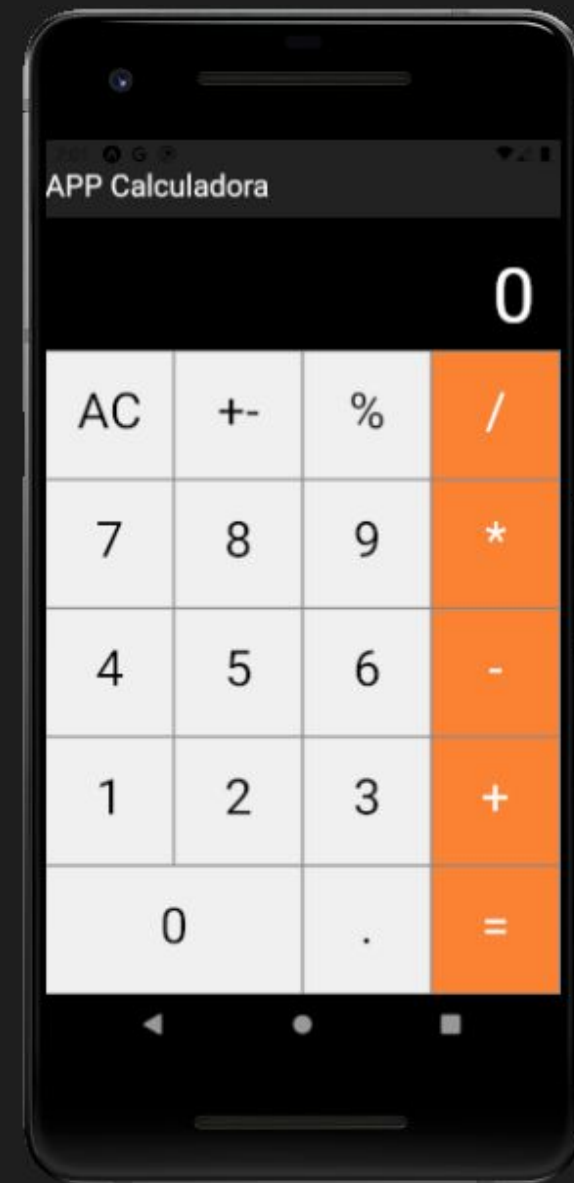
```
73   return (  
74     <View style={styles.container}>  
75       <Text style={styles.titulo}>APP Calculadora</Text>  
76       <Display value={displayValue} />  
77  
78       <View style={styles.buttons}>  
79         <Button label='AC' onClick={clearMemory} />  
80         <Button label='+-' onClick={calculate} />  
81         <Button label='%' onClick={calculate} />  
82         <Button label='/' operation onClick={calculate} />  
83         <Button label='7' onClick={addDigit} />  
84         <Button label='8' onClick={addDigit} />  
85         <Button label='9' onClick={addDigit} />  
86         <Button label='*' operation onClick={calculate} />  
87         <Button label='4' onClick={addDigit} />  
88         <Button label='5' onClick={addDigit} />  
89         <Button label='6' onClick={addDigit} />  
90         <Button label='- ' operation onClick={calculate} />  
91         <Button label='1' onClick={addDigit} />  
92         <Button label='2' onClick={addDigit} />  
93         <Button label='3' onClick={addDigit} />  
94         <Button label='+' operation onClick={calculate} />  
95         <Button label='0' double onClick={addDigit} />  
96         <Button label='.' onClick={addDigit} />  
97         <Button label='=' operation onClick={calculate} />  
98       </View>  
99     </View>  
  )
```

App.js





```
51 calculate = op => {
52   if (current === 0) {
53     // console.warn('Foi pro 2º valor, operação', op, values)
54     setOperation(op)
55     setCurrent(1)
56     setClearDisplay(true)
57   } else {
58     // console.warn('calcular', op, values)
59     const equals = op === '='
60     const valuesTemp = [...values]
61
62     // Precisa tratar as operações (+- e a %),
63     // Sugestão usar um if, antes das próximas linhas...
64
65     valuesTemp[0] = eval(`${values[0]} ${operation} ${values[1]}`)
66     valuesTemp[1] = 0
67     // console.warn('calcular', op, valuesTemp)
68
69     setDisplayValue(`${valuesTemp[0]}`)
70     setClearDisplay(!equals)
71     setOperation(equals ? null : op)
72     setValues(valuesTemp)
73     setCurrent(equals ? 0 : 1)
74   }
75 }
```



**Usabilidade,
Desenvolvimento Web, Mobile e Jogos**

Obrigado!



Prof. Dr. Juliano Gaspar
julianogaspar@gmail.com

<http://lattes.cnpq.br/3926707936198077>

Prof. Juliano Gaspar



Desenvolvimento Mobile

APP Calculadora