

Usabilidade, Desenvolvimento Web, Mobile e Jogos

APP IMC



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 Arquitetura Empresarial
- 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

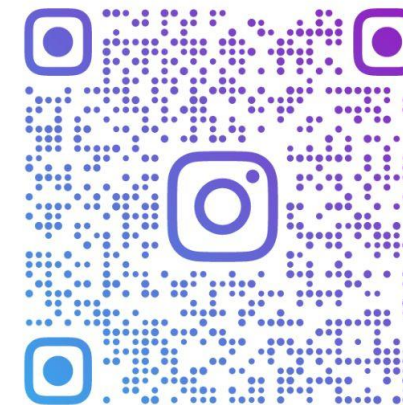
<http://lattes.cnpq.br/3926707936198077>

Orcid ID

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

Research Gate

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



PROF.JULIANOGASPAR



APP - IMC

Altura

Exemplo 1.75

Peso

Exemplo 78.245

Calcular

APP - IMC - LISTA

Seu imc é igual a:

2.00

Compartilhar

Calcular novamente

IMC: 1.25

IMC: 2.00

Tópicos abordados:

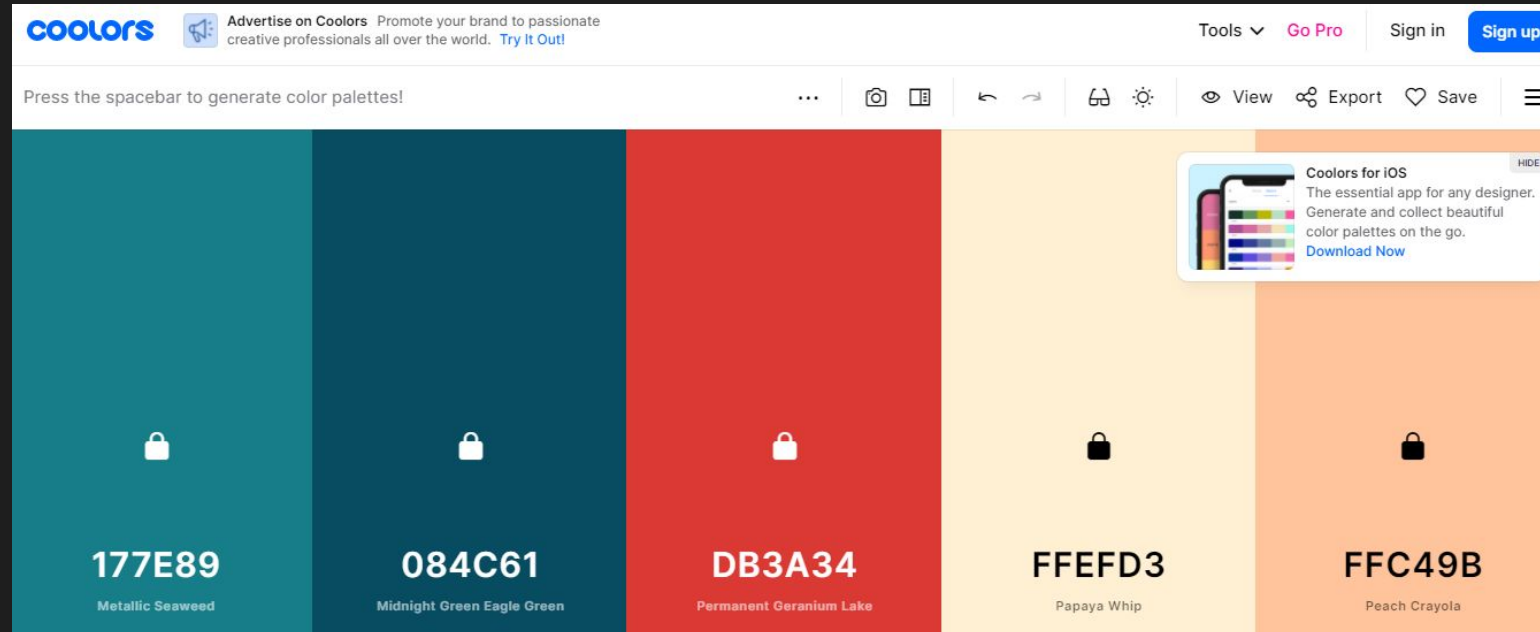
- JSX
- Formulários
- StyleSheet
- MVC
- API's
- FlatList
- Erros
- Components
- useState
- entre outros



https://coolors.co/

Usabilidade, Desenvolvimento Web, Mobile e Jogos

APP IMC



APP - IMC

Altura

1.77

Peso

83

Calcular novamente

Seu imc é igual a:

26.49

Prof. Juliano Gaspar



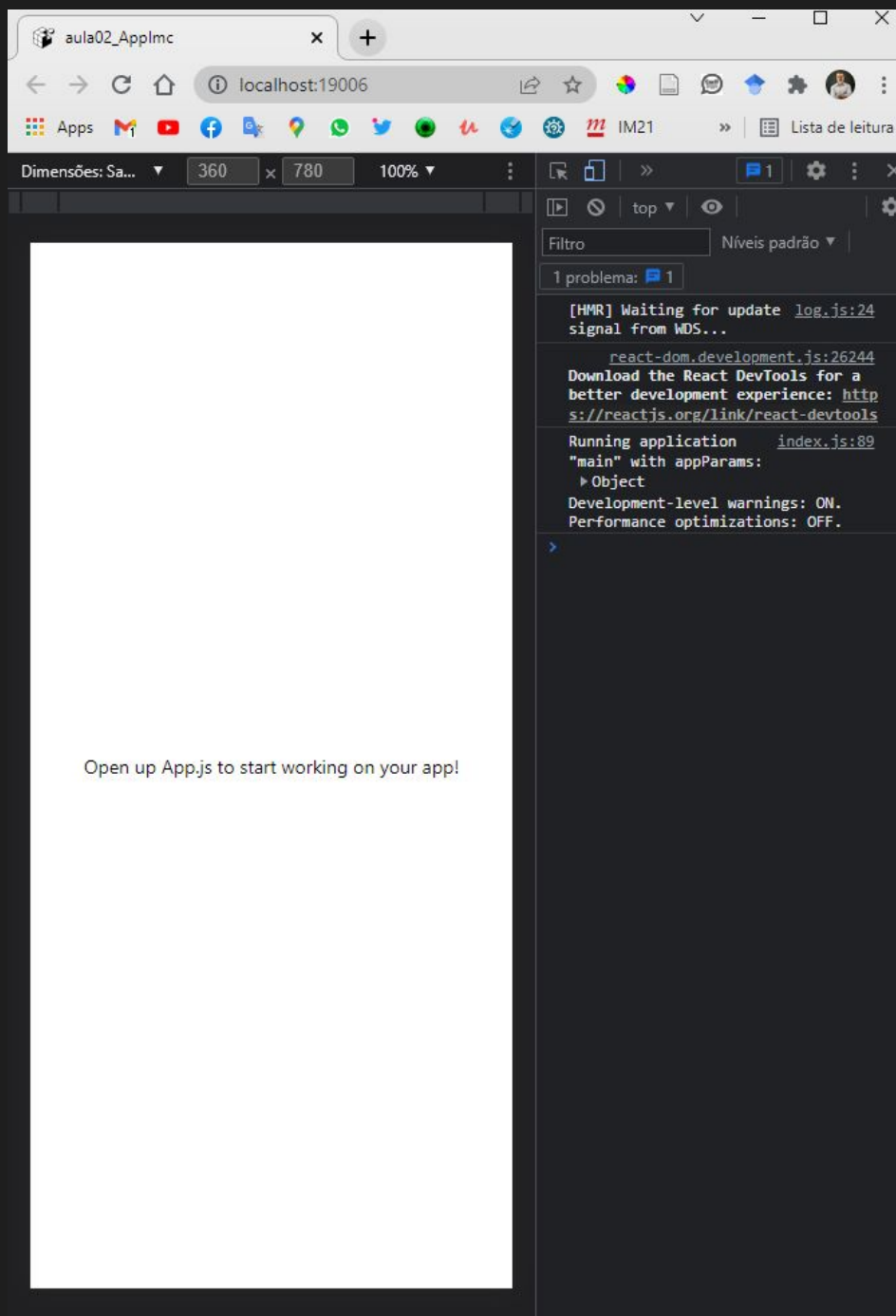
Windows PowerShell

```
PS C:\Users\Juliano\Documents\ProjetosReact\Testes> expo init aula02_AppImc
✓ Choose a template: » blank          a minimal app as clean as an empty canvas
✓ Downloaded template.
Ⓟ Using npm to install packages.
✓ Installed JavaScript dependencies.

Ⓟ Your project is ready!

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

- cd aula02_AppImc
- npm start # you can open iOS, Android, or web from here, or run them directly with the commands below.
- npm run android
- npm run ios # requires an iOS device or macOS for access to an iOS simulator
- npm run web
PS C:\Users\Juliano\Documents\ProjetosReact\Testes> cd .\aula02_AppImc\
PS C:\Users\Juliano\Documents\ProjetosReact\Testes\aula02_AppImc> npm start_
```





EXPLORADOR

PROJETOSREACT

> .idea

> AnosAnteriores

Testes

> aula01

aula02_AppImc

> .expo

> .expo-shared

> assets

> node_modules

src\components

> Form

> ResultIMC

Title

JS index.js

JS style.js

.gitignore

JS App.js

{ } app.json

Babel.config.js

{ } package-lock.json

{ } package.json

> aula02_imc

> aula02_imc_api

aula05_imc_lista

> .expo

JS App.js

M X

Testes > aula02_AppImc > JS App.js > ...

1 import React from 'react';

2 import { StyleSheet, Text, View } from 'react-native';

3

4

5

6 export default function App() {

7 return (

8 <View style={styles.container}>

9 <Text style={styles.textTitle}>APP - IMC</Text>

10 </View>

11);

12 }

13

14 const styles = StyleSheet.create({

15 container: {

16 flex: 1,

17 backgroundColor: '#084C61',

18 paddingTop: 60,

19 },

20 textTitle: {

21 color: "#FFF",

22 fontSize: 24,

23 fontWeight: "bold",

24 }

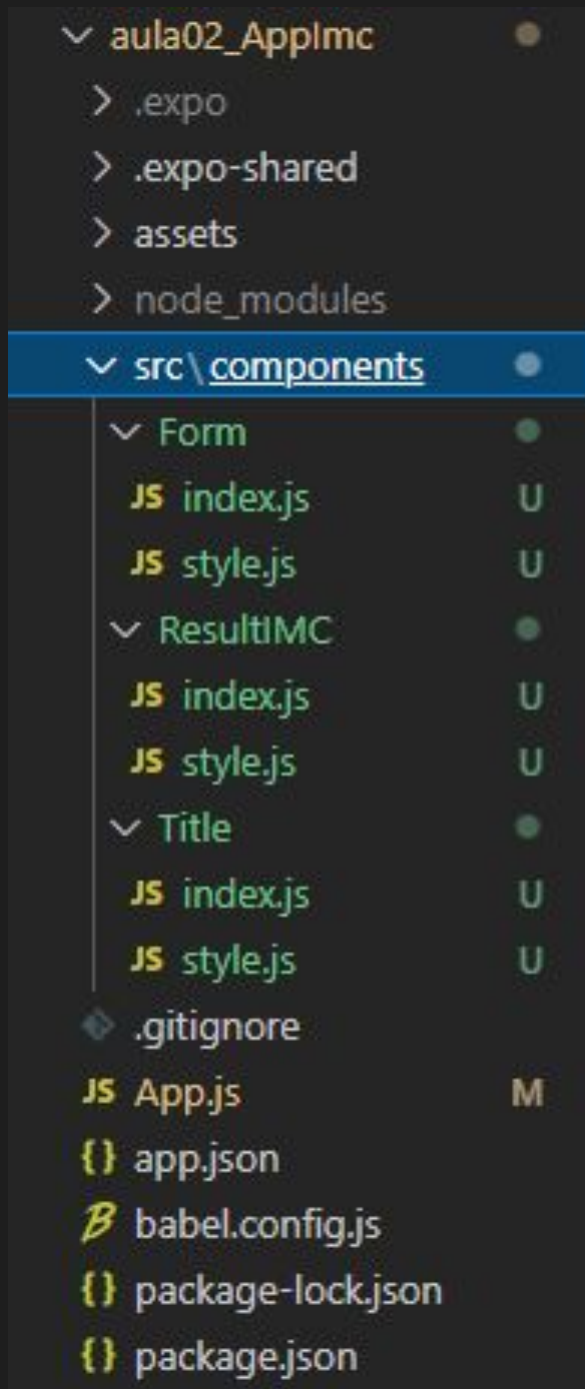
25 });

APP - IMC

Style

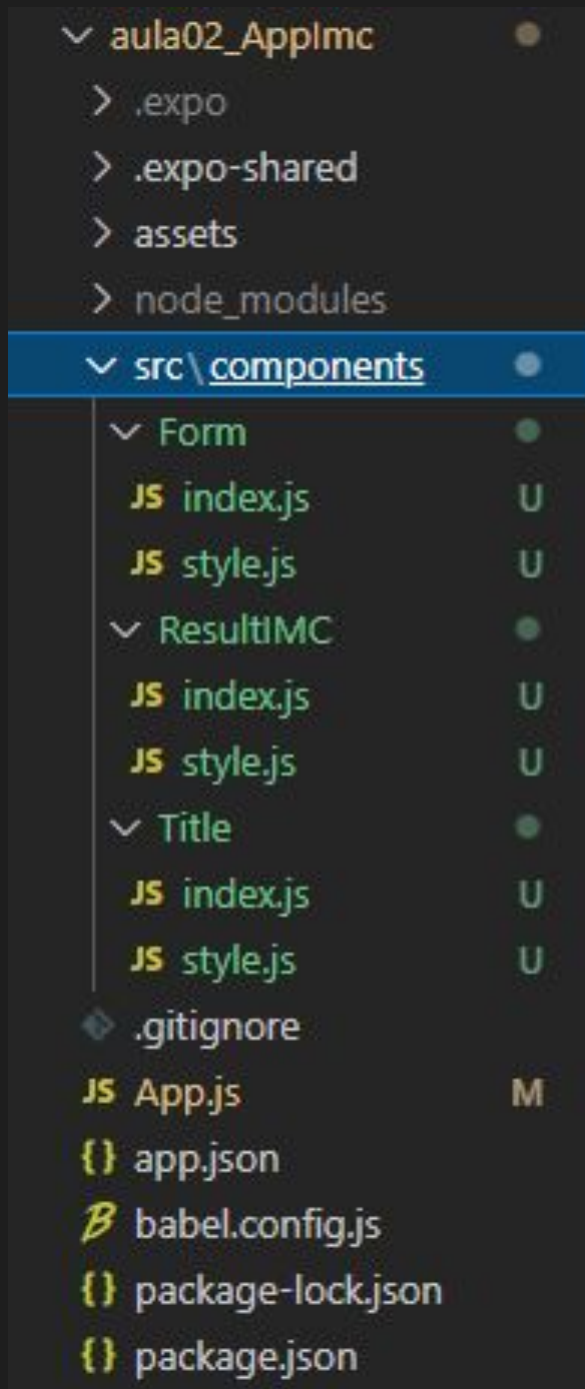
O React Native passa um objeto como style.

É parecido com CSS só não usa o hífen para separar as propriedades.
Usa CamelCase.



Model View Control - MVC

Um arquivo de style.js para cada index.js



```
import { StyleSheet } from "react-native";

const styles = StyleSheet.create({
  Title: {

  },
});

export default styles
```

**Um arquivo de style.js
para cada index.js**



JS App.js M X JS index.js U ●

Testes > aula02_imc > JS App.js > ...

```
1 import React from 'react';
2 import { StyleSheet, Text, View } from 'react-native';
3 import Title from './src/components/title/'
4
5 export default function App() {
6   return (
7     <View style={styles.container}>
8       <Title></Title>
9     </View>
10  );
11 }
12
13 const styles = StyleSheet.create({
14   container: {
15     flex: 1,
16     backgroundColor: '#fff',
17     alignItems: 'center',
18     justifyContent: 'center',
19   },
20 });
21
```

JS App.js M JS index.js U ●

Testes > aula02_imc > src > components > title > JS index.js > ...

```
1 import React from 'react';
2 import { StyleSheet, Text, View } from 'react-native';
3
4 export default function Title() {
5   return (
6     <View>
7       <Text>APP - IMC</Text>
8     </View>
9   );
10 }
11
```

Criar pasta src / component / (nome do componente)



```
import React from 'react';
import { TextInput, Text, View, Button } from 'react-native';

export default function Form() {
  return (
    <View>
      <View>
        <Text>Altura</Text>
        <TextInput></TextInput>

        <Text>Peso</Text>
        <TextInput></TextInput>

        <Button title='Calcular' ></Button>
      </View>
    </View>
  );
}
```





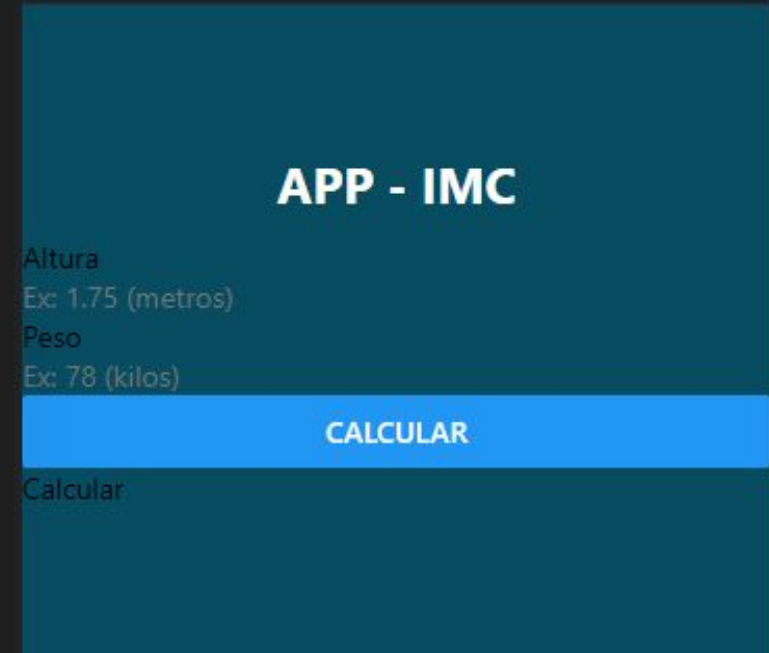
```
import React from 'react';
import { TextInput, Text, View, Button, TouchableOpacity } from 'react-native';

export default function Form() {
  return (
    <View>
      <View>
        <Text>Altura</Text>
        <TextInput
          placeholder="Ex: 1.75 (metros)"
          keyboardType='numeric'
        ></TextInput>
        <Text>Peso</Text>
        <TextInput
          placeholder="Ex: 78 (kilos)"
          keyboardType='numeric'
        ></TextInput>

        <Button title='Calcular' ></Button>

        <TouchableOpacity>
          <Text>Calcular</Text>
        </TouchableOpacity>
      </View>
    </View>
  );
}
```

TouchableOpacity permite mais formatação de estilo que button.





```
import React from 'react';
import { TextInput, Text, View, TouchableOpacity } from 'react-native';
import styles from './style';

export default function Form() {
  return (
    <View style={styles.formContext}>
      <View style={styles.form}>
        <Text style={styles.formLabel}>Altura</Text>
        <TextInput
          style={styles.input}
          placeholder="Ex: 1.75 (metros)"
          keyboardType='numeric'
        ></TextInput>
        <Text style={styles.formLabel}>Peso</Text>
        <TextInput
          style={styles.input}
          placeholder="Ex: 78 (kilos)"
          keyboardType='numeric'
        ></TextInput>

        <TouchableOpacity style={styles.buttonCalculator}>
          <Text style={styles.textButtonCalculator}>Calcular</Text>
        </TouchableOpacity>
      </View>
    </View>
  );
}
```

APP - IMC

Altura

Ex: 1.75 (metros)

Peso

Ex: 78 (kilos)

Calcular



Form/style.js

```
import { StyleSheet } from "react-native";
```

```
const styles = StyleSheet.create({
  formContext: {
    flex: 1,
    backgroundColor: '#fff',
    borderTopLeftRadius: 50,
    borderTopRightRadius: 50,
    alignItems: 'center',
    paddingTop: 30,
    marginTop: 30,
  },
  form: {
    width: '100%',
    height: 'auto',
  },
  formLabel: {
    color: "#084C61",
    fontSize: 20,
    paddingLeft: 20,
  },
  input: {
    width: '90%',
    borderRadius: 50,
    backgroundColor: '#f6f6f6',
    height: 40,
    margin: 12,
    paddingLeft: 10,
  },
```

```
  buttonCalculator: {
    borderRadius: 50,
    alignItems: 'center',
    justifyContent: 'center',
    width: '90%',
    backgroundColor: '#177E89',
    paddingTop: 14,
    paddingBottom: 14,
    marginLeft: 12,
    marginTop: 30,
  },
  textButtonCalculator: {
    fontSize: 20,
    color: '#fff'
  },
  errorMessage: {
    fontSize: 12,
    color: '#DB3A34',
    fontWeight: 'bold',
    paddingLeft: 20,
  },
},
```

APP - IMC

Altura

Ex: 1.75 (metros)

Peso

Ex: 78 (kilos)

Calcular



useState

O hook mais comum utilizado para controlarmos alguma variável de estado dentro de um functional component no React. Para utilizar definimos:

```
const [count, setCount] = useState(0);
```

O primeiro valor count representa o valor do estado que será manipulado pela função setCount recebida através da desestruturação realizada no useState. O valor 0 repassado ao hook é o valor inicial do estado.

Então, para manipularmos o valor de count podemos simplesmente executar:

```
<button onClick={() => setCount(count + 1)}>+</button>
```




React Hooks: useState

useState é uma forma de definir e atualizar (mudar) o estado de um componente para que seja refletido na tela do usuário..

```
import React from 'react';
import { TextInput, Text, View, TouchableOpacity } from 'react-native';
import styles from './style';
import { useState } from 'react';

export default function Form() {

  const [height, setHeight] = useState(null)
  const [weight, setWeight] = useState(null)
  const [messageIMC, setMessageIMC] = useState('Preencha o peso e a altura!')
  const [imc, setImc] = useState(null)
  const [errorMessageHeight, setErrorMessageHeight] = useState(null)
  const [errorMessageWeight, setErrorMessageWeight] = useState(null)
  const [imcList, setImcList] = useState([])
```



```
function imcCalculator() {  
}  
  
function messageErro() {  
}  
  
function validationImc() {  
    //console.warn('imc=' + imc + ' height=' + height + ' weight=' + weight)  
    if (weight != null && height != null) {  
        imcCalculator()  
        setMessageIMC('Seu imc é igual a:')  
        setErrorMessageHeight(null)  
        setErrorMessageWeight(null)  
    } else {  
        messageErro()  
        setImc(null)  
        setMessageIMC('Preencha o peso e a altura!')  
    }  
}  
  
function mostrarFormulario() {  
    setHeight(null)  
    setWeight(null)  
    setErrorMessageHeight(null)  
    setErrorMessageWeight(null)  
    setImc(null)  
}
```



```
function imcCalculator() {
    // Tratando se digita com vírgula - troca (,) por (.)
    let tempHeight = height.replace(',', '.');
    let tempWeight = weight.replace(',', '.');
    let tempIMC = (tempWeight / (tempHeight * tempHeight)).toFixed(2)
    setImc(tempIMC)
    setHeight(null)
    setWeight(null)
}

function messageErro() {
    if (height == null) {
        setErrorMessageHeight('campo obrigatório*')
    } else {
        setErrorMessageHeight(null)
    }
    if (weight == null) {
        setErrorMessageWeight('campo obrigatório*')
    } else {
        setErrorMessageWeight(null)
    }
}
```



```
return (  
  <View style={styles.formContext}>  
    <View style={styles.form}>  
      <Text style={styles.formLabel}>Altura</Text>  
      <Text style={styles.errorMessage}>{errorMessageHeight}</Text>  
      <TextInput  
        style={styles.input}  
        onChangeText={setHeight}  
        placeholder="Exemplo 1.75"  
        keyboardType='numeric'  
      ></TextInput>  
      <Text style={styles.formLabel}>Peso</Text>  
      <Text style={styles.errorMessage}>{errorMessageWeight}</Text>  
      <TextInput  
        style={styles.input}  
        onChangeText={setWeight}  
        placeholder="Exemplo 78.245"  
        keyboardType='numeric'  
      ></TextInput>  
      <TouchableOpacity  
        style={styles.buttonCalculator}  
        onPress={() => validationImc()}  
      >  
        <Text style={styles.textButtonCalculator}>Calcular</Text>  
      </TouchableOpacity>  
    </View>  
  </View>  

```




Dimensões: Sa... 360 x 780 100%

APP - IMC

Altura

1,77

Peso

83

Calcular

1 problema: 1

```
[HMR] Waiting for update log.js:24
signal from WDS...

react-dom.development.js:26244
Download the React DevTools for a
better development experience: http
s://reactjs.org/link/react-devtools

Running application index.js:89
"main" with appParams:
  {rootTag: '#root'}
Development-level warnings: ON.
Performance optimizations: OFF.

▶ imc= 26.49 height=1.77 weight=83
```

Resultado calculando e aparecendo
no console.log.
Agora falta mostrar o resultado na
tela.

17:17 51%

APP - IMC - API

Altura

Exemplo 1.75

Peso

Exemplo 78.245

Calcular

1 2 3 ✕

4 5 6 OK

7 8 9 .-

0 /



ResultIMC/index.js

```
import React from 'react';
import { Text, View, TouchableOpacity } from 'react-native';
import styles from './style';

export default function ResultIMC(props) {
  return (
    <View style={styles.resultImc}>
      <Text style={styles.information}>{props.messageResultIMC}</Text>
      <Text style={styles.numIMC}>{props.resultIMC}</Text>

      <View style={styles.boxButton}>
        <TouchableOpacity
          style={styles.buttonShared}
        >
          <Text style={styles.textButtonShared}>Compartilhar</Text>
        </TouchableOpacity>
      </View>
    </View>
  );
}
```



ResultIMC/style.js

```
import { StyleSheet } from "react-native";
```

```
const styles = StyleSheet.create({  
  resultImc: {  
    flex: 1,  
    marginTop: 20,  
    paddingTop: 15,  
    borderRadius: 50,  
    alignItems: 'center',  
    width: '100%',  
  },  
  information: {  
    fontSize: 18,  
    color: "#084C61",  
  },  
  numIMC: {  
    fontSize: 48,  
    color: "#177E89",  
    fontWeight: 'bold',  
  },  
});
```

```
  boxButton: {  
    width: '100%',  
    alignItems: 'center',  
    marginBottom: 10,  
    marginTop: 30,  
  },  
  buttonShared: {  
    backgroundColor: '#ffc49b',  
    borderRadius: 50,  
    paddingTop: 5,  
    paddingBottom: 5,  
  },  
  textButtonShared: {  
    fontSize: 20,  
    color: '#fff',  
    paddingHorizontal: 30,  
  },  
});  
  
export default styles
```



Form/index.js

```
...
import ResultIMC from '../ResultIMC';
export default function Form() {
  ...

  return (
    <View style={styles.formContext}>
      <View style={styles.form}>
        ...
        </View>
        <ResultIMC
          resultIMC={imc}
          messageResultIMC={messageIMC}
        ></ResultIMC>
      </View>
    );
  }
}
```

APP - IMC

Altura

Exemplo 1.75

Peso

Exemplo 78.245

Calcular

Preencha o peso e a altura!

Compartilhar

Seu imc é igual a:

25.25

Compartilhar



API's

Vamos usar algumas API do celular para melhorar a interação com o usuário.

- vibração
- compartilhamento

APP - IMC

Altura

1.77

Peso

83

Calcular novamente

Seu imc é igual a:

26.49



APIs

AccessibilityInfo

Alert

Animated

Animated.Value

Animated.ValueXY

Appearance

AppRegistry

AppState

DevSettings

Dimensions

Easing

InteractionManager

Keyboard

LayoutAnimation

Linking

PanResponder

PixelRatio

Share

Example

Function Component

Class Component

Function Component Example ⓘ ↗

```
import React from 'react';
import { Share, View, Button } from 'react-native';

const ShareExample = () => {
  const onShare = async () => {
    try {
      const result = await Share.share({
        message:
          'React Native | A framework for building native apps using
React',
      });
    }
    if (result.action === Share.sharedAction) {
      if (result.activityType) {
        // shared with activity type of result.activityType
      } else {
```

Download the image





ResultIMC/index.js

```
import React from 'react';
import { Text, View, TouchableOpacity, Share } from 'react-native';
import styles from './style';

export default function ResultIMC(props) {
  const onShare = async () => {
    const result = await Share.share({
      message: "Meu imc hoje é: " + props.resultIMC,
    })
  }
  return (
    <View style={styles.resultImc}>
      <Text style={styles.information}>{props.messageResultIMC}</Text>
      <Text style={styles.numIMC}>{props.resultIMC}</Text>
      <View style={styles.boxButton}>
        <TouchableOpacity
          style={styles.buttonShared}
          onPress={onShare}
        >
          <Text style={styles.textButtonShared}>Compartilhar</Text>
        </TouchableOpacity>
      </View>
    </View>
  );
}
```



16:18 53%

APP - IMC - API

Altura
campo obrigatório*

1.77

Peso
campo obrigatório*

83

Calcular

1	2	3	✕
4	5	6	OK
7	8	9	.
	0		

APP - IMC - API

Altura

1.77

Peso

83

Calcular novamente

Seu imc é igual a:

26.49

Compartilhar

APP - IMC - API

Altura

—

Meu imc hoje é: 1.25

Copiar

Compartilhar por proximidade

Quick Share

Compartilhe instantaneamente com pessoas próximas. No aparelho da outra pessoa, verifique se Quick Share está ativado no painel rápido. [Dicas](#)

WhatsApp

Instagram Chats

Gmail

Calendário



APIs

AccessibilityInfo

Alert

Animated

Animated.Value

Animated.ValueXY

Appearance

AppRegistry

AppState

DevSettings

Dimensions

Easing

InteractionManager

Keyboard

LayoutAnimation

Linking

PanResponder

PixelRatio

Vibration

Vibrates the device.

Example

Function Component

Class Component

Vibration ⓘ ↗

```
import React from "react";
import { Button, Platform, Text, Vibration, View, SafeAreaView,
  StyleSheet } from "react-native";

const Separator = () => {
  return <View style={Platform.OS === "android" ? styles.separator :
    null} />;
}

const App = () => {
```

Download the image





```
import React from 'react';
import { TextInput, Text, View, TouchableOpacity, Vibration } from 'react-native';

export default function Form() {
  ...

  function messageErro() {
    if (height == null) {
      setErrorMessageHeight('campo obrigatório*')
      Vibration.vibrate()
    } else {
      setErrorMessageHeight(null)
    }
    if (weight == null) {
      setErrorMessageWeight('campo obrigatório*')
      Vibration.vibrate()
    } else {
      setErrorMessageWeight(null)
    }
  }
}
```

Só consegue ver diferença testando no celular.



Form/index.js

```
return (  
  <View style={styles.formContext}>  
    {imc == null ?  
      <View style={styles.form}>  
  
        ...  
  
      </View>  
  
      :  
      <View style={styles.result}>  
        <ResultIMC resultIMC={imc} mensagemResultIMC={mensagemIMC}></ResultIMC>  
  
        <TouchableOpacity  
          style={styles.buttonCalculador}  
          onPress={() => mostrarFormulario()}  
        >  
          <Text style={styles.textButtonCalculador}>Calcular novamente</Text>  
        </TouchableOpacity>  
  
      </View>  
    }  
  </View>  
)  
);
```

Alterar para mostrar só o formulário ou o resultado.
O Resultado dentro de uma view.
Criamos um botão para limpar o formulario.



Form/style.js

```
Result: {  
  width: '100%',  
  height: '50%',  
},  
FlatList: {  
  margin: 20,  
  width: '90%',  
},  
FlatListItem: {  
  fontSize: 16,  
  color: '#177E89',  
  backgroundColor: "#E6E9ED",  
  height: 40,  
  width: '100%',  
  padding: 10,  
  marginTop: 2,  
}  
});
```





Form/index.js

```
import React from 'react';
import { TextInput, Text, View,
  TouchableOpacity, Vibration, FlatList } from 'react-native';

function ImcCalculator() {
  // Tratando se digita com vírgula - troca (,) por (.)
  let tempHeight = height.replace(',', '.');
  let tempWeight = weight.replace(',', '.');
  let tempIMC = (tempWeight / (tempHeight * tempHeight)).toFixed(2);
  setImc(tempIMC);
  console.warn('imc= ' + tempIMC + ' height=' + tempHeight + ' weight=' + tempWeight);
  setHeight(null);
  setWeight(null);

  // ListView foi substituído pelo FlatList no React Native, mais eficiente
  // Carrega só os itens que cabem na tela, cria um scroll automático,
  // FlatList exige passar um ID único, no exemplo pegar a data em milissegundos
  // Adicionar no array um item com um JSON contendo id e imc
  let tempId = new Date().getTime();
  setImcList((arr) => [...arr, { id: tempId, imc: tempIMC }])
}
```



Form/index.js



```
<FlatList
  style={styles.FlatList}
  data={imcList}
  keyExtractor={({item, index}) => index.toString()}
  renderItem={({ item }) => {
    return (
      <Text style={styles.FlatListItem}>IMC: {item.imc}</Text>
    )
  }}
/>

</View>
);
}
```



APP - IMC

Seu imc é igual a:

27.76

Compartilhar

Calcular novamente

IMC: 27.76

APP - IMC

Seu imc é igual a:

26.04

Compartilhar

Calcular novamente

IMC: 27.76

IMC: 24.38

IMC: 23.10

IMC: 26.04

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

Obrigado!



Prof. Dr. Juliano Gaspar
julianogaspar@gmail.com

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

Prof. Juliano Gaspar