

# Programação para Dispositivos Móveis

Prof<sup>a</sup> Letícia Pieper Jandt

80 Horas



# FrameWork React-Native. Componentes, Pages, Expo, NPM e NodeJS.

# O que é um framework?

Um **framework** em desenvolvimento de software, é uma abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica. Um **framework** pode atingir uma funcionalidade específica, por configuração, durante a programação de uma aplicação.

# O que é um framework?

Ao contrário das bibliotecas, é o *framework* quem dita o fluxo de controle da aplicação



Como um ambiente de execução JavaScript assíncrono orientado a eventos, o Node.js é projetado para desenvolvimento de aplicações escaláveis de rede.

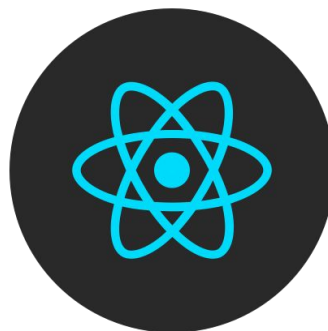
<https://nodejs.org/pt-br/>

# React Native

React Native combina a melhor parte do desenvolvimento nativo com react. O melhor da biblioteca *javascript* para construir a interface do usuário.

Escrito em JavaScript, compilado em código nativo!

<https://reactnative.dev/>



## Expo

Expo é um framework e uma plataforma para desenvolvimento de aplicações React. É um conjunto de ferramentas e serviços construídos com React Native e plataformas nativas que ajudam você a desenvolver, construir, compilar e publicar na plataforma iOS, Android e Web Apps com os mesmos códigos JavaScript.

<https://expo.io/>

# Documentações

<https://reactnative.dev/docs/getting-started>

<https://docs.expo.io/versions/latest/>

<https://pt-br.reactjs.org/docs/getting-started.html>

# Iniciando com o Expo

1- Instalar o Node.js

2- rodar o comando de instalar o expo

**`npx install-expo-modules@latest`**

**ou `npm install expo`**

3- instalar o package de criar projetos

`npm install create-expo-app`

# Iniciando com o Expo

1-cd caminho da pasta onde vai ser criado o app

3- npx create-expo-app my-app --template

*onde my-app é o nome do app*

4 - Escolher o tipo do projeto:

```
C:\Users\letic\Documents\PESSOAL\PIEPER04\FASIPE\Dispositivos Móveis\2024>npx create-expo-app app2024 --template
Need to install the following packages:
create-expo-app@3.0.0
? Choose a template: » - Use arrow-keys. Return to submit.
> Default - includes tools recommended for most app developers
  Blank
  Blank (TypeScript)
  Navigation (TypeScript)
  Blank (Bare)
```



# Iniciando com o Expo

```
found 0 vulnerabilities

📄 Your project is ready!

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

- cd app2024
- npm run android
- npm run ios # you need to use macOS to build the iOS project - use the Expo app if you need
out a Mac
- npm run web
```

5- abrir a pasta do app: `cd app2024`

# VAMOS TESTAR! (redes diferentes)

6 – npm install @expo/ngrok

7 – Alterar o package.json alterando o script start

"start": "expo start --tunnel",

8– npm start

# VAMOS TESTAR! (mesma rede)

6 – expo start ou npm start

estar logado na conta expo ou pelo menos na mesma rede!

\*Caso tenhamos problema de debuggar no celular, vamos fazer o start via ip!

set REACT\_NATIVE\_PACKAGER\_HOSTNAME= seu ip  
depois npm start

# Estrutura

## **/node\_modules**

Todos os pacotes que estamos utilizando no projeto. É Gerenciado pelo NPM ou pelo Yarn, e ainda, no nosso caso, pelo expo.

## **App.json**

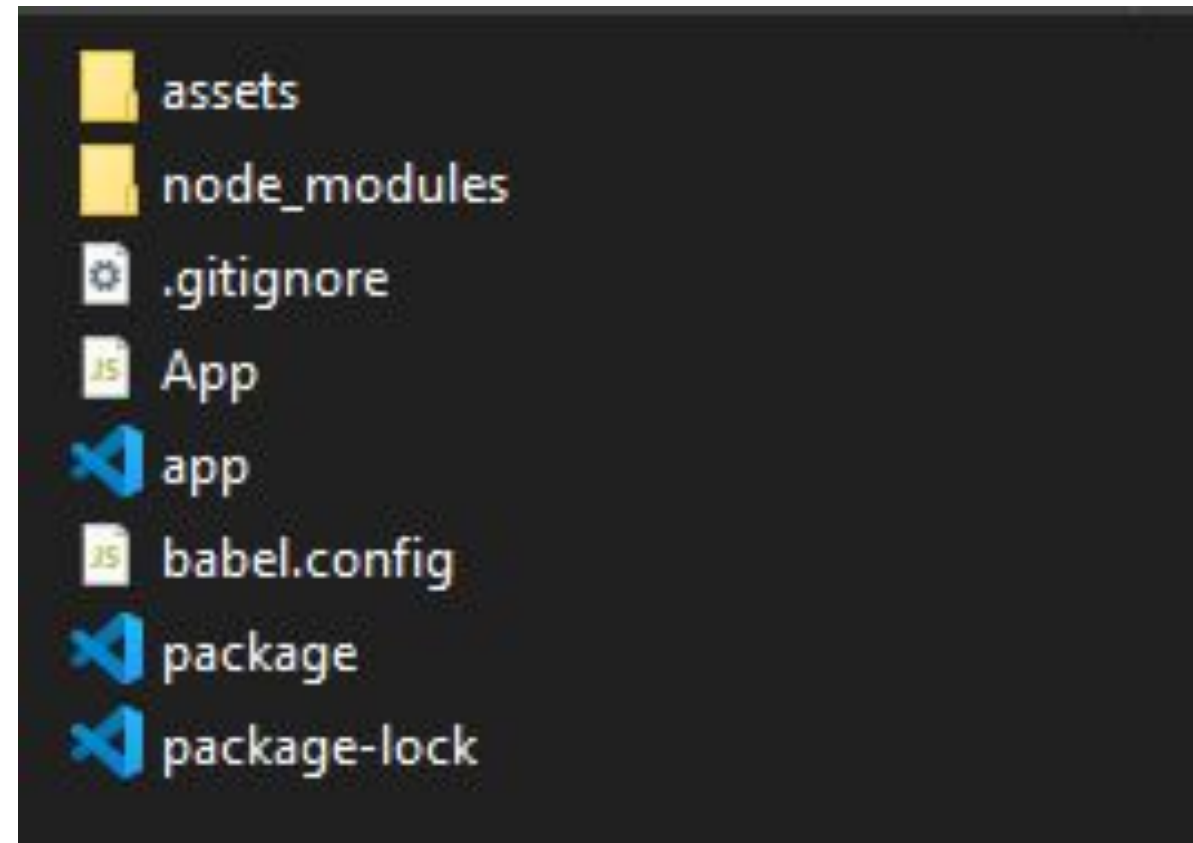
É o Arquivo que diz como nossa aplicação será mostrada para o usuário, assim como nome da loja etc.

## **App.js**

Arquivo inicial do projeto.

## **package.json**

Configuração das dependências do projeto.



# Estrutura

JS App.js > ...

```
1  import { StatusBar } from 'expo-status-bar';
2  import { StyleSheet, Text, View } from 'react-native';
3
4  export default function App() {
5    return (
6      <View style={styles.container}>
7        <Text>Open up App.js to start working on your app!</Text>
8        <StatusBar style="auto" />
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  });
```

Importações de Componentes

Export default, diz o que será enviado para visualização do usuário

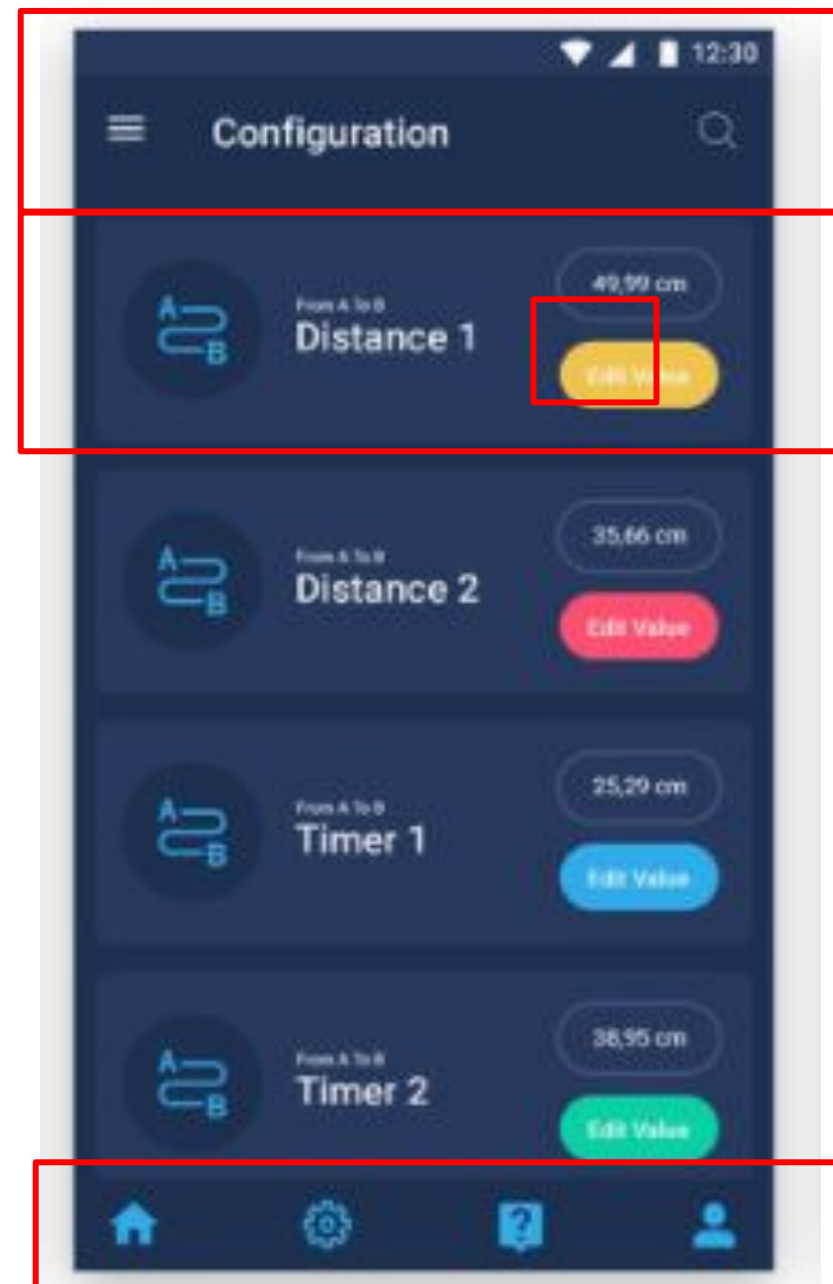
Styles são os estilos, como um .css da web

# Componentes

É a Modularização do Código. Basicamente é um conjunto de códigos que é separado entre parte visual e funcional, para ser reutilizado em diversas partes do código.

Quando organizado, os arquivos das classes de componente ajudam a ter uma melhor organização do código.

Tudo em nossa aplicação, será um componente.



# Componentes

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

Variável com nome do  
nosso Componente no app

Local onde esse  
componente está.

OBS: A biblioteca react-native possui muito componentes dentro, ou seja, não há um componente “padrão dentro dela. Por isso usa-se o próprio nome para identificar o que está se importando”

# Componentes

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

```
<View>  
</View>
```

A View é uma “Caixa Vazia”.

React native tem uma regra que  
Nunca pode haver mais de um elemento  
“solto” dentro de um componente.



# StyleSheet

```
const styles = StyleSheet.create({  
  container: {  
    flex: 1,  
    alignItems: 'center',  
    justifyContent: 'center',  
  },  
});
```

É UM  
OBJETO  
JSON =)

# Consumindo dados de uma API

Vamos revisar um pouco HTTP.

# Consumindo dados de uma API

GET <https://viacep.com.br/ws/78557730/json/> Send 200 OK 637 ms 236 B

Body Auth Query Headers Docs

Add Delete All Toggle Description

Name	Value
------	-------

```
1 {  
2   "cep": "78557-730",  
3   "logradouro": "Rua das Primaveras",  
4   "complemento": "até 515/516",  
5   "bairro": "Jardim Jacarandás",  
6   "localidade": "Sinop",  
7   "uf": "MT",  
8   "ibge": "5107909",  
9   "gia": "",  
10  "ddd": "66",  
11  "siafi": "8985"  
12 }
```

Verbos,  
StatusCode,  
conexões.

NAME	VALUE
Server	nginx/1.22.0
Date	Fri, 18 Aug 2023 14:18:31 GMT
Content-Type	application/json; charset=utf-8
Transfer-Encoding	chunked
Connection	keep-alive
Expires	Fri, 18 Aug 2023 15:18:31 GMT
Cache-Control	max-age=3600
Pragma	public
Cache-Control	public
Access-Control-Allow-Origin	*
Access-Control-Allow-Methods	GET
Access-Control-Allow-Headers	Content-Type, X-Request-With, X-Requested-By
Access-Control-Allow-Credentials	true
Access-Control-Max-Age	86400

# Estados - State

Estado é responsável por armazenar TUDO o que é manipulado dentro de nossa classe.

Não se cria Variáveis armazenando dados por exemplo.

# Estados - State

Outra característica do estado, é que o react fica “ouvindo” se há alterações em alguma informação contida no mesmo.

# Estados - State

Toda propriedade dentro do state, precisa passar pro procedimentos GETS e SETS.

# useEffect

O `useEffect` é um Hook que permite executar efeitos colaterais em seus componentes. Alguns desses efeitos colaterais podem ser: busca de dados em uma API, atualizar diretamente o VirtualDOM e controlar temporizadores (`setTimeout` e `setInterval`).

no cmd

- npm install axios



```
1 | import axios from 'axios';
2 | import { StatusBar } from 'expo-status-bar';
3 | import { useEffect, useState } from 'react';
4 | import { StyleSheet, Text, View } from 'react-native';
5 |
6 | export default function App() {
7 |   const [MeuEnd, setMeuEnd] = useState([]);
8 |
9 |   useEffect(() => {
10 |     getCep();
11 |   }, []);
12 |
```

```
13  async function getCep() {
14      try {
15          await axios
16              .get(`https://viacep.com.br/ws/78555000/json/`, {
17                  headers: {
18                      "Content-Type": "application/json",
19                  },
20              })
21              .then((response) => {
22                  console.log(response);
23                  setMeuEnd(response.data)
24              })
25              .catch((error) => {
26                  console.log(error);
27              });
28      } catch (erro) {
29          console.log(erro);
30      }
31  }
32
```

```
33
34   return (
35     <View style={styles.container}>
36       <Text>Meu endereço é: { MeuEnd.logradouro }</Text>
37       <StatusBar style="auto" />
38     </View>
39   );
40 }
41
42 const styles = StyleSheet.create({
43   container: {
44     flex: 1,
45     backgroundColor: '#fff',
46     alignItems: 'center',
47     justifyContent: 'center',
48   },
49 });
```

Vamos testar

# TextInput

Campo de texto - inserção de dados.



```
1 | import axios from 'axios';
2 | import { StatusBar } from 'expo-status-bar';
3 | import { useEffect, useState } from 'react';
4 | import { Button, ProgressBarAndroid, StyleSheet, Text, TextInput, View } from 'react-native';
5 |
6 | export default function App() {
7 |   const [MeuCep, setMeuCep] = useState("");
8 |   const [MeuEnd, setMeuEnd] = useState([]);
9 |   const [Load, setLoad] = useState(false);
10 |
11 |   useEffect(() => {
12 |
13 |     }, [MeuEnd]);
14 |
15 |   const onChangeCEP = (cep) => {
16 |     setMeuCep(cep);
17 |   }
18 |
```

```
19  async function getCep() {
20      setLoad(true);
21      setMeuEnd([]);
22
23      try {
24          await axios
25              .get(`https://viacep.com.br/ws/${ MeuCep }/json/`, {
26                  headers: {
27                      "Content-Type": "application/json",
28                  },
29              })
30              .then((response) => {
31                  console.log(response);
32                  setMeuEnd(response.data)
33                  setLoad(false);
34              })
35              .catch((error) => {
36                  console.log(error);
37                  setLoad(false);
38              });
39      } catch (erro) {
40          console.log(erro);
41      }
42  }
43
```

```
45 return (
46   <View style={styles.container}>
47     <Text>Digite seu cep:</Text>
48     <TextInput
49       style={styles.input}
50       onChangeText={CEP => onChangeCEP(CEP)}
51       value={MeuCep}
52       placeholder="CEP"
53       keyboardType="numeric"
54       maxLength={8}
55     />
56     { Load ? (
57       <View>
58         <ProgressBarAndroid />
59       </View>
60     ) : null }
61
62     <Button
63       title="Pesquisar CEP"
64       onPress={() => getCep() }
65     />
66
67     <Text style={{ marginTop: 10 }}>Meu endereço é: { MeuEnd.logradouro }</Text>
68     <StatusBar style="auto" />
69   </View>
70 );
71 }
```



```
73 const styles = StyleSheet.create({
74   container: {
75     flex: 1,
76     backgroundColor: '#fff',
77     alignItems: 'center',
78     justifyContent: 'center',
79   },
80   input: {
81     height: 40,
82     width: 100,
83     margin: 12,
84     borderWidth: 1,
85     padding: 10,
86   }
87 });
```

Mostre todo o endereço na tela

mude o layout, seja feliz