

**Especialização em Desenvolvimento Full Stack**

# **React-Native**

**Prof. Esp. Marco Antônio Gusmão Carvalho**

# Planejamento

**04/06**

- Introdução ao React Native
- Expo
- Layout e Navegação da Aplicação

**11/06**

- Context API
- Redux

**18/06**

- Dados e Recursos
- Projeto Final

## Agenda do dia

### Manhã

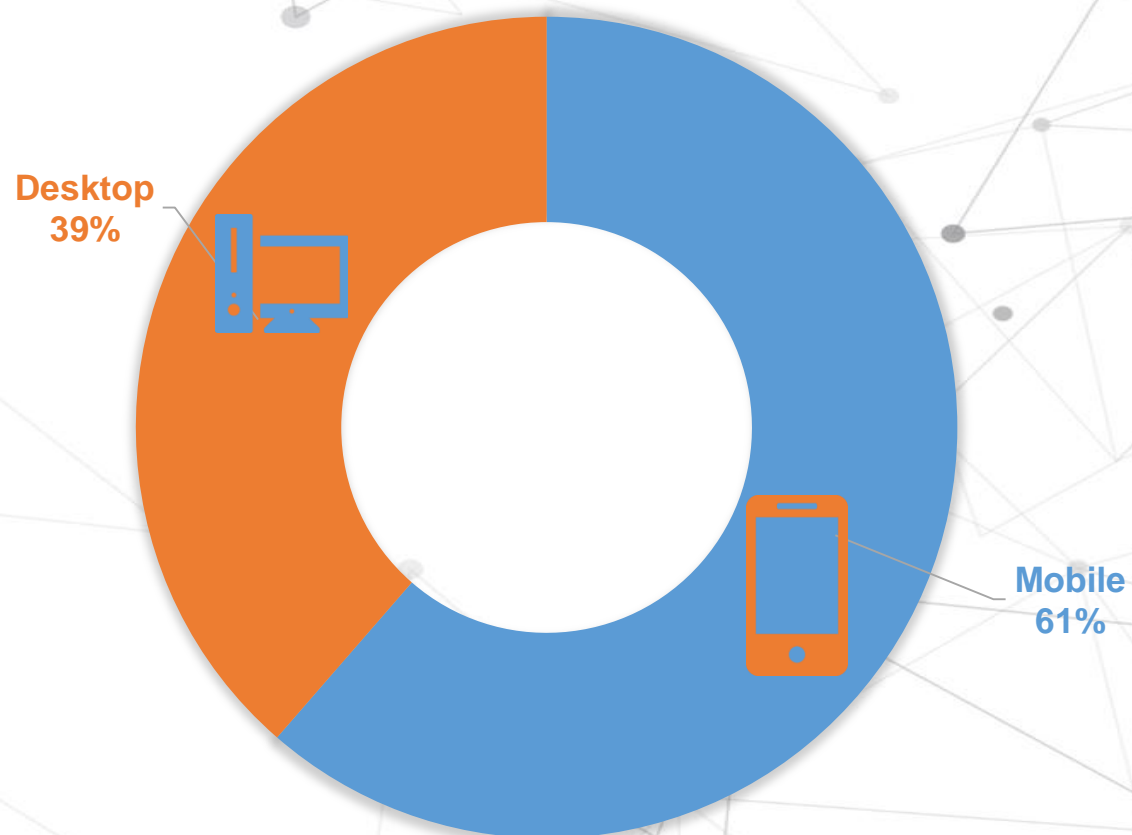
- Mercado Mobile e Modelos de Desenvolvimento
- O Que é o React-Native
- Ferramental
  - Expo
  - React-Native CLI
- Criando a primeira aplicação
- Estruturas Básicas da Aplicação

### Tarde

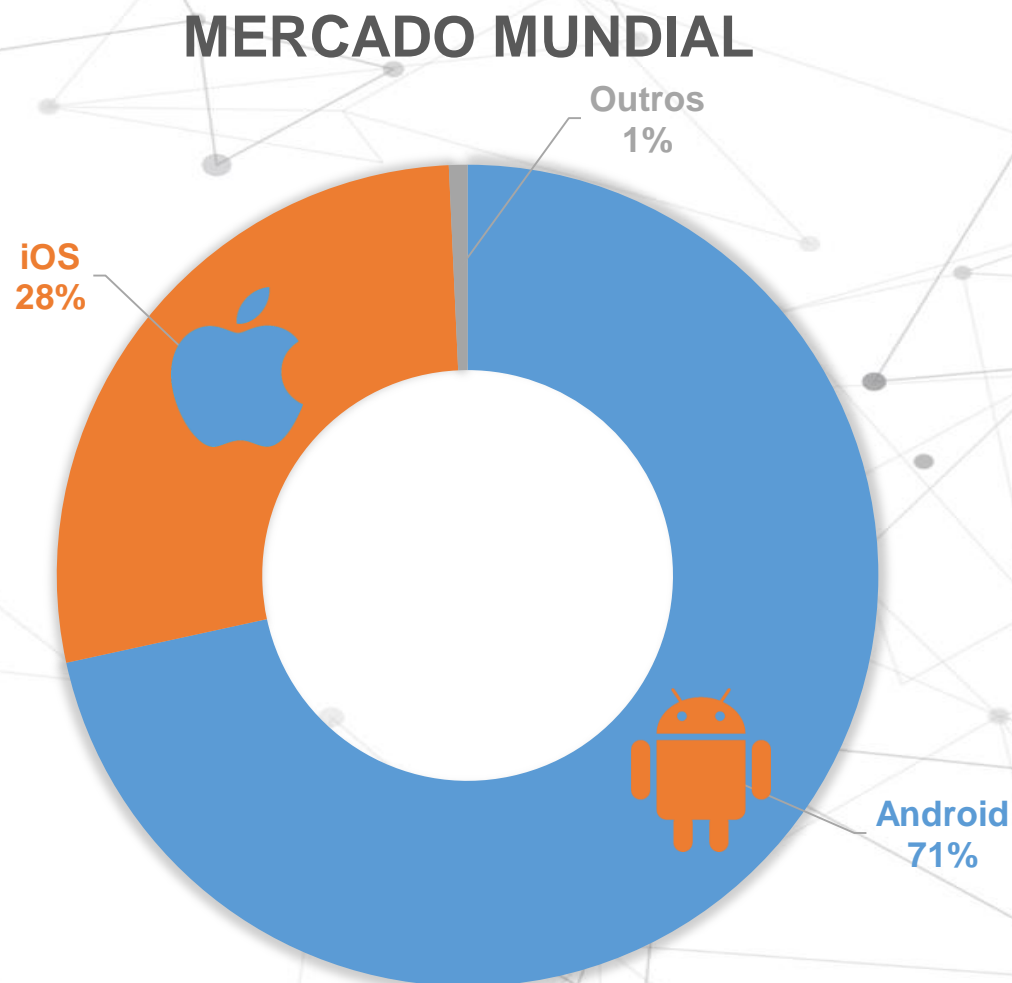
- Componentes Básicos
- Paginação

## Mercado

### MERCADO MUNDIAL



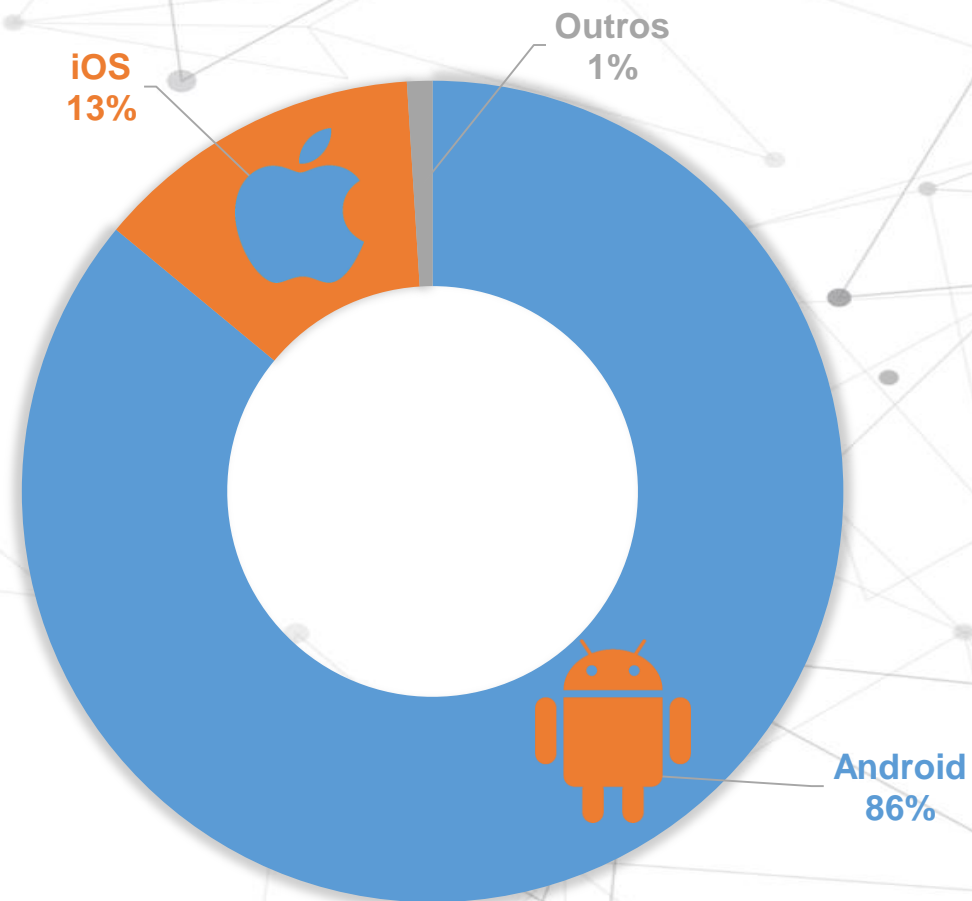
# Mercado Mobile





# Mercado Mobile

### MERCADO BRASILEIRO



## *Desenvolvimento Nativo*



## *Desenvolvimento Nativo Multiplataforma*



React Native



## *Desenvolvimento Web Híbrido*



APACHE  
CORDOVA™

# O que é ReactJS

- Disponibilizado em 2015 pela equipe do Facebook sob licença MIT tendo surgido alguns anos antes em um Hackaton Interno da empresa.
- Biblioteca Baseada em React para desenvolvimento de aplicações Mobile Nativas.
- Possui versões para desenvolvimento Windows e MacOS
- Possui versão para desenvolvimento web
- Possui grande aceitação no mercado



Walmart



TESLA

Uber Eats

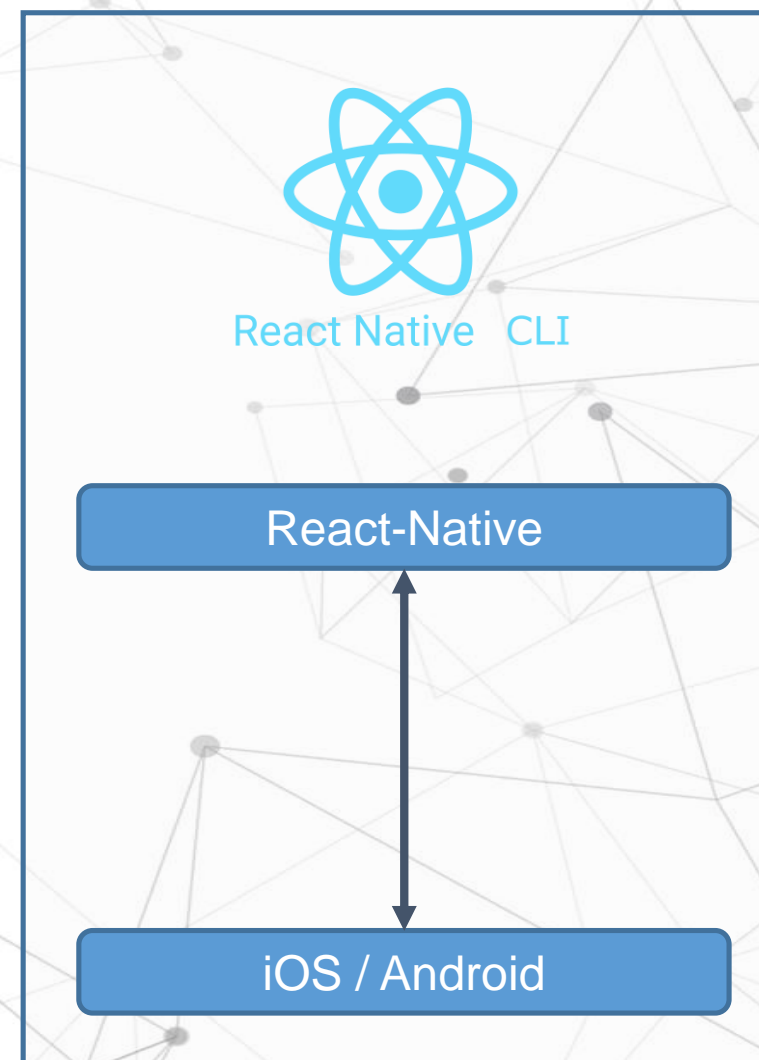
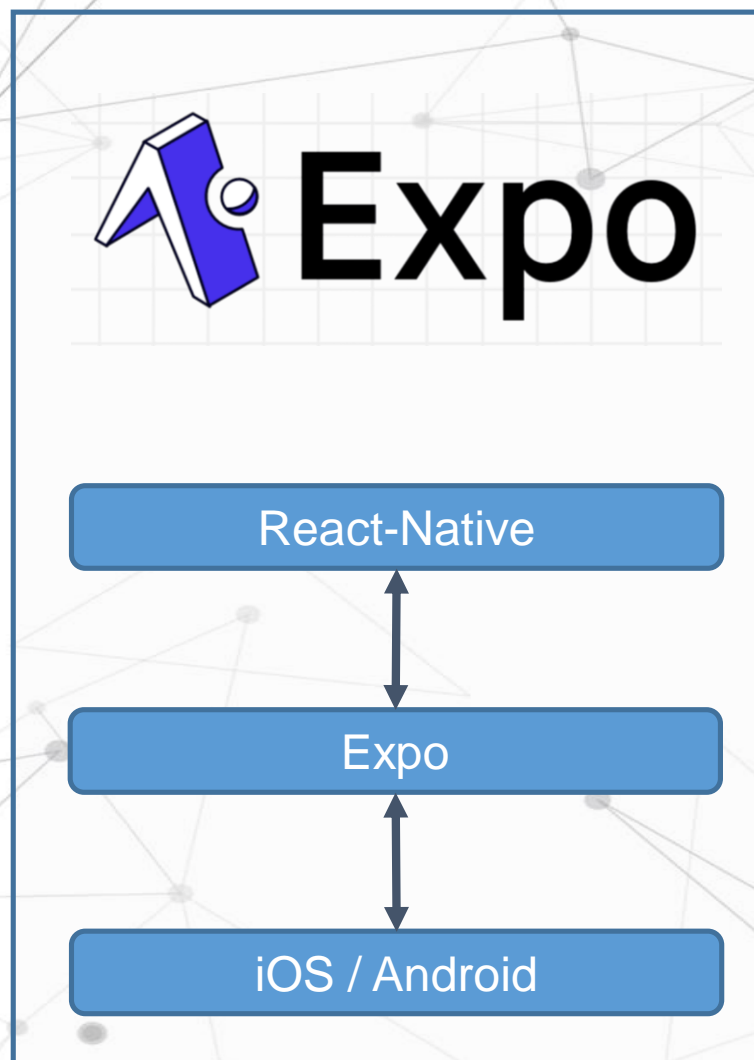
WIX



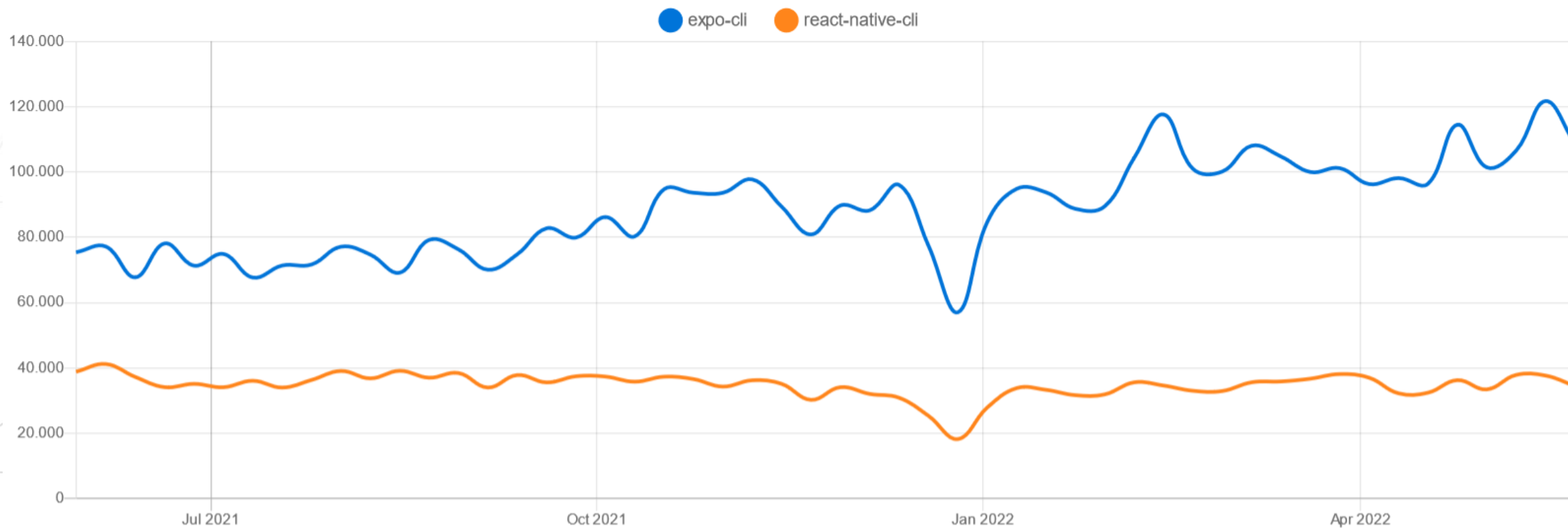
airbnb



## Ferramentas



# Especialização em Desenvolvimento Full Stack

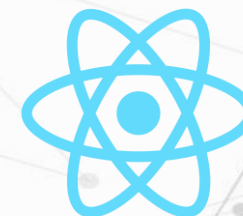


## Ferramentas



```
npm install -g expo-cli
```

```
expo init projeto
```



React Native CLI

```
npm install -g react-native-cli
```



Android  
Studio



```
react-native init projectrn
```

## **Passo a Passo**

### **Criando uma aplicação react native com expo**

## Projeto Expo

```
Projects expo init expo-first-app
Choose a template: » - Use arrow-keys. Return to submit.
----- Managed workflow -----
blank a minimal app as clean as an empty canvas
blank (TypeScript) same as blank but with TypeScript configuration
tabs (TypeScript) several example screens and tabs using react-navigation and TypeScript
----- Bare workflow -----
minimal bare and minimal, just the essentials to get you started
```

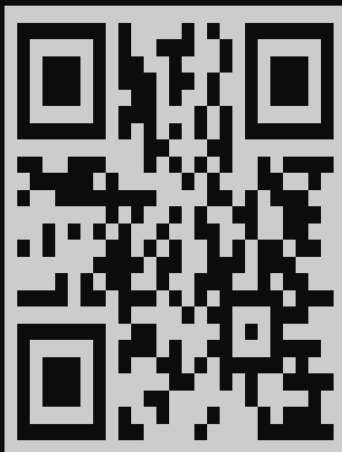
```
Projects cd expo-first-app
expo-first-app git:(master) code .
```

```
expo-first-app git:(master) expo start
```



## Projeto Expo

```
Starting project at C:\Users\marco\Projects\expo-first-app  
Developer tools running on http://localhost:19002  
Starting Metro Bundler
```



```
> Metro waiting on exp://172.16.0.134:19000  
> Scan the QR code above with Expo Go (Android) or the Camera app (iOS)  
  
> Press a | open Android  
> Press w | open web  
  
> Press r | reload app  
> Press m | toggle menu  
> Press d | show developer tools  
> shift+d | toggle auto opening developer tools on startup (disabled)  
  
> Press ? | show all commands
```

```
Logs for your project will appear below. Press Ctrl+C to exit.  
Started Metro Bundler
```

## Expo Developer Tools

Precisa ter instalado a respectiva SDK

### Metro Bundler

PROCESS (2) - 11:14:47 AM

Run on Android device/emulator

Run on iOS simulator

Run in web browser

Send link with email...

Publish or republish project...

PRODUCTION MODE

CONNECTION

Tunnel LAN Local

exp://172.16.0.134:19000



LOGGED IN AS

METRO BUNDLER

INFO Starting Metro Bundler  
11:14

INFO Started Metro Bundler  
11:14

App Expo em  
Dispositivo (LAN)

App Expo em  
Emulador (PC)

App Expo em  
Dispositivo (WAN)

App.js

```
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 });
```

## Criando Telas

screens > JS Home.js > ...

```
1  import { StatusBar } from 'expo-status-bar';
2  import { Text, View } from 'react-native';
3  import { Button } from 'react-native';
4  import styles from './styles';
5
6  export default function Home({navigation}) {
7    return (
8      <View style={styles.container}>
9        <Text>Home</Text>
10       <StatusBar style="auto" />
11       <Button title='Sobre' onPress={() => {navigation.navigate('Sobre')}} />
12     </View>
13   );
14 }
```

## Criando Telas

```
screens > Js About.js > ...  
1  import { StatusBar } from 'expo-status-bar';  
2  import { Text, View } from 'react-native';  
3  import styles from './styles';  
4  
5  export default function About() {  
6    return (  
7      <View style={styles.container}>  
8        <Text>Sobre</Text>  
9        <StatusBar style="auto" />  
10     </View>  
11   );  
12 }
```



## ReactNavigation

```
npm install @react-navigation/native
```

```
expo install react-native-screens react-native-safe-area-context
```

```
npm install @react-navigation/native-stack
```

## Navegando entre as Telas

```
JS App.js > ...
1  import * as React from 'react';
2  import { View, Text } from 'react-native';
3  import { NavigationContainer } from '@react-navigation/native';
4  import { createNativeStackNavigator } from '@react-navigation/native-stack';
5
6  import Home from './screens/Home';
7  import About from './screens/About';
8
9  const Stack = createNativeStackNavigator();
10
11  function App() {
12    return (
13      <NavigationContainer>
14        <Stack.Navigator initialRouteName="Home">
15          <Stack.Screen name="Home" component={Home} options={{ title: 'Primeiro App' }} />
16          <Stack.Screen name="Sobre" component={About} />
17        </Stack.Navigator>
18      </NavigationContainer>
19    );
20  }
21
22  export default App;
```

## Core Components

View

Text

Image

TextInput

ScrollView

StyleSheet

Button

Switch

FlatList

SectionList

## Exercício

Desenvolva um App com React Native que possua 3 telas com navegação entre Elas

**Especialização em Desenvolvimento Full Stack**

# **React-Native**

**Prof. Esp. Marco Antônio Gusmão Carvalho**