

**Vitor Amadeu Souza**

Introdução ao

# **React Native**

No ambiente Expo Snack

Parte VII

© 2024 by Cerne Tecnologia e Treinamento Ltda.

© 2024 by Vitor Amadeu Souza

Nenhuma parte desta publicação poderá ser reproduzida sem autorização prévia e escrita de **Cerne Tecnologia e Treinamento Ltda.** Este livro publica nomes comerciais e marcas registradas de produtos pertencentes a diversas companhias. O editor utiliza as marcas somente para fins editoriais e em benefício dos proprietários das marcas, sem nenhuma intenção de atingir seus direitos.

**Outubro de 2024**

Direitos reservados por:

Cerne Tecnologia e Treinamento Ltda

*Produção: Cerne Tecnologia e Treinamento*

*E-mail da Empresa: cerne@cerne-tec.com.br*

*Home Page: www.cerne-tec.com.br.com.br*

*Atendimento ao Consumidor: sac@cerne-tec.com.br*

*Contato com o Autor: vitor@cerne-tec.com.br*



**FEITO NO BRASIL**

***“Venha também sobre mim a tua benignidade, ó Senhor, e a tua  
salvação, segundo a tua palavra.”***

**Sl 119:41**

## **Cerne Tecnologia**

A Cerne Tecnologia tem uma equipe preparada para desenvolvimento de projetos eletrônicos em diversas áreas: Médica, Entretenimento, Industrial, Robótica, Científica, Automobilística, Aeronáutica, etc. Trabalhamos com tecnologia microcontrolada usando o PIC, ARM, AVR, 8051, dsPIC, PIC24, PIC32 além do Arduino, Raspberry, Beaglebone etc. Desenvolvemos o projeto desde sua concepção até a entrega do produto final, passando pelas etapas de esquema elétrico, protótipo e desenvolvimento de circuito impresso.

Desenvolvemos aplicativos para smartphones/tablets Android, iOS, Blackberry, Windows Phone e no desenvolvimento de softwares a nível PC para plataforma Windows, usando ferramentas como o Visual Basic, C# e C++.

Atuamos na parte de montagem de placas, onde podemos fornecer ambos os serviços de desenvolvimento de projetos e produção ou apenas um destes.

Desenvolvemos esquemas elétricos e layout de PCI, tanto em tecnologia convencional como SMD.

Temos a flexibilidade de customizar um de nossos produtos, de modo a atender a uma necessidade específica do cliente, tornando o custo de desenvolvimento menor se comparado a construção de um projeto desde a sua fase inicial.

Desenvolvemos e fornecemos kits didáticos para diversos microcontroladores além de apostilas, livros e e-books.

Na hora de desenvolver um projeto ou equipar seu laboratório não hesite em nos contatar. Entre em contato conosco através do endereço [cerne-tec.com.br](http://cerne-tec.com.br) para obter mais informações.



# Sumário

<b>Capítulo I – Metodologia de desenvolvimento.....</b>	<b>6</b>
1. Introdução.....	6
<b>Capítulo II – Programação em React Native.....</b>	<b>7</b>
1. Componente NavigationContainer.....	7
2. Componente useWindowDimensions.....	13
3. Exemplo Dark Mode.....	17

# Capítulo I

## Metodologia de desenvolvimento

### 1. Introdução

Neste livro, você encontrará uma parte da série dedicada ao desenvolvimento de projetos com React Native. Ao longo desta obra, diversos experimentos serão apresentados conforme o sumário, permitindo um aperfeiçoamento contínuo no uso deste framework para a construção de interfaces responsivas com foco em dispositivos móveis.

# Capítulo II

## Programação em React Native

### 1. Componente NavigationContainer

O componente `NavigationContainer` é essencial na biblioteca `React Navigation`, utilizada em aplicações `React Native` para gerenciar a navegação entre telas. Ele atua como um contêiner que mantém o estado da navegação, devendo envolver toda a aplicação, permitindo que as funcionalidades de navegação se apliquem a todos os componentes filhos. Uma de suas principais características é o gerenciamento de estado, que facilita a transição entre telas e a administração do histórico de navegação. Além disso, o `NavigationContainer` permite o aninhamento de diferentes tipos de navegadores, como pilhas de navegação dentro de navegadores de abas.

O componente aceita diversas propriedades de configuração, como `onStateChange`, que aciona uma função sempre que o estado da navegação muda. Também suporta `deep linking`, possibilitando que a navegação comece a partir de URLs externas. Para

implementar a navegação programática, oferece funções como `navigate` e `goBack`, que podem ser utilizadas em resposta a eventos dentro do aplicativo. Em um exemplo básico, o `NavigationContainer` pode ser usado junto com um `stack navigator` para definir as telas da aplicação, como uma tela inicial e uma tela de detalhes, permitindo ao usuário navegar entre elas. Essa estrutura é fundamental para criar experiências de usuário dinâmicas e ricas em aplicativos `React Native`.

```
import React from 'react';
import { Button, View, Text, StyleSheet } from 'react-native';
import { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from '@react-navigation/native-stack';

const Stack = createNativeStackNavigator();

const HomeScreen = ({ navigation }) => {
  return (
    <View style={styles.container}>
      <Text style={styles.title}>Página Principal</Text>
      <Button
        title="Ir para Página 1"
        onPress={() => navigation.navigate('Page1')}
      />
      <Button
        title="Ir para Página 2"
        onPress={() => navigation.navigate('Page2')}
      />
    </View>
  );
};

const Page1 = ({ navigation }) => {
  return (
    <View style={styles.container}>
      <Text style={styles.title}>Página 1</Text>
    </View>
  );
};
```

```

        <Button title="Voltar" onPress={() => navigation.goBack()} />
    </View>
  );
};

const Page2 = ({ navigation }) => {
  return (
    <View style={styles.container}>
      <Text style={styles.title}>Página 2</Text>
      <Button title="Voltar" onPress={() => navigation.goBack()} />
    </View>
  );
};

const App = () => {
  return (
    <NavigationContainer>
      <Stack.Navigator initialRouteName="Home">
        <Stack.Screen name="Home" component={HomeScreen} />
        <Stack.Screen name="Page1" component={Page1} />
        <Stack.Screen name="Page2" component={Page2} />
      </Stack.Navigator>
    </NavigationContainer>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#f0f0f0',
  },
  title: {
    fontSize: 24,
    marginBottom: 20,
  },
});

export default App;

```

O código apresentado é um exemplo de um aplicativo React Native que utiliza a biblioteca `@react-navigation/native` para

implementar a navegação entre diferentes telas. O aplicativo possui uma tela inicial chamada "Home" e duas outras telas, "Page1" e "Page2".

O código começa com as importações necessárias, incluindo React, Button, View, Text, StyleSheet do React Native, além dos componentes de navegação da biblioteca. A função `createNativeStackNavigator` é utilizada para criar um stack navigator, que gerencia a navegação entre as telas.

A tela inicial, chamada `HomeScreen`, exibe um título "Página Principal" e contém dois botões. Cada botão, ao ser pressionado, navega para uma das páginas (`Page1` ou `Page2`) utilizando a função `navigation.navigate`. A `Page1` exibe o título "Página 1" e um botão "Voltar", que permite retornar à tela anterior com a função `navigation.goBack`. A `Page2` tem uma estrutura semelhante, exibindo o título "Página 2" e também um botão "Voltar".

O componente principal do aplicativo, chamado `App`, encapsula o stack navigator dentro de um `NavigationContainer`, que é necessário para gerenciar o estado da navegação. O stack navigator é configurado com três telas: `Home`, `Page1` e `Page2`, sendo a `Home` definida como a tela inicial.