

Vitor Amadeu Souza

Introdução ao

React Native

No ambiente Expo Snack

Parte XV

© 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 para obter mais informações.



Sumário

Capítulo I – Metodologia de desenvolvimento.....	6
1. Introdução.....	6
Capítulo II – Programação em React Native.....	7
1. Contador de toques.....	7
2. MapView.....	10
3. Conexão com API.....	12
4. Quiz.....	15

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. Contador de toques

O código apresentado é uma aplicação de contador de toques em React Native. A aplicação permite que o usuário conte quantas vezes um botão foi pressionado, exibindo o total de toques na tela.

A estrutura do código começa com a importação das bibliotecas necessárias, incluindo React e useState, além de componentes do react-native, como View, Text, Button e StyleSheet. O componente principal é chamado TapCounter, onde um estado chamado count é definido para armazenar o número de toques.

Dentro do componente, a interface é composta por uma View que centraliza o conteúdo. Um texto exibe a contagem atual de toques, e um botão com o título "Tocar" chama a função que incrementa o contador em um ao ser pressionado. A contagem é atualizada em tempo real, refletindo imediatamente na interface.

Os estilos são definidos usando StyleSheet, proporcionando uma aparência organizada à aplicação. O contêiner é configurado

para ocupar toda a tela e centralizar os elementos, enquanto o texto que exibe a contagem é estilizado com um tamanho de fonte grande para fácil visualização.

Desta forma, este código exemplifica como criar uma aplicação de contador de toques em React Native, utilizando conceitos fundamentais como gerenciamento de estado e manipulação de eventos. A lógica do aplicativo é simples e fornece uma experiência interativa ao usuário, permitindo que ele veja rapidamente quantas vezes pressionou o botão. A estrutura modular do código facilita a manutenção e possíveis expansões futuras, tornando-se um bom exemplo para desenvolvedores iniciantes que desejam aprender a construir aplicativos móveis interativos.

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

const TapCounter = () => {
  const [count, setCount] = useState(0);

  return (
    <View style={styles.container}>
      <Text style={styles.countText}>Toques: {count}</Text>
      <Button title="Tocar" onPress={() => setCount(count + 1)} />
    </View>
  );
};

const styles = StyleSheet.create({
  container: { flex: 1, justifyContent: 'center', alignItems: 'center' },
  countText: { fontSize: 48, marginBottom: 20 },
});
```

```
});
```

```
export default TapCounter;
```



2. MapView

O código apresentado é uma aplicação simples em React Native que utiliza o componente MapView para exibir um mapa com um marcador na localização de São Paulo, Brasil. A aplicação é construída com a biblioteca react-native-maps, que permite integrar mapas interativos nas aplicações móveis.

A estrutura do código começa com a importação das bibliotecas necessárias, incluindo React, StyleSheet e View do react-native, além de MapView e Marker da biblioteca react-native-maps. O componente principal é chamado MapExample, que renderiza um mapa preenchendo toda a tela. Dentro do componente, um marcador (Marker) é adicionado à coordenada correspondente a São Paulo, com as propriedades de latitude e longitude definidas.

Os estilos são definidos usando StyleSheet, onde o estilo do mapa é configurado para ocupar 100% da largura e altura da tela. O componente é exportado como padrão, permitindo que seja utilizado em outras partes da aplicação.

Desta forma, este código exemplifica como criar uma aplicação básica de mapa em React Native, demonstrando conceitos