

Vitor Amadeu Souza

Introdução ao

React Native

No ambiente Expo Snack

Parte XXIV

© 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.



cerne-tec.com.br

Sumário

Capítulo I – Metodologia de desenvolvimento.....	6
1. Introdução.....	6
Capítulo II – Programação em React Native.....	7
1. Envio de SMS.....	7
2. Trocando a cor da tela.....	11
3. Trocando a cor da tela.....	14
4. Sintetizador de voz em outras línguas.....	17
5. Youtube.....	21

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. Envio de SMS

Este código cria um aplicativo em React Native que permite o envio de mensagens SMS para um número de telefone fornecido pelo usuário. O código utiliza a biblioteca expo-sms, que facilita a implementação do envio de SMS, mas funciona apenas se o dispositivo suportar essa funcionalidade.

No aplicativo, o usuário insere o número de telefone e a mensagem que deseja enviar. A validação do número de telefone é feita por meio de uma expressão regular (phoneRegex), que garante que o formato inserido seja correto, como "(XX) XXXXX-XXXX" ou "(XX) XXXX-XXXX". Caso o número de telefone esteja no formato correto, a função sendSMS verifica se o envio de SMS está disponível no dispositivo. Se estiver, a mensagem é enviada para o número indicado; caso contrário, uma mensagem de erro é exibida.

O layout do aplicativo consiste em um Text para o título, dois TextInput para o número de telefone e a mensagem, e um Button

para acionar o envio. A estilização é feita através do StyleSheet, com o objetivo de centralizar os elementos e aplicar bordas, cores e espaçamentos.

Esse código exemplifica o uso básico do expo-sms para criar uma funcionalidade de SMS em um aplicativo React Native.

```
import React, { useState } from 'react';
import { View, Text, TextInput, Button, StyleSheet, Alert }
from 'react-native';
import * as SMS from 'expo-sms';

const App = () => {
  const [phoneNumber, setPhoneNumber] = useState('');
  const [message, setMessage] = useState('');

  // Regex para validar o número de telefone
  // (ex: (XX) XXXXX-XXXX ou (XX) XXXX-XXXX)
  const phoneRegex = /^\\(\\d{2}\\) ?\\d{4,5}-\\d{4}$/;

  const sendSMS = async () => {
    if (phoneRegex.test(phoneNumber)) {
      const isAvailable = await SMS.isAvailableAsync();
      if (isAvailable) {
        await SMS.sendSMSAsync(
          [phoneNumber.replace(/\\D/g, '')],
          // Remove caracteres não numéricos
          message
        );
      } else {
        Alert.alert('Erro', 'O envio de SMS não está disponível
          neste dispositivo');
      }
    } else {
      Alert.alert('Número inválido', 'Por favor, insira um
        número de telefone válido no formato
        (XX) XXXXX-XXXX ou (XX) XXXX-XXXX');
    }
  };

  return (
    <View style={styles.container}>
```

```

    <Text style={styles.title}>Envio de SMS</Text>
    <TextInput
      style={styles.input}
      placeholder="Digite seu telefone (XX) XXXXX-XXXX"
      value={phoneNumber}
      onChangeText={setPhoneNumber}
      keyboardType="phone-pad"
    />
    <TextInput
      style={styles.input}
      placeholder="Digite sua mensagem"
      value={message}
      onChangeText={setMessage}
      multiline
    />
    <Button title="Enviar SMS" onPress={sendSMS} />
  </View>
);
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#f0f0f0',
    padding: 20,
  },
  title: {
    fontSize: 24,
    marginBottom: 20,
  },
  input: {
    width: '100%',
    padding: 10,
    borderColor: '#ccc',
    borderWidth: 1,
    borderRadius: 5,
    marginBottom: 20,
  },
});

export default App;

```

