

Vitor Amadeu Souza

Introdução ao

React Native

No ambiente Expo Snack

Parte IX

© 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 e-mail	7
2. Comunicação com API PokeAPI.....	13
3. Comunicação com API PokeAPI com Imagem...	18

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 e-mail

O código a ser estudado é um componente React Native chamado `SendEmail`, que permite ao usuário enviar um e-mail de forma simples e intuitiva. Utilizando o estado local gerenciado pelo hook `useState`, o componente coleta informações do destinatário, assunto e mensagem através de campos de entrada (`TextInput`).

A interface é enriquecida com ícones da biblioteca `react-native-vector-icons`, proporcionando uma experiência visual agradável. O botão "Enviar Email" aciona a função `sendEmail`, que valida se os campos obrigatórios estão preenchidos antes de construir uma URL de e-mail usando o protocolo `mailto`. Essa URL é então aberta utilizando a função `Linking.openURL`, que redireciona o usuário para seu cliente de e-mail padrão.

Além disso, o código inclui um tratamento de erros para garantir que, caso a abertura do cliente de e-mail

falhe, uma mensagem de alerta seja exibida. O estilo do componente é definido utilizando StyleSheet, garantindo que a interface seja responsiva e esteticamente agradável. Com essa implementação, o usuário tem uma maneira eficiente de compor e enviar e-mails diretamente de um aplicativo React Native.

```
import React, { useState } from 'react';
import { View, TextInput, Button, StyleSheet, Alert, Linking }
from 'react-native';
import Icon from 'react-native-vector-icons/FontAwesome';

const SendEmail = () => {
  const [email, setEmail] = useState('');
  const [subject, setSubject] = useState('');
  const [message, setMessage] = useState('');

  const sendEmail = () => {
    if (email === '' || message === '') {
      Alert.alert('Erro', 'Por favor, preencha todos os campos');
      return;
    }

    const mailUrl = `mailto:${email}?subject=${
      encodeURIComponent(subject)}&body=${encodeURIComponent(message)}`;

    Linking.openURL(mailUrl)
      .then(() => {
        setEmail('');
        setSubject('');
        setMessage('');
      })
      .catch(() => {
        Alert.alert('Erro', 'Não foi possível abrir o cliente de e-mail.');
```

```

return (
  <View style={styles.container}>
    <View style={styles.inputContainer}>
      <Icon name="envelope" size={24} color="gray" style={styles.icon}/>
      <TextInput
        style={styles.input}
        placeholder="Email do destinatário"
        value={email}
        onChangeText={setEmail}
        keyboardType="email-address"
      />
    </View>

    <TextInput
      style={styles.input}
      placeholder="Assunto"
      value={subject}
      onChangeText={setSubject}
    />
    <TextInput
      style={styles.input}
      placeholder="Mensagem"
      value={message}
      onChangeText={setMessage}
      multiline
    />
    <Button title="Enviar Email" onPress={sendEmail} />
  </View>
);
};

```

```

const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 20,
  },
  inputContainer: {
    flexDirection: 'row',
    alignItems: 'center',
    marginBottom: 15,
    borderWidth: 1,
    borderColor: '#ccc',
    paddingHorizontal: 10,
  },
});

```

```

    marginTop: 100,
  },
  icon: {
    marginRight: 10,
  },
  input: {
    flex: 1,
    height: 50,
    borderColor: '#ccc',
    paddingHorizontal: 10,
  }
});

```

```
export default SendEmail;
```

O componente SendEmail é uma aplicação React Native que permite ao usuário enviar e-mails diretamente de um aplicativo móvel. Ele utiliza o hook useState para gerenciar três estados: o endereço de e-mail do destinatário, o assunto da mensagem e o corpo do e-mail.

Ao ser renderizado, o componente exibe uma interface com campos de entrada (TextInput) para que o usuário insira as informações necessárias. O primeiro campo é acompanhado por um ícone de envelope, que adiciona um toque visual à interface. Os campos são projetados para coletar o e-mail do destinatário, o assunto e a mensagem, sendo que o campo de e-mail é