

Vitor Amadeu Souza

Introdução ao

React Native

No ambiente Expo Snack

Parte XX

© 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 PCB, 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. Cadastro com 3 telas.....	7
2. Navegador com tela inicial.....	14
3. Navegador com tela inicial.....	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. Cadastro com 3 telas

O código apresentado implementa um aplicativo em React Native que guia o usuário por um processo de cadastro. A aplicação utiliza um sistema de navegação em pilha, gerenciado pelo `createStackNavigator`, permitindo a transição entre diferentes telas que coletam informações do usuário, incluindo nome, idade, email, telefone e senha.

O arquivo principal, `App.js`, configura o `NavigationContainer` e define a navegação entre quatro telas: `PersonalInfoScreen`, `ContactInfoScreen`, `PasswordScreen` e `ConfirmationScreen`. Cada tela é responsável por coletar um conjunto específico de dados. Por exemplo, na `PersonalInfoScreen`, o usuário insere seu nome e idade, que são passados para a próxima tela ao clicar no botão "Próximo". Na `ContactInfoScreen`, o usuário fornece seu email e telefone, e esses dados são enviados à tela de criação de senha.

Na `PasswordScreen`, o usuário insere uma senha, que, junto com as informações anteriores, é encaminhada para a

ConfirmationScreen. Essa última tela exibe todos os dados inseridos e permite que o usuário envie uma confirmação por email utilizando a biblioteca react-native-email, ocultando a senha no corpo do email. O aplicativo tem uma estrutura clara e organizada, com cada componente gerenciando sua parte do processo de cadastro, resultando em uma experiência de usuário fluida e intuitiva. Além disso, o uso de estilos CSS com StyleSheet proporciona uma apresentação visual coesa em todas as telas.

```
//App.js
import React from 'react';
import { NavigationContainer } from '@react-navigation/native';
import { createStackNavigator } from '@react-navigation/stack';
import PersonalInfoScreen from './PersonalInfoScreen';
import ContactInfoScreen from './ContactInfoScreen';
import PasswordScreen from './PasswordScreen';
import ConfirmationScreen from './ConfirmationScreen';

const Stack = createStackNavigator();

export default function App() {
  return (
    <NavigationContainer>
      <Stack.Navigator initialRouteName="PersonalInfo">
        <Stack.Screen name="PersonalInfo"
          component={PersonalInfoScreen} options={{ title: 'Informações
Pessoais' }} />
        <Stack.Screen name="ContactInfo" component={ContactInfoScreen}
          options={{ title: 'Informações de Contato' }} />
        <Stack.Screen name="Password" component={PasswordScreen}
          options={{ title: 'Criar Senha' }} />
        <Stack.Screen name="Confirmation"
          component={ConfirmationScreen} options={{ title: 'Confirmação' }} />
      </Stack.Navigator>
    </NavigationContainer>
  );
}
```

```

//ConfirmationScreen.js
import React from 'react';
import { View, Text, Button, StyleSheet, Alert } from 'react-native';
import email from 'react-native-email';

const ConfirmationScreen = ({ route, navigation }) => {
  const { name, age, email: userEmail, phone, password } =
    route.params;

  const handleSendEmail = () => {
    const to = ['destino@exemplo.com']; // Substitua pelo email de
    destino

    email(to, {
      subject: 'Confirmação de Cadastro',
      body: `Nome: ${name}\nIdade: ${age}\nEmail: ${userEmail}\n
      Telefone: ${phone}\nSenha: ${password.replace(/./g, '*')}`
    }).catch((error) => Alert.alert('Erro', 'Ocorreu um erro ao enviar
    o e-mail.'));
  };

  return (
    <View style={styles.container}>
      <Text style={styles.label}>Nome: {name}</Text>
      <Text style={styles.label}>Idade: {age}</Text>
      <Text style={styles.label}>Email: {userEmail}</Text>
      <Text style={styles.label}>Telefone: {phone}</Text>
      <Text style={styles.label}>Senha: {password.replace(/./g,
      '*')}</Text>

      <Button title="Enviar Confirmação por Email"
      onPress={handleSendEmail} />
      <Button title="Concluir" onPress={() =>
      navigation.popToTop()} />
    </View>
  );
};

const styles = StyleSheet.create({
  container: { flex: 1, padding: 16, justifyContent: 'center' },
  label: { fontSize: 18, marginBottom: 12 },
});

export default ConfirmationScreen;

```

```

//ContactInfoScreen.js
import React, { useState } from 'react';
import { View, Text, TextInput, Button, StyleSheet } from 'react-native';

const ContactInfoScreen = ({ navigation, route }) => {
  const [email, setEmail] = useState('');
  const [phone, setPhone] = useState('');
  const { name, age } = route.params;

  return (
    <View style={styles.container}>
      <Text style={styles.label}>Email:</Text>
      <TextInput style={styles.input} value={email}
onChangeText={setEmail} placeholder="Digite seu email"
keyboardType="email-address" />

      <Text style={styles.label}>Telefone:</Text>
      <TextInput style={styles.input} value={phone}
onChangeText={setPhone} placeholder="Digite seu telefone"
keyboardType="phone-pad" />

      <Button title="Próximo" onPress={() =>
navigation.navigate('Password', { name, age, email, phone })} />
    </View>
  );
};

const styles = StyleSheet.create({
  container: { flex: 1, padding: 16 },
  label: { fontSize: 16, marginBottom: 8 },
  input: { borderWidth: 1, borderColor: 'ccc', padding: 8,
marginBottom: 16, borderRadius: 4 },
});

export default ContactInfoScreen;

//PasswordScreen.js
import React, { useState } from 'react';
import { View, Text, TextInput, Button, StyleSheet } from 'react-native';

const PasswordScreen = ({ navigation, route }) => {
  const [password, setPassword] = useState('');
  const { name, age, email, phone } = route.params;

  return (
    <View style={styles.container}>
      <Text style={styles.label}>Senha:</Text>

```