

Vitor Amadeu Souza

Introdução ao

React Native

No ambiente Expo Snack

Parte VI

© 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. Componente Progress.Bar.....	7
2. Componente Animated.View.....	10
3. Componente ImageBackground.....	15
4. Componente Platform.....	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. Componente Progress.Bar

O componente `Progress.Bar` em React Native é um elemento visual usado para indicar o progresso de uma tarefa em andamento. Ele pertence à biblioteca `react-native-progress`, que oferece uma série de componentes de barras de progresso. A barra pode ser configurada para exibir o avanço de uma atividade de forma contínua e visual, sendo usada para representar a porcentagem completada de uma tarefa. No código, o valor do progresso varia de 0 a 1, onde 0 significa que a tarefa ainda não começou e 1 representa que a tarefa foi concluída.

Por exemplo, o componente pode ser utilizado com uma propriedade `progress` para controlar o valor atual do progresso e ajustar a barra de forma dinâmica. A barra é bastante personalizável, permitindo que se configure a cor, a largura, o arredondamento dos cantos e outros detalhes visuais. O código pode incluir um botão para aumentar o valor da barra conforme o progresso avança, criando

uma interface interativa. Isso é especialmente útil em aplicativos que envolvem operações como downloads, carregamento de dados ou qualquer atividade que exija feedback visual sobre o estado de conclusão.

Assim, o `Progress.Bar` facilita a implementação de indicadores de progresso claros e responsivos em aplicativos móveis, proporcionando uma melhor experiência ao usuário ao fornecer informações visuais sobre o andamento de processos.

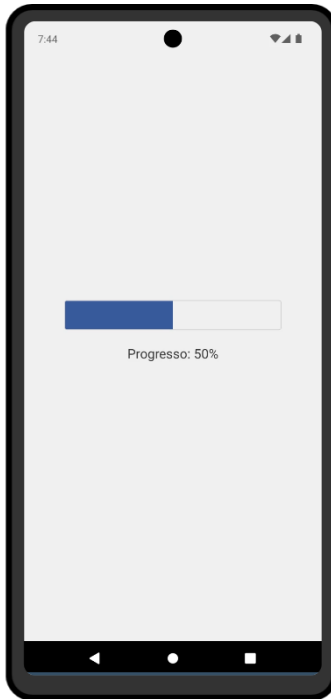
```
import React from 'react';
import { View, Text, StyleSheet } from 'react-native';
import * as Progress from 'react-native-progress';

const ProgressBarExample = () => (
  <View style={styles.container}>
    <Progress.Bar
      progress={0.5} // Progresso atual (0 a 1)
      width={300} // Largura da barra
      height={40} // Altura da barra
      color="#3b5998" // Cor da barra
      borderWidth={1} // Largura da borda
      borderColor="#ccc" // Cor da borda
    />
    <Text style={styles.text}>Progresso: 50%</Text>
  </View>
);

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#f0f0f0',
  },
  text: {
```

```
    marginTop: 20,  
    fontSize: 18,  
    color: '#333',  
  },  
});  
  
export default ProgressBarExample;
```

O código `ProgressBarExample` em React Native demonstra o uso do componente `Progress.Bar`, que exibe uma barra de progresso horizontal representando 50% de conclusão. O componente é personalizado para ter uma largura de 300 pixels, altura de 40 pixels, cor da barra definida como `#3b5998` (um tom de azul), e uma borda cinza clara com a cor `#ccc`. A barra de progresso tem sua porcentagem visualmente informada através de um texto abaixo dela, indicando "Progresso: 50%". Tudo está centralizado dentro de uma View estilizada com fundo cinza claro (`#f0f0f0`), usando flexbox para alinhar os elementos.



2. Componente Animated.View

O componente `Animated.View` no React Native é uma versão especial do componente `View`, projetada para permitir animações suaves de propriedades como opacidade, posição, escala e rotação. Esse componente faz parte da API `Animated`, que possibilita criar animações interativas e fluidas, interpolando valores ao longo do tempo. Isso significa que, em vez de alterar diretamente os valores