

Vitor Amadeu Souza

Introdução ao

React Native

No ambiente Expo Snack

Parte XXVI

© 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.

Novembro 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. App de exercícios.....	7
2. Lista de Compras Colaborativa.....	13
3. Agenda de contatos.....	19

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. App de exercícios

Este código ajuda os usuários a realizar e monitorar exercícios físicos com um temporizador, permitindo acompanhar o progresso semanal. O aplicativo possui uma lista de exercícios, incluindo flexões, abdominais, agachamentos e corrida no lugar, que são apresentados com imagens animadas. Quando o usuário seleciona um exercício, ele é levado a uma tela onde o exercício escolhido é exibido com um temporizador circular de 30 segundos, utilizando a biblioteca `react-native-countdown-circle-timer` para marcar a duração do exercício.

Ao completar um exercício, ele é adicionado ao histórico de progresso, e a porcentagem de progresso semanal é atualizada de acordo com a quantidade de exercícios concluídos. A interface inclui botões de toque que permitem aos usuários selecionar exercícios e um botão para marcar o exercício como concluído. A visualização do progresso semanal é apresentada como uma porcentagem calculada

com base no número de exercícios realizados em relação ao total disponível.

O aplicativo inclui funcionalidades como exibição de imagens, manipulação de estados para gerenciar o exercício selecionado e o progresso, e uma lista de histórico que mostra quais exercícios foram concluídos. O design do aplicativo é moderno e fácil de usar, com uma interface clara e visualmente atraente. Ele é ideal para quem deseja um guia interativo e motivacional para manter sua rotina de treinos em dia.

```
import React, { useState } from 'react';
import { View, Text, StyleSheet, TouchableOpacity, FlatList, Image, TextInput
} from 'react-native';
import { CountdownCircleTimer } from 'react-native-countdown-circle-
timer'; // Para o temporizador
import { Colors } from 'react-native-paper';

const App = () => {
  const [selectedExercise, setSelectedExercise] = useState(null);
  const [progress, setProgress] = useState([]);
  const [weekProgress, setWeekProgress] = useState(0); // Progress percentage

  const exercises = [
    { id: 1, name: 'Flexões', image:
'https://anabelavila.com/wp-content/uploads/2022/08/flexiones-
crossfit.gif' },
    { id: 2, name: 'Abdominais', image:
'https://media.giphy.com/media/TcUvUYWXHlAaL2g8sf/giphy.gif' },
    { id: 3, name: 'Agachamento', image:
'https://www.mundoboaforma.com.br/wp-content/uploads/2020/11/agachamento-com-
barra.gif' },
    { id: 4, name: 'Corrida no lugar', image:
'https://i.gifer.com/WDtP.gif' },
  ];
```

```

const handleSelectExercise = (exercise) => {
  setSelectedExercise(exercise);
};

const handleCompleteExercise = () => {
  setProgress([...progress, selectedExercise.name]);
  setWeekProgress(((progress.length + 1) / exercises.length) * 100);
  setSelectedExercise(null);
};

return (
  <View style={styles.container}>
    <Text style={styles.title}>App de Exercícios</Text>

    <Text style={styles.progressText}>Progresso Semanal:
{Math.round(weekProgress)}%</Text>

    {/* Lista de Exercícios */}
    <FlatList
      data={exercises}
      renderItem={({ item }) => (
        <TouchableOpacity
          style={styles.exerciseItem}
          onPress={() => handleSelectExercise(item)}
        >
          <Image source={{ uri: item.image }}
style={styles.exerciseImage} />
          <Text style={styles.exerciseText}>{item.name}</Text>
        </TouchableOpacity>
      )}
      keyExtractor={(item) => item.id.toString()}
    />

    {selectedExercise && (
      <View style={styles.selectedExerciseContainer}>
        <Text style={styles.selectedExerciseTitle}>
          {selectedExercise.name}
        </Text>
        <Image source={{ uri: selectedExercise.image }}
style={styles.selectedExerciseImage} />
        <CountdownCircleTimer
          isPlaying
          duration={30} // 30 segundos de exercício
          colors={[Colors.green500, Colors.yellow500, Colors.red500]}
          onComplete={handleCompleteExercise}
        >
          {({ remainingTime }) => <Text
style={styles.timerText}>{remainingTime}s</Text>}
        </CountdownCircleTimer>
      )}
    )}
  </View>
);

```

```

        <TouchableOpacity onPress={handleCompleteExercise}
style={styles.completeButton}>
        <Text style={styles.completeButtonText}>Completar
Exercício</Text>
        </TouchableOpacity>
    </View>
  )}

  { /* Histórico de Progresso */ }
  <Text style={styles.historyTitle}>Histórico de Exercícios</Text>
  <FlatList
    data={progress}
    renderItem={({ item }) => <Text
style={styles.historyText}>{item}</Text>}
    keyExtractor={(index) => index.toString()}
  />
</View>
);
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 20,
    backgroundColor: '#f8f8f8',
  },
  title: {
    fontSize: 30,
    fontWeight: 'bold',
    textAlign: 'center',
    marginBottom: 20,
    marginTop: 50,
  },
  progressText: {
    fontSize: 18,
    marginBottom: 10,
    textAlign: 'center',
  },
  exerciseItem: {
    flexDirection: 'row',
    alignItems: 'center',
    marginBottom: 15,
    padding: 10,
    backgroundColor: Colors.white,
    borderRadius: 10,
    shadowColor: '#000',
    shadowOpacity: 0.2,
    shadowOffset: { width: 0, height: 2 },
  },
});

```