

Nome: **Saulo Henrique dos Santos**

Matrícula: **202208545378**

Curso: **Desenvolvimento Full Stack**

Campus: **POLO Indaiatuba/SP**

Github: **<https://github.com/SauloHenriqueSantos/M4N1>**



Missão Prática | Mundo 4 | Nível 1
RPG0023 - Vamos criar um App

Objetivo:

Cadastro de Fornecedores: O aplicativo deve permitir o cadastro de fornecedores, incluindo informações detalhadas, como nome, endereço, contato e categorias de produtos fornecidos. Essas informações serão exibidas utilizando componentes

como `<Text>`, `<TextInput>`, e `<Image>`.

Listagem de Fornecedores: Deve ser possível visualizar uma lista de fornecedores cadastrados, com opções de pesquisa e filtragem com base em critérios como categoria ou localização. A lista de fornecedores será exibida utilizando componentes como `<Text>` e `<Image>`.

Associação de Imagens: O aplicativo deve permitir a associação de imagens aos perfis dos fornecedores. Os usuários devem poder fazer o upload de logotipos ou fotos relacionadas ao fornecedor, utilizando o componente `<Image>`.

Experiência de Usuário Intuitiva: A interface do aplicativo deve ser intuitiva e fácil de usar, garantindo que os usuários possam navegar, adicionar e editar informações de forma eficiente. Isso será alcançado usando componentes como `<Text>`, `<TextInput>`, e `<Image>`.

Resultado Final

```
JS App.js X
src > JS App.js > ...
1 import React, { useState } from 'react';
2 import { Text, View, TextInput, ScrollView, TouchableOpacity, Image, StyleSheet } from 'react-native';
3
4
5 const App = () => {
6   // Estado para armazenar a lista de fornecedores
7   const [fornecedores, setFornecedores] = useState([]);
8   // Estado para armazenar os detalhes do fornecedor sendo adicionado
9   const [novoFornecedor, setNovoFornecedor] = useState({
10     nome: '',
11     endereco: '',
12     contato: '',
13     categorias: '',
14     imagem: ''
15   });
16
17   // Função para lidar com a adição de um novo fornecedor
18   const adicionarFornecedor = () => {
19     // Adiciona o novo fornecedor à lista de fornecedores
20     setFornecedores([...fornecedores, novoFornecedor]);
21     // Limpa os detalhes do fornecedor após a adição
22     setNovoFornecedor({
23       nome: '',
24       endereco: '',
25       contato: '',
26       categorias: '',
27       imagem: ''
28     });
29   };
30
31   return (
32     <ScrollView contentContainerStyle={styles.container}>
33       <Text style={styles.title}>Cadastro de Fornecedores</Text>
34
35       <View style={styles.form}>
36         <Text style={styles.input}
37           placeholder="Nome"
38           value={novoFornecedor.nome}
39           onChangeText={(text) => setNovoFornecedor({ ...novoFornecedor, nome: text })}
40         />
41         <Text style={styles.input}
42           placeholder="Endereco"
43           value={novoFornecedor.endereco}
44           onChangeText={(text) => setNovoFornecedor({ ...novoFornecedor, endereco: text })}
45         />
46       </View>
47     </ScrollView>
48   );
49 }
```

```
const App = () => {
  // Estado para armazenar a lista de fornecedores
  const [fornecedores, setFornecedores] = useState([]);
  // Estado para armazenar os detalhes do fornecedor sendo adicionado
  const [novoFornecedor, setNovoFornecedor] = useState({
    nome: '',
    endereco: '',
    contato: '',
    categorias: '',
    imagem: ''
  });

  // Função para lidar com a adição de um novo fornecedor
  const adicionarFornecedor = () => {
    // Adiciona o novo fornecedor à lista de fornecedores
    setFornecedores([...fornecedores, novoFornecedor]);
    // Limpa os detalhes do fornecedor após a adição
    setNovoFornecedor({
      nome: '',
      endereco: '',
      contato: '',
      categorias: '',
      imagem: ''
    });
  };

  return (
    <ScrollView contentContainerStyle={styles.container}>
      <Text style={styles.title}>Cadastro de Fornecedores</Text>

      <View style={styles.form}>
        <Text style={styles.input}
          placeholder="Nome"
          value={novoFornecedor.nome}
          onChangeText={(text) => setNovoFornecedor({ ...novoFornecedor, nome: text })}
        />
        <Text style={styles.input}
          placeholder="Endereco"
          value={novoFornecedor.endereco}
          onChangeText={(text) => setNovoFornecedor({ ...novoFornecedor, endereco: text })}
        />
        <Text style={styles.input}
          placeholder="Contato"
          value={novoFornecedor.contato}
          onChangeText={(text) => setNovoFornecedor({ ...novoFornecedor, contato: text })}
        />
        <Text style={styles.input}
          placeholder="Categorias"
          value={novoFornecedor.categorias}
          onChangeText={(text) => setNovoFornecedor({ ...novoFornecedor, categorias: text })}
        />
        <Text style={styles.input}
          placeholder="URL da Imagem"
          value={novoFornecedor.imagem}
          onChangeText={(text) => setNovoFornecedor({ ...novoFornecedor, imagem: text })}
        />
        <TouchableOpacity style={styles.button} onPress={adicionarFornecedor}>
          <Text style={styles.buttonText}>Adicionar Fornecedor</Text>
        </TouchableOpacity>
      </View>

      <View style={styles.lista}>
        <Text>Lista de fornecedores cadastrados</Text>
        <View key={index} style={styles.fornecedor}>
          <Image source={{ uri: fornecedor.imagem }} style={styles.imagem} />
          <Text style={styles.nome}>{fornecedor.nome}</Text>
          <Text>{fornecedor.endereco}</Text>
          <Text>{fornecedor.contato}</Text>
          <Text>{fornecedor.categorias}</Text>
        </View>
      </View>
    </ScrollView>
  );
};

const styles = StyleSheet.create({
  container: {
    flexGrow: 1,
    padding: 20
  },
  title: {
    text-align: center,
    marginBottom: 10
  },
  form: {
    margin: 10
  },
  input: {
    margin: 5,
    padding: 5
  },
  button: {
    margin: 10,
    padding: 10
  },
  buttonText: {
    text-align: center
  },
  lista: {
    margin: 10
  },
  fornecedor: {
    margin: 5,
    padding: 5
  },
  imagem: {
    width: 50px,
    height: 50px
  },
  nome: {
    margin: 5
  }
});
```