



**Missão Prática – Mundo 04 – Nível 01**

**Gilvan Pereira de Oliveira – 2023.01.53256-6**

**1197 – POLO CENTRO – SÃO LOURENÇO DA MATA - PE**

**RPG0023 – Vamos criar um App – 9001 – 2024.2**

**Repositório GitHub: [GilvanPOliveira/Meeting \(github.com\)](https://github.com/GilvanPOliveira/Meeting)**

## **Contextualização**

A empresa "Meeting" busca criar um aplicativo móvel eficaz para o cadastro de fornecedores, com listas e imagens de alta qualidade, economizando recursos e proporcionando uma excelente experiência ao usuário. A escolha da tecnologia React Native é crucial para estabelecer uma presença sólida no mercado móvel. Nesta atividade, você aprenderá os princípios básicos do React Native.

## **Requisitos Funcionais**

1. **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>`.
2. **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>`.
3. **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>`.
4. **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>`.

## **Desenvolvendo seu aplicativo**

Após a conclusão das atividades e com base nos códigos e conhecimentos adquiridos, é hora de desenvolver o seu aplicativo, levando em consideração os requisitos apresentados na contextualização. Utilize as habilidades e os componentes do React Native que você aprendeu para criar um aplicativo eficaz que atenda às necessidades da empresa "Meeting". Certifique-se de incorporar os princípios do desenvolvimento React Native, como a criação de componentes, o uso de props e a exibição de imagens, para alcançar os objetivos definidos. Este é o momento de aplicar seus conhecimentos e criar uma presença sólida no mercado móvel para a empresa.

## **Resultados esperados**

Após seguir esses passos, você teve uma compreensão inicial dos princípios do React Native, incluindo como criar componentes, utilizar props e adicionar imagens. Estas são as bases para a construção de aplicativos móveis eficazes e de alta qualidade. Com base nos códigos acima você verá os seguintes componentes que serão base para a construção do seu aplicativo.

## **RPG0023 - Vamos criar um App**

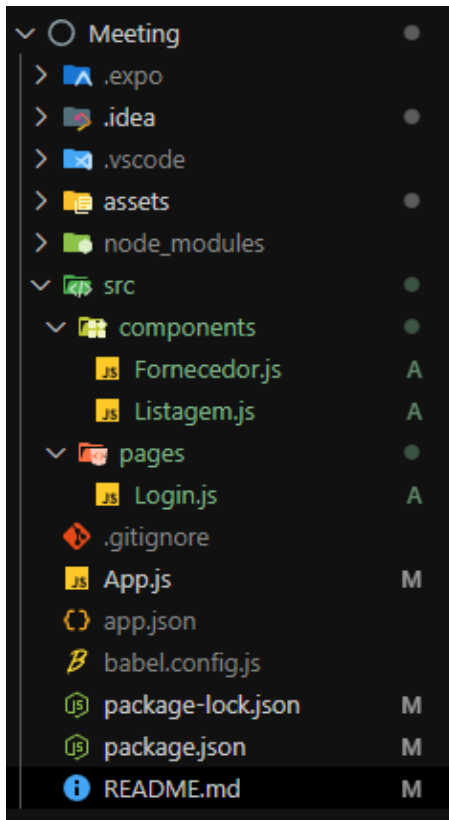
O projeto foi desenvolvido utilizando o Android Studio e o Expo, escolhidos pela praticidade que oferecem no desenvolvimento com React Native. Optei por utilizar o editor de código Visual Studio Code (VS Code), com o qual tenho maior familiaridade, para realizar todo o processo de criação e implementação do aplicativo solicitado na missão prática do nível 01 do mundo 04. O Expo facilitou a configuração do ambiente e acelerou o fluxo de trabalho, permitindo maior foco na escrita do código e testes contínuos do aplicativo, sem precisar de fato utilizar o editor de código do Android Studio.

Após as configurações feitas, o projeto foi criado com o seguinte código:

- expo init Meeting;
- choose a template: blank – a minimal app as clean as an empty canvas;
- para criar o projeto instalando as dependências em JavaScript, linguagem utilizada, com o React Native;
- após a criação, foi selecionado a pasta do projeto: cd Meeting;
- e aberto o editor de código com o comando: code .
- se toda configuração foi feita corretamente, basta executar o comando: npm start, e executar o projeto.

Obs: Informações adicionais:

- Instalar: `npm install expo-image-picker`
  - Para poder fazer a seleção das imagens no celular;
- Paleta de cores utilizada:
  - #000
  - #16423C
  - #6A9C89
  - #C4DAD2
  - #E9EFEC
  - #FF6F6F
  - #fff

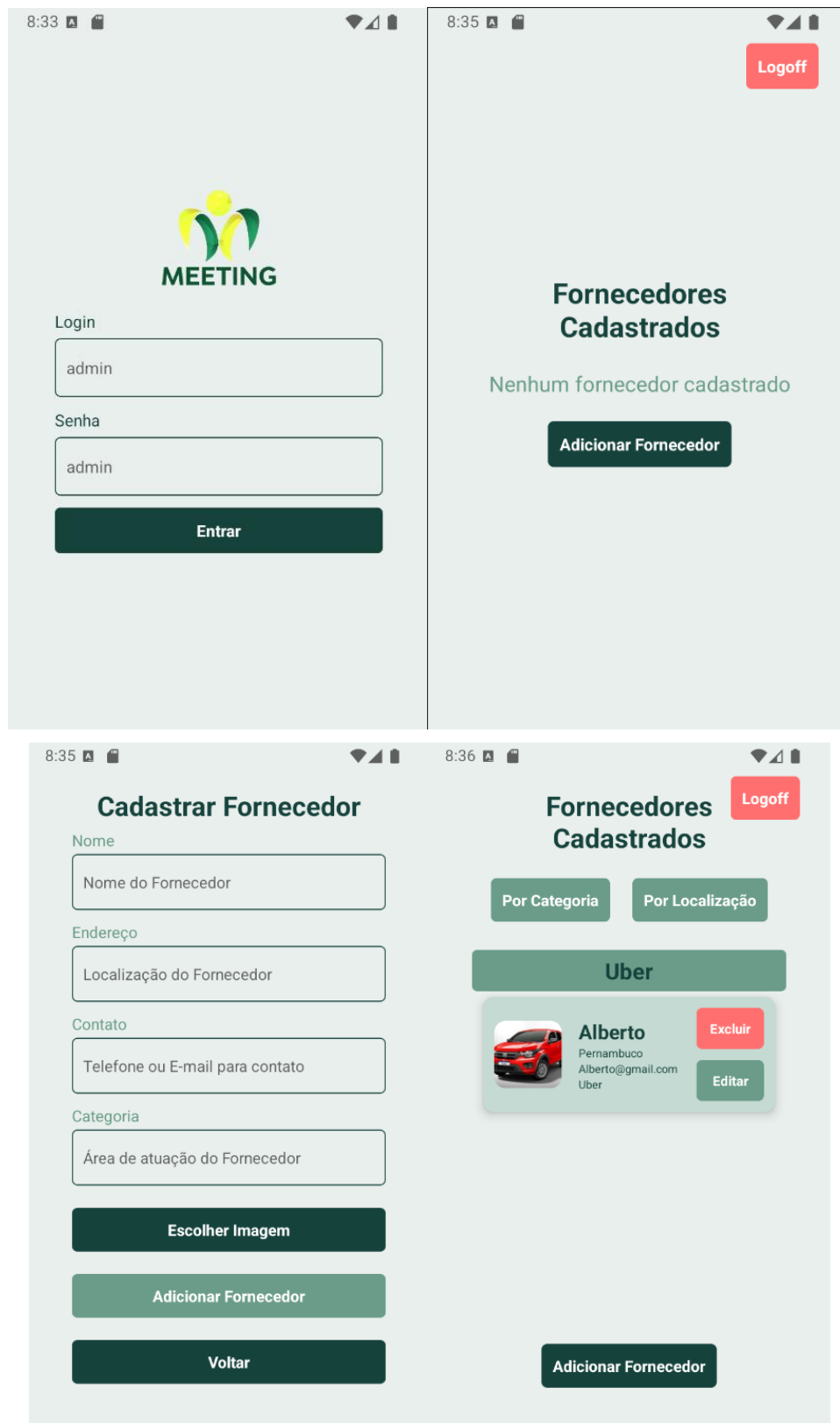


Foi idealizado a seguinte estrutura demonstrada ao lado:

```
Meeting
├── src
│   ├── components
│   │   ├── Fornecedor.js
│   │   └── Listagem.js
│   └── pages
│       └── Login.js
├── App.js
└── README.md
```

Onde o aplicativo se inicia com uma tela de Login, seguida por uma segunda tela exibindo a listagem dos Fornecedores, uma filtragem por Categoria ou Localização nos cadastros realizados, Cadastro de Fornecedores, exclusão ou edição dos cadastros feitos e logoff do aplicativo.

Seguem as telas do aplicativo em execução:




Obs: Foi adicionado uma opção para edição da imagem escolhida, mas também está comentado no código para o caso de necessidade de se retirar.

Abaixo a tela de edição da imagem escolhida e da edição do fornecedor cadastrado

8:38

← ↺ [ ] CROP



8:39

Editar Fornecedor

Nome

Alberto

Endereço

Pernambuco


Contato

Alberto@gmail.com

Categoria

Uber

Escolher Imagem



Atualizar Fornecedor

Filtrados por Categoria ou Localização:

8:49

Fornecedores Cadastrados

Logoff

Por Categoria Por Localização

Uber



**Alberto**  
Pernambuco  
Alberto@gmail.com  
Uber

Excluir Editar



**Carlos**  
Bahia  
123456789  
Uber

Excluir Editar

Adicionar Fornecedor

8:48

Fornecedores Cadastrados

Logoff

Por Categoria Por Localização

Bahia



**Carlos**  
Bahia  
123456789  
Uber

Excluir Editar

Pernambuco



**Alberto**  
Pernambuco  
Alberto@gmail.com  
Uber

Excluir Editar

Adicionar Fornecedor

Abaixo, seguem os códigos referente as páginas:

### **App.js:**

```
import React, { useState } from "react";
import { View, StyleSheet } from "react-native";
import Fornecedor from "../src/components/Fornecedor";
import Listagem from "../src/components/Listagem";
import Login from "../src/pages/Login";

export default function App() {
  const [paginaAtual, setPaginaAtual] = useState("login");
  const [fornecedores, setFornecedores] = useState([]);
  const [fornecedorEditando, setFornecedorEditando] = useState(null);

  const adicionarFornecedor = (novoFornecedor) => {
    if (fornecedorEditando) {
      setFornecedores(fornecedores.map(f => f.id === novoFornecedor.id ?
novoFornecedor : f));
      setFornecedorEditando(null);
    } else {
      setFornecedores([...fornecedores, novoFornecedor]);
    }
    setPaginaAtual("listagem");
  };

  const excluirFornecedor = (id) => {
    setFornecedores(fornecedores.filter((fornecedor) => fornecedor.id !== id));
  };

  const editarFornecedor = (fornecedor) => {
    setFornecedorEditando(fornecedor);
    setPaginaAtual("fornecedor");
  };
};
```

```

const handleLogin = () => {
  setPaginaAtual("listagem");
};

const handleLogoff = () => {
  setPaginaAtual("login");
};

return (
  <View style={styles.container}>
    {paginaAtual === "login" ? (
      <Login onLogin={handleLogin} />
    ) : paginaAtual === "listagem" ? (
      <Listagem
        fornecedores={fornecedores}
        onAdicionar={() => {
          setFornecedorEditando(null);
          setPaginaAtual("fornecedor");
        }}
        onExcluir={excluirFornecedor}
        onEditar={editarFornecedor}
        onLogoff={handleLogoff}
      />
    ) : (
      <Fornecedor
        onAdicionar={adicionarFornecedor}
        onVoltar={() => setPaginaAtual("listagem")}
        fornecedor={fornecedorEditando}
      />
    )}
  </View>
);
}

```

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 20,
    justifyContent: "center",
    backgroundColor: "#E9EFEC",
  },
});
```

### **Login.js:**

```
import React, { useState } from "react";
import {
  View,
  Text,
  TextInput,
  TouchableOpacity,
  Image,
  StyleSheet,
  Alert,
} from "react-native";
```

```
export default function Login({ onLogin }) {
  const [login, setLogin] = useState("");
  const [senha, setSenha] = useState("");
```

*//Caso precise, basta alterar o login e senha para o que desejar*

```
const handleLogin = () => {
  if (login === "admin" && senha === "admin") {
    onLogin();
  } else {
    Alert.alert(
      "Credenciais inválidas",
      "O login ou a senha estão incorretos."
```



```

    );
  }
};

return (
  <View style={styles.container}>
    <Image source={require("../assets/meeting.png")} style={styles.logo} />
    <Text style={styles.texto}>Login</Text>
    <TextInput
      placeholder="admin"
      value={login}
      onChangeText={setLogin}
      style={styles.input}
    />
    <Text style={styles.texto}>Senha</Text>
    <TextInput
      placeholder="admin"
      value={senha}
      onChangeText={setSenha}
      secureTextEntry
      style={styles.input}
    />
    <TouchableOpacity onPress={handleLogin} style={styles.btn}>
      <Text style={styles.btnTexto}>Entrar</Text>
    </TouchableOpacity>
  </View>
);
}

```

```

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",

```

```
    backgroundColor: "#E9EFEC",
    padding: 20,
  },
  texto: {
    fontSize: 14,
    color: "#16423C",
    marginBottom: 5,
    textAlign: "left",
    width: "100%",
  },
  logo: {
    width: 100,
    height: 85,
    marginBottom: 20,
  },
  input: {
    borderWidth: 1,
    borderColor: "#16423C",
    padding: 10,
    marginBottom: 10,
    width: "100%",
    borderRadius: 5,
  },
  btn: {
    backgroundColor: "#16423C",
    padding: 10,
    width: "100%",
    borderRadius: 5,
    alignItems: "center",
  },
  btnTexto: {
    color: "#fff",
    fontWeight: "bold",
  },
},
```

```
});
```

### Listagem:

```
import React, { useState } from "react";
```

```
import {
```

```
  View,
```

```
  Text,
```

```
  SectionList,
```

```
  TouchableOpacity,
```

```
  Image,
```

```
  StyleSheet,
```

```
} from "react-native";
```

```
export default function Listagem({
```

```
  fornecedores,
```

```
  onAdicionar,
```

```
  onExcluir,
```

```
  onEditar,
```

```
  onLogoff,
```

```
}) {
```

```
  const [criterio, setCriterio] = useState("categoria");
```

```
  const fornecedoresAgrupados = fornecedores.reduce((secoes, fornecedor) =>  
  {
```

```
    const chave =
```

```
      criterio === "categoria" ? fornecedor.categoria : fornecedor.endereco;
```

```
    if (!secoes[chave]) {
```

```
      secoes[chave] = [];
```

```
    }
```

```
    secoes[chave].push(fornecedor);
```

```
    return secoes;
```

```
  }, {});
```

```
  const secoesFormatadas = Object.keys(fornecedoresAgrupados)
```

```

.sort()
.map((chave) => ({
  title: chave,
  data: fornecedoresAgrupados[chave].sort((a, b) =>
    a.nome.localeCompare(b.nome)
  ),
}));

```

```

const renderItem = ({ item }) => (
  <View style={styles.cardContainer}>
    <View style={styles.card}>
      <Image source={{ uri: item.imagem }} style={styles.imagem} />
      <View style={styles.infoContainer}>
        <Text style={styles.itensTitulo}>{item.nome}</Text>
        <Text style={styles.itens}>{item.endereco}</Text>
        <Text style={styles.itens}>{item.contato}</Text>
        <Text style={styles.itens}>{item.categoria}</Text>
      </View>

      <View style={styles.btnContainer}>
        <TouchableOpacity
          onPress={() => onExcluir(item.id)}
          style={styles.btnExcluir}
        >
          <Text style={styles.excTexto}>Excluir</Text>
        </TouchableOpacity>

        <TouchableOpacity
          onPress={() => onEditar(item)}
          style={styles.btnEditar}
        >
          <Text style={styles.edtTexto}>Editar</Text>
        </TouchableOpacity>
      </View>

```

```

    </View>
  </View>
);

const renderSectionHeader = ({ section: { title } }) => (
  <Text style={styles.secaoTitulo}>{title}</Text>
);

return (
  <View style={styles.container}>
    <Text style={styles.titulo}>Fornecedores Cadastrados</Text>

    {fornecedores.length > 0 && (
      <View style={styles.filtroContainer}>
        <TouchableOpacity
          style={[
            styles.btnFiltro,
            criterio === "categoria" && styles.btnFiltroAtivo,
          ]}
          onPress={() => setCritério("categoria")}
        >
          <Text style={styles.btnFiltroTexto}>Por Categoria</Text>
        </TouchableOpacity>
        <TouchableOpacity
          style={[
            styles.btnFiltro,
            criterio === "endereço" && styles.btnFiltroAtivo,
          ]}
          onPress={() => setCritério("endereço")}
        >
          <Text style={styles.btnFiltroTexto}>Por Localização</Text>
        </TouchableOpacity>
      </View>
    )}
  </View>
);

```

```

    {secoesFormatadas.length > 0 ? (
      <SectionList
        sections={secoesFormatadas}
        keyExtractor={({item}) => item.id}
        renderItem={renderItem}
        renderSectionHeader={renderSectionHeader}
      />
    ) : (
      <Text style={styles.texto}>Nenhum fornecedor cadastrado</Text>
    )}

    <TouchableOpacity style={styles.btn} onPress={onAdicionar}>
      <Text style={styles.btnTexto}>Adicionar Fornecedor</Text>
    </TouchableOpacity>

    <TouchableOpacity style={styles.btnLogoff} onPress={onLogoff}>
      <Text style={styles.logoffTexto}>Logoff</Text>
    </TouchableOpacity>
  </View>
);
}

```

```

const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 20,
    backgroundColor: "#E9EFEC",
    alignItems: "center",
    justifyContent: "center",
  },
  titulo: {
    fontSize: 24,
    fontWeight: "bold",

```

```
marginBottom: 20,
textAlign: "center",
color: "#16423C",
},
filtroContainer: {
  flexDirection: "row",
  justifyContent: "center",
  marginBottom: 20,
},
btnFiltro: {
  backgroundColor: "#6A9C89",
  padding: 10,
  borderRadius: 5,
  marginHorizontal: 10,
},
btnFiltroAtivo: {
  backgroundColor: "#16423C",
},
btnFiltroTexto: {
  color: "fff",
  fontWeight: "bold",
},
texto: {
  fontSize: 18,
  textAlign: "center",
  color: "#6A9C89",
},
itens: {
  color: "#16423C",
  fontSize: 10,
},
cardContainer: {
  flexDirection: "row",
  alignItems: "center",
```

```
marginBottom: 10,
paddingLeft: 10,
paddingRight: 10,
width: 280,
},
card: {
  flex: 1,
  flexDirection: "row",
  backgroundColor: "#C4DAD2",
  padding: 10,
  borderRadius: 10,
  alignItems: "center",
  shadowColor: "#000",
  shadowOffset: { width: 0, height: 1 },
  shadowOpacity: 0.8,
  shadowRadius: 2,
  elevation: 5,
},
imagem: {
  width: 60,
  height: 60,
  borderRadius: 10,
  marginRight: 15,
},
infoContainer: {
  flex: 1,
},
itemsTitulo: {
  fontSize: 18,
  fontWeight: "bold",
  color: "#16423C",
},
btnContainer: {
  flexDirection: "column",
```



```
justifyContent: "center",
alignItems: "center",
marginLeft: 10,
},
btnExcluir: {
  backgroundColor: "#FF6F6F",
  padding: 10,
  borderRadius: 5,
  marginBottom: 10,
  width: 60,
},
excTexto: {
  fontSize: 12,
  color: "#fff",
  fontWeight: "bold",
  textAlign: "center",
},
btnEditar: {
  backgroundColor: "#6A9C89",
  padding: 10,
  width: 60,
  borderRadius: 5,
},
edtTexto: {
  fontSize: 12,
  color: "#fff",
  fontWeight: "bold",
  textAlign: "center",
},
secaoTitulo: {
  fontSize: 20,
  fontWeight: "bold",
  marginVertical: 5,
  textAlign: "center",
```

```

    backgroundColor: "#6A9C89",
    paddingVertical: 5,
    borderRadius: 5,
    color: "#16423C",
  },
  btn: {
    backgroundColor: "#16423C",
    padding: 10,
    alignItems: "center",
    marginTop: 20,
    borderRadius: 5,
  },
  btnTexto: {
    color: "#fff",
    fontWeight: "bold",
  },
  btnLogoff: {
    position: "absolute",
    top: 10,
    right: 8,
    backgroundColor: "#FF6F6F",
    padding: 10,
    borderRadius: 5,
  },
  logoffTexto: {
    color: "#fff",
    fontWeight: "bold",
  },
});

```

### **Fornecedor.js:**

```

import React, { useState, useEffect } from "react";
import {

```

```
View,  
Text,  
TextInput,  
btn,  
TouchableOpacity,  
Image,  
Alert,  
StyleSheet,  
ScrollView,  
} from "react-native";  
import * as ImagePicker from "expo-image-picker";
```

```
export default function Fornecedor({ onAdicionar, onVoltar, fornecedor }) {  
  const [nome, setNome] = useState("");  
  const [endereco, setEndereco] = useState("");  
  const [contato, setContato] = useState("");  
  const [categoria, setCategoria] = useState("");  
  const [imagem, setImagem] = useState(null);
```

```
  useEffect(() => {  
    if (fornecedor) {  
      setNome(fornecedor.nome);  
      setEndereco(fornecedor.endereco);  
      setContato(fornecedor.contato);  
      setCategoria(fornecedor.categoria);  
      setImagem(fornecedor.imagem);  
    }  
  }, [fornecedor]);
```

*//Opção para escolher imagem da galeria*

```
const escolherImagem = async () => {  
  let result = await ImagePicker.launchImageLibraryAsync({  
    mediaTypes: ImagePicker.MediaTypeOptions.All,
```

*//as opções abaixo são para editar a imagem, caso queira desabilitar, basta comentar a linha abaixo*

```

        allowsEditing: true,
        aspect: [4, 3],
        quality: 1,
    });

    if (!result.canceled && result.assets && result.assets.length > 0) {
        setImagem(result.assets[0].uri);
    }
};

const adicionarFornecedor = () => {
    if (nome && endereco && contato && categoria && imagem) {
        const novoFornecedor = {
            id: fornecedor ? fornecedor.id : Math.random().toString(),
            nome,
            endereco,
            contato,
            categoria,
            imagem,
        };
        onAdicionar(novoFornecedor);
        setNome("");
        setEndereco("");
        setContato("");
        setCategoria("");
        setImagem(null);
    } else {
        Alert.alert("Preencha todos os campos e selecione uma imagem.");
    }
};

return (
    <ScrollView>
        <View style={styles.container}>

```

```
<Text style={styles.tituloTexto}>
  {fornecedor ? "Editar Fornecedor" : "Cadastrar Fornecedor"}
</Text>
```

```
<Text style={styles.texto}>Nome</Text>
```

```
<TextInput
  placeholder="Nome do Fornecedor"
  value={nome}
  onChangeText={setNome}
  style={styles.input}
  />
```

```
<Text style={styles.texto}>Endereço</Text>
```

```
<TextInput
  placeholder="Localização do Fornecedor"
  value={endereço}
  onChangeText={setEndereço}
  style={styles.input}
  />
```

```
<Text style={styles.texto}>Contato</Text>
```

```
<TextInput
  placeholder="Telefone ou E-mail para contato"
  value={contato}
  onChangeText={setContato}
  style={styles.input}
  />
```

```
<Text style={styles.texto}>Categoria</Text>
```

```
<TextInput
  placeholder="Área de atuação do Fornecedor"
  value={categoria}
  onChangeText={setCategoria}
  style={styles.input}
  />
```

```
<TouchableOpacity onPress={escolherImagem} style={styles.btn}>
```

```
    <Text style={styles.btnTexto}>Escolher Imagem</Text>
  </TouchableOpacity>
```

```
{imagem && <Image source={{ uri: imagem }} style={styles.imagem} />}
```

```
  <TouchableOpacity
    onPress={adicionarFornecedor}
    style={styles.btnAdicionar}
  >
    <Text style={styles.btnTexto}>
      {fornecedor ? "Atualizar Fornecedor" : "Adicionar Fornecedor"}
    </Text>
  </TouchableOpacity>
```

```
  <TouchableOpacity onPress={onVoltar} style={styles.btn}>
    <Text style={styles.btnTexto}>Voltar</Text>
  </TouchableOpacity>
```

```
  </View>
```

```
</ScrollView>
```

```
);
```

```
}
```

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    padding: 20,
    backgroundColor: "#E9EFEC",
  },
  tituloTexto: {
    fontSize: 24,
    fontWeight: "bold",
    marginBottom: 5,
    textAlign: "center",
```

```
    color: "#16423C",
  },
  texto: {
    fontSize: 14,
    color: "#6A9C89",
    marginBottom: 3,
  },
  input: {
    borderWidth: 1,
    borderColor: "#16423C",
    padding: 10,
    marginBottom: 10,
    borderRadius: 5,
  },
  btn: {
    backgroundColor: "#16423C",
    padding: 10,
    alignItems: "center",
    marginVertical: 10,
    borderRadius: 5,
  },
  btnTexto: {
    color: "fff",
    fontWeight: "bold",
  },
  btnAdicionar: {
    backgroundColor: "#6A9C89",
    padding: 10,
    alignItems: "center",
    marginVertical: 10,
    borderRadius: 5,
  },
  imagem: {
    width: 100,
```

```
height: 100,  
borderRadius: 10,  
marginVertical: 10,  
alignSelf: "center",  
},  
));
```