



Jonathan Domingos Carneiro da Silva
matricula: 2023.04.12744-1
3274 POLO CENTRO - ITAITINGA – CE

Nível 1: RPG0023 - Vamos criar um App

– DESENVOLVIMENTO FULL STAC – 2024.3

Objetivos da prática

- Configurar o ambiente de desenvolvimento React Native;
- Implementar a funcionalidade de entrada de texto em um componente React Native;
- Implementar um Componente de Lista Dinâmica (ScrollView);
- em listas;
- Empregar elementos visuais em um aplicativo React Native.

Introdução

Desenvolvi um aplicativo mobile utilizando a tecnologia React Native para auxiliar no gerenciamento de fornecedores. A aplicação possibilita o cadastro completo de novos fornecedores, incluindo um campo para inserção de imagens. Os dados cadastrados abrangem informações como nome da empresa, endereço completo, contatos para comunicação e categorias de produtos fornecidos.

App.js

```
import React, { useState } from "react";
import { StyleSheet, View, ScrollView, Text, TouchableOpacity } from "react-native";
import { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from '@react-navigation/native-stack';
import CadastroFornecedor from "../components/cadastro";
import ListaFornecedores from "../components/fornecedores";

const Stack = createNativeStackNavigator();

export default function App() {
  const [fornecedores, setFornecedores] = useState([]);

  const handleCastroSubmit = (novoFornecedor) => {
    setFornecedores([...fornecedores, novoFornecedor]);
  };

  const handleRemove = (fornecedorId) => {
    setFornecedores(currentFornecedores => currentFornecedores.filter(f =>
f.id !== fornecedorId));
  };
}
```



```
const handleEdit = (fornecedorId, updatedData) => {
  setFornecedores(currentFornecedores => currentFornecedores.map(f => {
    if (f.id === fornecedorId) {
      return { ...f, ...updatedData };
    }
    return f;
  }));
};

return (
  <NavigationContainer>
    <Stack.Navigator initialRouteName="Cadastro" screenOptions={{
      headerStyle: {
        backgroundColor: '#f7f7f7',
      },
      headerTintColor: '#333',
      headerTitleStyle: {
        fontWeight: 'bold',
      }
    }}>
      <Stack.Screen name="Cadastro" options={{ title: 'Cadastrar Fornecedores' }}>
        {props => (
          <ScrollView contentContainerStyle={styles.scroll}>
            <View style={styles.section}>
              <CadastroFornecedor
                onCadastroSubmit={handleCastroSubmit} {...props} />
              <TouchableOpacity style={styles.button}
                onPress={() => props.navigation.navigate('Fornecedores')}>
                <Text style={styles.buttonText}>Lista de fornecedores</Text>
              </TouchableOpacity>
            </View>
          </ScrollView>
        )}
      </Stack.Screen>
      <Stack.Screen name="Fornecedores" options={{ title: 'Lista de Fornecedores' }}>
        {() => <ListaFornecedores fornecedores={fornecedores}>
          </Stack.Screen>
        }
      </Stack.Navigator>
    </NavigationContainer>
  );
}
```



```
const styles = StyleSheet.create({
  scroll: {
    flexGrow: 1,
    justifyContent: "center",
  },
  section: {
    flex: 1,
    alignItems: "center",
    justifyContent: "center",
    padding: 20,
  },
  button: {
    marginTop: 20,
    backgroundColor: "#eb984e",
    borderRadius: 16,
    width: '80%',
    paddingVertical: 15,
    paddingHorizontal: 20,
    alignItems: 'center',
    justifyContent: 'center',
    elevation: 3,
    shadowColor: '#000',
    shadowOffset: { width: 0, height: 2 },
    shadowOpacity: 0.1,
    shadowRadius: 1.5,
  },
  buttonText: {
    color: '#000',
    fontSize: 18,
    fontWeight: 'bold'
  }
});
```

forneceedores.js

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

const useFiltrarFornecedores = (fornecedores, categoriaFiltro) => {
```



```
const fornecedoresFiltrados = useMemo(
  () =>
    fornecedores.filter(
      (fornecedor) =>
        categoriaFiltro === "" ||
        fornecedor.categorias.includes(categoriaFiltro)
    ),
  [fornecedores, categoriaFiltro]
);
return fornecedoresFiltrados;
};

const ListaFornecedores = ({ fornecedores, onRemove }) => {
  const [categoriaFiltro, setCategoriaFiltro] = useState("");

  const fornecedoresFiltrados = useFiltrarFornecedores(
    fornecedores,
    categoriaFiltro
  );

  const handleCategoriaFiltroChange = (text) => {
    setCategoriaFiltro(text);
  };

  const handleDelete = (fornecedor) => {
    Alert.alert(
      "Confirmar exclusão",
      "Você realmente deseja excluir este fornecedor?",
      [
        { text: "Cancelar", style: "cancel" },
        { text: "Excluir", onPress: () => onRemove(fornecedor.id) },
      ]
    );
  };

  return (
    <View style={styles.container}>
      <TextInput
        style={styles.input}
        placeholder="Pesquisar por Categoria"
        value={categoriaFiltro}
        onChangeText={handleCategoriaFiltroChange}
      />

      {fornecedoresFiltrados.map((fornecedor, index) => (
        <View key={index} style={styles.fornecedorContainer}>
          <Image
```



```
        source={{ uri: fornecedor.imagem }}
        style={styles.imagemFornecedor}
      />
      <View style={styles.infoContainer}>
        <Text style={styles.nomeFornecedor}>{fornecedor.nome}</Text>
        <Text style={styles.detalheFornecedor}>
          Endereço: {fornecedor.endereco}
        </Text>
        <Text style={styles.detalheFornecedor}>
          Contato: {fornecedor.contato}
        </Text>
        <Text style={styles.detalheFornecedor}>
          Categoria: {fornecedor.categorias}
        </Text>
        <TouchableOpacity
          style={styles.actionButton}
          onPress={() => handleDelete(fornecedor)}
        >
          <Text style={styles.actionText}>Apagar</Text>
        </TouchableOpacity>
      </View>
    </View>
  )))
</View>
);
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    paddingHorizontal: 16,
    backgroundColor: "#f0f0f0",
  },
  input: {
    fontSize: 16,
    padding: 16,
    marginBottom: 12,
    borderWidth: 1,
    borderColor: "#ccc",
    borderRadius: 25,
    backgroundColor: "#ffffff",
  },
  fornecedorContainer: {
    flexDirection: "row",
    marginBottom: 16,
    padding: 16,
    backgroundColor: "#ffffff",
  },
});
```



```
borderWidth: 1,
borderColor: "#ddd",
borderRadius: 12,
shadowColor: "#000",
shadowOffset: {
  width: 0,
  height: 2,
},
shadowOpacity: 0.25,
shadowRadius: 3.84,
elevation: 5,
alignItems: "center",
},
infoContainer: {
  flex: 1,
  marginLeft: 12,
},
nomeFornecedor: {
  fontSize: 20,
  fontWeight: "bold",
  color: "#333",
},
detalheFornecedor: {
  fontSize: 16,
  color: "#666",
},
imagemFornecedor: {
  width: 100,
  height: 100,
  borderRadius: 50,
},
actionButton: {
  padding: 8,
  width: 70,
  marginTop: 8,
  backgroundColor: "#ddd",
  borderRadius: 10,
},
actionText: {
  color: "#333",
},
});

export default ListaFornecedores;
```



cadastro.js

```
import * as ImagePicker from "expo-image-picker";
import React, { Component } from "react";
import { Alert, Image, StyleSheet, TextInput, View, } from "react-native";
import CustomButton from "../botao";

class CadastroFornecedor extends Component {
  state = {
    nome: "",
    endereco: "",
    contato: "",
    categorias: "",
    imagem: null,
  };

  handleCadastro = () => {
    const { nome, endereco, contato, categorias, imagem } = this.state;
    if (!nome || !endereco || !contato || !categorias) {
      this.showAlert("Erro", "Preencha todos os campos.");
      return;
    }

    const novoFornecedor = { nome, endereco, contato, categorias, imagem };
    this.props.onCadastroSubmit(novoFornecedor);
    this.clearFields();
    this.showAlert("Sucesso", "cadastrado com sucesso!");
  };

  pickImage = async () => {
    let result = await ImagePicker.launchImageLibraryAsync({
      mediaTypes: ImagePicker.MediaTypeOptions.Images,
      allowsEditing: true,
      aspect: [4, 3],
      quality: 1,
    });

    if (!result.canceled) {
      this.setState({
        imagem:
          result.assets && result.assets.length > 0
            ? result.assets[0].uri
            : null,
      });
    }
  };

  clearFields = () => {
```



```
this.setState({
  nome: "",
  endereco: "",
  contato: "",
  categorias: "",
  imagem: null,
});
});

showAlert = (title, message) => {
  Alert.alert(title, message);
};

render() {
  const { nome, endereco, contato, categorias, imagem } = this.state;
  return (
    <View style={styles.container}>
      <TextInput
        style={styles.input}
        placeholder="Nome Completo"
        value={nome}
        onChangeText={(text) => this.setState({ nome: text })}
      />
      <TextInput
        style={styles.input}
        placeholder="Endereço"
        value={endereco}
        onChangeText={(text) => this.setState({ endereco: text })}
      />
      <TextInput
        style={styles.input}
        placeholder="Contato"
        value={contato}
        onChangeText={(text) => this.setState({ contato: text })}
      />
      <TextInput
        style={styles.input}
        placeholder="Categoria/Produto"
        value={categorias}
        onChangeText={(text) => this.setState({ categorias: text })}
      />

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

      <View style={styles.buttonContainer}>
        <CustomButton title="Escolher Imagem" onPress={this.pickImage} />
        <CustomButton
```




```
        title="Cadastrar"
        color="#45b39d"
        onPress={this.handleCadastro}
      />
    </View>
  </View>
);
}
```

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    paddingHorizontal: 20,
  },
  input: {
    width: 350,
    fontSize: 16,
    marginBottom: 16,
    paddingHorizontal: 12,
    paddingVertical: 10,
    backgroundColor: "ffffff",
    shadowColor: "black",
    shadowOffset: {
      width: 0,
      height: 2,
    },
    shadowOpacity: 0.25,
    shadowRadius: 3.84,
    elevation: 5,
    borderRadius: 12,
  },
  image: {
    width: 160,
    height: 160,
    marginTop: 12,
    borderRadius: 100,
    marginBottom: 12,
  },
  buttonContainer: {
    flexDirection: "column-reverse",
    justifyContent: "flex-start",
    width: 350,
  },
});
```



```
export default CadastroFornecedor;
```

botão.js

```
import React, { Component } from "react";
import { StyleSheet, Text, TouchableOpacity } from "react-native";

class CustomButton extends Component {
  render() {
    const { title, onPress, color = "#5180D2" } = this.props;
    return (
      <TouchableOpacity style={[styles.buttonImg, { backgroundColor: color }]}
onPress={onPress}>
        <Text style={styles.buttonText}>{title}</Text>
      </TouchableOpacity>
    );
  }
}

const styles = StyleSheet.create({
  buttonImg: {
    padding: 12,
    borderRadius: 12,
    marginBottom: 8,
    alignItems: "center",
  },
  buttonText: {
    color: "ffffff",
    fontSize: 16,
    fontWeight: "bold",
  },
});

export default CustomButton;
```



Conclusão

O aplicativo fará o cadastro de fornecedores com os campos nome, endereço, contato e categoria/produto. Sendo possível ver todos os fornecedores cadastrados, e podendo até excluí-los caso seja necessário.

Tela responsiva e rotação de tela adaptativa.