



FACULDADE ESTÁCIO DE SÁ

CURSO: DESENVOLVIMENTO FULL STACK

4º SEMESTRE – MATRÍCULA 202302595341

Repositório GitHub - [alaimalmeida/vamosCriarUmApp \(github.com\)](https://github.com/alaimalmeida/vamosCriarUmApp)

ALAIM ALMEIDA DE OLIVEIRA

Vamos criar um App

Adicionando interatividade com o React Native

Salvador – BA

2024

1. Introdução

Esta atividade tem como objetivo por em prática tudo ou boa parte do que foi aprendido nas Missões Conceituais, sendo capaz de criar um aplicativo do zero com React Native e algumas bibliotecas.

Com base no que foi aprendido nos conteúdos estudados, foi desenvolvido um app cujo o nome é Meeting que é um aplicativo para fazer cadastro de fornecedores de uma empresa fictícia, com cadastro de nome, endereço, contato, categoria de produtos e cadastro de imagem.

Nesse aplicativo, vou utilizado uma estrutura chamada Expo que facilita o desenvolvimento de aplicativos Android e IOS.

Com esse framework, criamos o aplicativo de forma mais fácil e podemos espelhar tanto em dispositivos android's, IOS e Web para podermos testar as aplicações.

2. Instalação

Para instalar o React no Windows, é preciso instalar primeiro o Node.js e o npm(Node Package Manager) no sistema.

Feito a instalação corretamente no site do Node.js, é dado alguns comandos para ter certeza que ambos estejam instalados no sistema operacional na sua versão atual. Os comandos são **node -v** (para saber a versão do Node.js) e **npm -v**(para saber a versão do NPM).

Logo a pois a checagem das versões do React e Npm, seguimos para instalação do Expo CLI com o seguinte comando: **npm install -g expo-cli**. Para criar um projeto, devemos digitar o seguinte comando **expo init MeuProjeto** (onde tem "MeuProjeto" varia de acordo com o nome desejado para a criação do aplicativo).

Navegando até a dependência do projeto, **cd MeuProjeto** para começar a criar seus componentes e toda a estrutura ficando da seguinte forma:

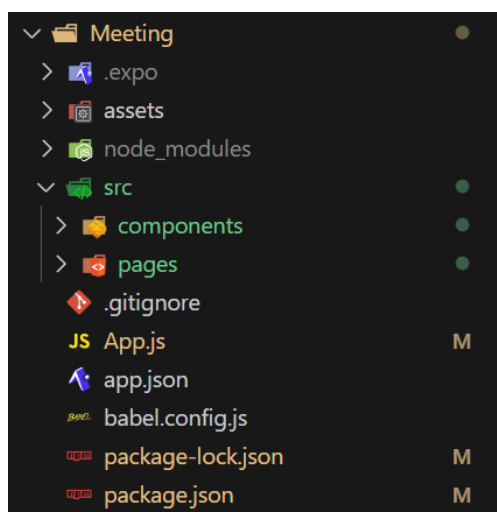


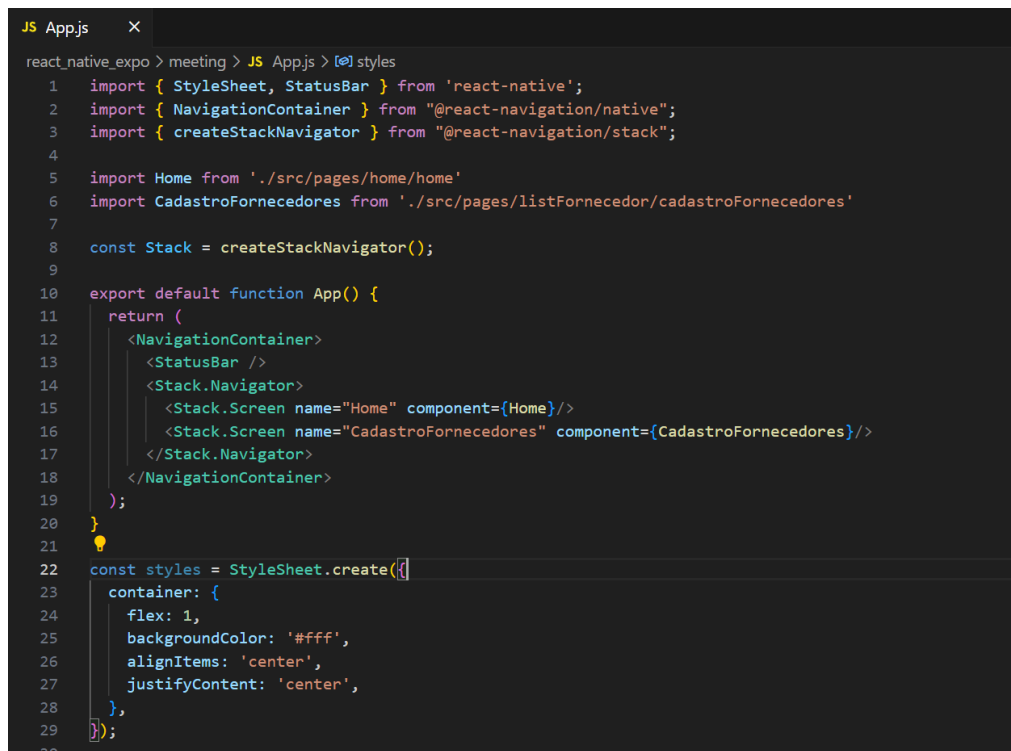
Imagem 01 - estrutura

3. Estrutura

No arquivo App.js, será importado as bibliotecas importantes para poder configurar o projeto, sendo importado StyleSheet e StatusBar, um para poder usar o CSS diretamente no mesmo arquivo e o StatusBar para poder aparecer as informações na parte superior do dispositivo virtual.

Logo em seguida, fazemos a importação das páginas que é a Home e a CadastroFornecedores para poder incrementar na navegação.

Criando uma função padrão para o App, implementamos o return para retornar um NavigationContainer, onde será configurado as telas/ páginas que compõe o projeto. Segue abaixo a estrutura do arquivo:



```
JS App.js X
react_native_expo > meeting > JS App.js > [?] styles
1  import { StyleSheet, StatusBar } from 'react-native';
2  import { NavigationContainer } from "@react-navigation/native";
3  import { createStackNavigator } from "@react-navigation/stack";
4
5  import Home from '../src/pages/home/home'
6  import CadastroFornecedores from '../src/pages/listFornecedor/cadastroFornecedores'
7
8  const Stack = createStackNavigator();
9
10 export default function App() {
11   return (
12     <NavigationContainer>
13       <StatusBar />
14       <Stack.Navigator>
15         <Stack.Screen name="Home" component={Home}/>
16         <Stack.Screen name="CadastroFornecedores" component={CadastroFornecedores}/>
17       </Stack.Navigator>
18     </NavigationContainer>
19   );
20 }
21
22 const styles = StyleSheet.create({
23   container: {
24     flex: 1,
25     backgroundColor: '#fff',
26     alignItems: 'center',
27     justifyContent: 'center',
28   },
29 });
```

Imagem 02 – estrutura do App.js

Foi criado duas pastas, dentro de src, uma pages, como o nome mesmo informa, fica as páginas como a Home e a CadastroFornecedores e na outra pasta, fica os componentes fornecedor e updateImagem. Segue abaixo a página home e cadastroFornecedor:

```
JS App.js JS home.js X JS cadastroFornecedores.js
react_native_expo > meeting > src > pages > home > JS home.js > Home
1  import React from "react";
2  import {View, Text, StyleSheet} from 'react-native'
3
4  export default function Home({navigation}){
5    return(
6      <View style={styles.container}>
7        <Text style={styles.textTitle}>Meeting</Text>
8
9        <Text
10         style={styles.button}
11         onPress={()=> navigation.navigate('CadastroFornecedores')}
12         >Cadastro </Text>
13      </View>
14    )
15  }
16
```

Imagem 03 – estrutura da página home

```
17  const styles = StyleSheet.create({
18    textTitle:{
19      color: "#ff0043",
20      fontSize: 70,
21      fontWeight: "bold",
22      textAlign: 'center',
23    },
24    button:{
25      textAlign: 'center',
26      backgroundColor: 'green',
27      width: 100,
28      borderRadius: 5,
29      padding: 10,
30      color: 'ffffff',
31      justifyContent: 'center',
32      fontSize: 16
33    },
34    container:{
35      justifyContent: 'center',
36      flex: 1,
37      textAlign: 'center',
38      alignItems: 'center'
39    }
40  });
```

Imagem 04 – estrutura do css

```
JS App.js JS home.js JS cadastroFornecedores.js X
react_native_expo > meeting > src > pages > listFornecedor > JS cadastroFornecedores.js > ...
1  import React from "react";
2  import {View, StyleSheet} from 'react-native'
3  import Fornecedores from "../../components/fornecedor";
4
5  export default function CadastroFornecedores(){
6    return(
7      <View style={styles.container}>
8        <Fornecedores/>
9      </View>
10    )
11  }
12
13  const styles = StyleSheet.create({
14    container: {
15      flex: 1,
16      backgroundColor: 'ffff',
17      alignItems: 'center',
18      justifyContent: 'center',
19    },
20  });
21
```

Imagem 05 – estrutura cadastroFornecedores

Finalizando a parte das páginas, vamos para a parte dos componentes, onde criamos cada parte do aplicativo, onde separamos por componente cada processo de uma página, ou melhor, cada função que determinado componente irá executar no aplicativo. Muitos desses componentes são criados para reduzir o trabalho, onde cada componente fica separado do código principal do aplicativo. Quando é preciso fazer alguma modificação em determinada parte do aplicativo, caso essa parte esteja em um componente, fica mais fácil e ágil encontrar onde está o problema ou o ponto para alteração. Segue abaixo os componentes criados que compõe o aplicativo:

```
JS index.js X
react_native_expo > meeting > src > components > fornecedor > JS index.js > Fornecedores

1 import React, { useState } from "react";
2 import { View, Text, TextInput, TouchableOpacity, FlatList, Image, ScrollView } from "react-native";
3 import styles from "../style";
4 import * as ImagePicker from 'expo-image-picker';
5
6 export default function Fornecedores(){
7   const [imagem, setImagem] = useState(null)
8   const [nome, setNome] = useState(null)
9   const [endereco, setEndereco] = useState(null)
10  const [contato, setContato] = useState(null)
11  const [catProduto, setCatProduto] = useState(null)
12  const [imageUri, setImageUri] = useState(null);
13  const [fornecedor, setfornecedor] = useState([]);
14
15  const logo = {
16    uri: 'https://reactnative.dev/img/tiny_logo.png',
17    width: 100,
18    height: 100,
19  };
20
21  const selectImage = async () => {
22    const permissionResult = await ImagePicker.requestMediaLibraryPermissionsAsync();
23    if (permissionResult.granted === false) {
24      alert('Permission to access gallery is required!');
25      return;
26    }
27  }
```

Imagem 06 – código do componente fornecedor

```
28 const pickerResult = await ImagePicker.launchImageLibraryAsync({
29   mediaTypes: ImagePicker.MediaTypeOptions.Images,
30   quality: 1,
31 });
32
33 if (!pickerResult.canceled) {
34   setImageUri(pickerResult.assets[0].uri);
35 }
36
37
38 const addFornecedor = () => {
39   if (nome && endereco && contato && catProduto) {
40     const newfornecedor = {
41       id: Math.random().toString(),
42       nome: nome,
43       endereco,
44       contato,
45       catProduto,
46       imageUri,
47     };
48
49     setfornecedor([...fornecedor, newfornecedor]);
50
51     // Limpar os campos após adicionar
52     setNome('');
53     setEndereco('');
54     setContato('');
```

Imagem 07 – código do componente fornecedor

```

55         setCatProduto('');
56         setImagUri(null);
57     }else {
58         alert('Por favor, preencha todos os campos!');
59     }
60 };
61
62 function fornecedoresCadastrados(){
63     return setListform (nome + "\n" + endereco + "\n" + contato + "\n" + catProduto)
64 }
65
66 function validationFornecedores(){
67     if (nome != null && endereco != null && contato != null && catProduto != null){
68         fornecedoresCadastrados()
69         return
70     }
71 }
72
73 return(
74     <ScrollView style={styles.boxTitle}>
75         <Text style={styles.textTitle}>Meeting</Text>
76         <View style={styles.boxConteudo}>
77             <View style={styles.boxCadastro}>
78                 <Text style={styles.tituloCadastro}>Nome:</Text>
79                 <TextInput
80                     onChangeText={setNome}

```

Imagem 08 – código do componente fornecedor

```

81         value={nome}
82         style={styles.input}
83         placeholder="Ex: Amazon"
84     />
85
86     <Text style={styles.tituloCadastro}>Endereço:</Text>
87     <TextInput
88         onChangeText={setEndereco}
89         value={endereco}
90         style={styles.input}
91         placeholder="Ex: Rua das araras"
92     />
93
94     <Text style={styles.tituloCadastro}>Contato:</Text>
95     <TextInput
96         onChangeText={setContato}
97         value={contato}
98         style={styles.input}
99         placeholder="Ex: (71) 998735858"
100         keyboardType="numeric"
101     />
102
103     <Text style={styles.tituloCadastro}>Categoria de produtos:</Text>
104     <TextInput
105         value={catProduto}
106         onChangeText={setCatProduto}
107         style={styles.input}

```

Imagem 09 – código do componente fornecedor

```

108         placeholder="Ex: Carro"
109     />
110
111     <View style={styles.botoes}>
112         <TouchableOpacity
113             style={styles.buttonImg}
114             onPress={selectImage}
115         >
116             <Text style={styles.TextButtonCadastro}>Add imagem </Text>
117         </TouchableOpacity>
118
119         <TouchableOpacity
120             style={styles.buttonCadastro}
121             onPress={addFornecedor}
122         >
123             <Text style={styles.TextButtonCadastro}>Cadastrar</Text>
124         </TouchableOpacity>
125     </View>
126 </View>
127
128 <FlatList
129     style={styles.list}
130     data={([...fornecedor].reverse())}
131     keyExtractor={({item}) => item.id}
132     renderItem={({ item }) => (
133         <View style={styles.listFornecedor}>
134             <View style={styles.imgFornecedor}>
135                 {item.imageUri && (
136                     <Image source={{ uri: item.imageUri }} style={styles.supplierImage} />

```

Imagem 10 – código do componente fornecedor

```

137         })
138     </View>
139     <View>
140       <Text style={styles.titleText}>Nome: {item.nome}</Text>
141       <Text style={styles.titleText}>Endereço: {item.endereco}</Text>
142       <Text style={styles.titleText}>Contato: {item.contato}</Text>
143       <Text style={styles.titleText}>Categoria: {item.catProduto}</Text>
144     </View>
145   </View>
146   })
147   scrollEnabled={false}
148 </>
149 </View>
150 </ScrollView>
151 );
152 }

```

Imagem 11 – código do componente fornecedor

```

import { StyleSheet } from "react-native";

const styles = StyleSheet.create({
  boxTitle: {
    paddingTop: 5,
    flex: 1,
    backgroundColor: '#dddddd',
    width: '100%',
    paddingBottom: 30,
  },
  boxConteudo: {
    backgroundColor: '#ffffff',
    width: '100%',
    borderTopLeftRadius: 50,
    borderTopRightRadius: 50,
    paddingTop: 0,
    paddingBottom: 0,
    height: '100%',
    marginBottom: 40,
  },
  textTitle: {
    color: '#ff0043',
    fontSize: 36,
    fontWeight: 'bold',
    marginBottom: 10,
    textAlign: 'center',
  },
});

```

Imagem 12 – css do componente fornecedor

```

28   boxCadastro: {
29     marginTop: 20,
30     paddingLeft: 25,
31     paddingRight: 25,
32     paddingBottom: 10,
33   },
34   tituloCadastro: {
35     fontSize: 18,
36     alignSelf: 'left',
37     alignItems: 'center',
38     marginBottom: 2
39   },
40   input: {
41     height: 30,
42     borderColor: 'gray',
43     borderWidth: 1,
44     marginBottom: 5,
45     paddingHorizontal: 15,
46     alignItems: 'left',
47     borderRadius: 5
48   },
49   img: {
50     alignItems: 'center',
51     marginBottom: 20,
52   },

```

Imagem 13 – css do componente fornecedor

```

53     buttonCadastro:{
54         backgroundColor: 'green',
55         padding: 5,
56         width: 150,
57         alignItems: 'center',
58         borderRadius: 5,
59         marginTop: 5,
60     },
61     buttonImg:{
62         backgroundColor: '#2196f3',
63         padding: 5,
64         width: 150,
65         alignItems: 'center',
66         borderRadius: 5,
67         marginTop: 5,
68         marginRight: 5
69     },
70     botoes:{
71         flex: 1,
72         flexDirection: 'row',
73         flexWrap: 'wrap',
74     },
75     TextButtonCadastro:{
76         color: 'ffffff',
77         fontSize: 18,
78     },

```

Imagem 14 – css do componente fornecedor

```

79     resultadoFornecedores:{
80         fontSize: 40,
81         fontWeight: 'bold'
82     },
83     listFornecedor: {
84         padding: 10,
85         borderBottomColor: '#ccc',
86         borderBottomWidth: 1,
87         backgroundColor: '#f9f9f9',
88         flexDirection: 'row',
89         flexWrap: 'wrap',
90         height: 100
91     },
92     titleText: {
93         fontSize: 14,
94     },
95     imgFornecedor:{
96         marginRight: 10,
97         justifyContent: 'center'
98     },
99     supplierImage: {
100         width: 80,
101         height: 80,
102     },
103     list:{
104         marginTop: 0
105     }

```

Imagem 15 – css do componente fornecedor

```

106 });
107
108 export default styles

```

Imagem 16 – css do componente fornecedor


```

1  import React, { useState } from 'react';
2  import { View, Button, Image, FlatList, StyleSheet } from 'react-native';
3  import * as ImagePicker from 'expo-image-picker';
4
5  const App = () => {
6    const [images, setImages] = useState([]);
7
8    const selectImage = async () => {
9      const permissionResult = await ImagePicker.requestMediaLibraryPermissionsAsync();
10     if (permissionResult.granted === false) {
11       alert('Permissão para acessar a galeria é necessária!');
12       return;
13     }
14
15     const pickerResult = await ImagePicker.launchImageLibraryAsync({
16       mediaTypes: ImagePicker.MediaTypeOptions.Images,
17       quality: 1,
18     });
19
20     if (!pickerResult.canceled) {
21       setImages([...images, pickerResult.assets[0].uri]);
22     }
23   };
24
25   return (
26     <View style={styles.container}>
27       <Button title="Adicionar Imagem" onPress={selectImage} />
28

```

Imagem 17 – código do componente updateImagem

```

29     <FlatList
30       data={images}
31       keyExtractor={({item, index}) => index.toString()}
32       renderItem={({item}) => (
33         <Image source={{uri: item}} style={styles.image} />
34       )}
35     />
36   </View>
37 );
38 };
39
40 const styles = StyleSheet.create({
41   container: {
42     flex: 1,
43     padding: 5,
44     backgroundColor: '#fff',
45   },
46   image: {
47     width: '100%',
48     height: 100,
49     marginBottom: 10,
50   },
51 });
52
53 export default App;
54
55

```

Imagem 18 – código do componente updateImagem

Com isso, todos os códigos foram inseridos neste relatório para mostrar a forma de construir um aplicativo do zero com React Native.

Com base nos estudos e colocando em prática esse exercício, seguindo todos os passos, finalizamos o mesmo e estamos apito na construção de um app do início ao final de forma fácil e ágil.