

Documentação da missão 1

Nome: Israel dos Santos Hamdan D'Araújo

matricula: 202303592086

email: contatolsraelHamdan@gmail.com

Instruções

Requisitos:

Para rodar a aplicação é necessário o Nodejs e o Expo instalados na sua máquina. Além disso é necessário um dispositivo android ou iOS com o app Expo Go instalado.

Primeiros passos

1. Clonar o projeto <https://github.com/IsraelHamdan/Missao-pratica-1.git>
2. Instalar dependências, dentro da pasta missao_pratica, usar o comando `npm install` para instalar todas as dependências do projeto
3. Colocar o seu endereço de ip dentro da variável `api`, no arquivo `useFornecedores.ts`, para isso é necessário saber o seu endereço de IP, isso pelo motivo de que, para rodar o projeto em um dispositivo mobile, vc precisa do endereço da API dele, e como não estamos rodando tudo dentro do mesmo dispositivo, o localhost não funciona!
4. na pasta missao_pratica usar o comando `json-server --watch db.json --host 0.0.0.0 --port 3000` para iniciar o backend da aplicação
5. Depois de iniciar o servidor do backend é necessário iniciar o Expo, então dentro da pasta missao_pratica dar o comando `npx expo start`
6. Escanear o QR code com o app do expo então a aplicação irá iniciar

CÓDIGOS

Tela de inicia



```

21 const Cadastro: React.FC<Props> = ({ navigation }) => {
22   const [nome, setNome] = useState<string>("");
23   const [cnpj, setCnpj] = useState<string>("");
24   const [email, setEmail] = useState<string>("");
25   const [telefone, setTelefone] = useState<string>("");
26   const [imagem, setImagem] = useState<string>("");
27
28   const imagePickerRef = useRef<ImagePickerHandle>(null);
29
30   const {
31     fornecedores,
32     isLoading,
33     errorMessage,
34     findFornecedores,
35     postFornecedoresOnJSON,
36   } = useFornecedores();
37
38   useEffect(() => {
39     findFornecedores();
40   }, []);
41
42   const handlePostFornecedor = async () => {
43     const novoFornecedor: Fornecedor = {
44       nome,
45       email,
46       telefone,
47       cnpj,
48       imagem,
49     };
50
51     try {
52       await postFornecedoresOnJSON(novoFornecedor);
53       navigation.navigate("Home");
54     } catch (errorMessage) {
55       console.error(errorMessage);
56     }
57   };
58
59   return (
60     <SafeAreaView style={Styles.container}>
61       <ScrollView>
62         </ScrollView>
63     </SafeAreaView>
64   );
65 };
66
67 export default Cadastro;

```

Tela de cadastro

```

const Cadastro: React.FC<Props> = ({ navigation }) => {
  const [nome, setNome] = useState<string>("");
  const [cnpj, setCnpj] = useState<string>("");
  const [email, setEmail] = useState<string>("");
  const [telefone, setTelefone] = useState<string>("");
  const [imagem, setImagem] = useState<string>("");

  const imagePickerRef = useRef<ImagePickerHandle>(null);

  const {
    fornecedores,
    isLoading,
    errorMessage,
    findFornecedores,
    postFornecedoresOnJSON,
  } = useFornecedores();

  useEffect(() => {
    findFornecedores();
  }, []);

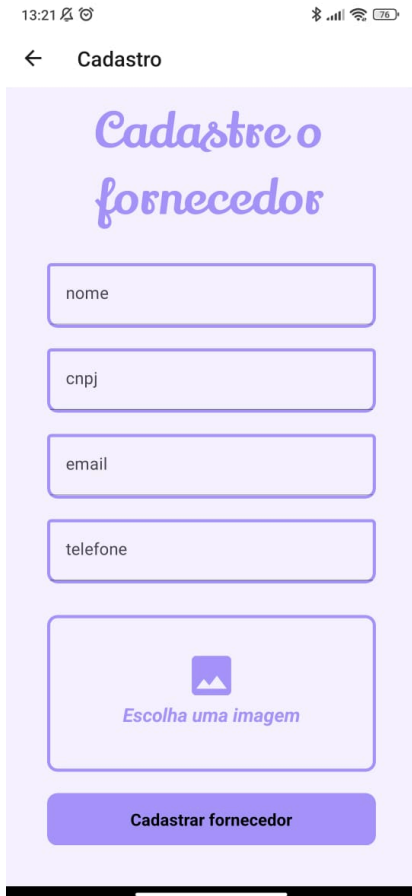
  const handlePostFornecedor = async () => {
    const novoFornecedor: Fornecedor = {
      nome,
      email,
      telefone,
      cnpj,
      imagem,
    };

    try {
      await postFornecedoresOnJSON(novoFornecedor);
      navigation.navigate("Home");
    } catch (errorMessage) {
      console.error(errorMessage);
    }
  };

  return (
    <SafeAreaView style={Styles.container}>
      <ScrollView>
        </ScrollView>
    </SafeAreaView>
  );
};

export default Cadastro;

```



Tela de listagem

```
missao_pratica > components > ListaFornecedor > @ ListaFornecedor.tsx > @ ListaFornecedores
1  import React, { useEffect } from "react";
2  import Styles from "../ListaFornecedor/Styles";
3  import { FlatList, SafeAreaView, View, Text, StyleSheet } from "react-native";
4  import { useFornecedores } from "../../hooks/useFornecedores";
5  import { ActivityIndicator, Button } from "react-native-paper";
6  import CardFornecedor from "../CardFornecedor/CardFornecedor";
7  import { Props } from "../../utils/Props";
8
9  const ListaFornecedores: React.FC<Props> = ({ navigation }) => {
10
11    const { fornecedores, isLoading, errorMessage, findFornecedores } =
12      useFornecedores();
13    useEffect(() => {
14      findFornecedores();
15    }, []);
16
17    if (isLoading) {
18      return <ActivityIndicator size="large" color="#A594F9" />;
19    }
20
21    if (errorMessage) {
22      return (
23        <View>
24          <Text>{errorMessage}</Text>
25        </View>
26      );
27    }
28
29    return (
30      <SafeAreaView style={Styles.container}>
31        <Button
32          onPress={() => {
33            navigation.navigate("Cadastro");
34          }}
35          style={styles.btn}>
36          Cadastrar Fornecedor
37        </Button>
38        <FlatList
39          data={fornecedores}
40          keyExtractor={(item, index) => index.toString()}
41          renderItem={({ item }) => <CardFornecedor fornecedor={item} />}
42        />
43      </SafeAreaView>
44    );
45
46    export default ListaFornecedores;
```

Infelizmente agora eu não consigo anexar a imagem da listagem de produtos, pois estou com um problema de rede e não estou conseguindo fazer a parte de cadastro e de listagem funcionar por causa da rede

Seletor de imagens

```

15 const ImagePicker = forwardRef((props, ref) => {
16   const [visible, setVisible] = useState<boolean>(false);
17   const { handleTakeGalleryImage, uri } = useMedia();
18   const showModal = () => setVisible(true);
19   const hideModal = () => setVisible(false);
20
21   useImperativeHandle(ref, () => ({
22     showModal,
23     hideModal,
24   }));
25
26   return (
27     <Provider>
28       <View style={Styles.view}>
29         {uri ? (
30           <View>
31             <Icon size={100} color={"#A594F9"} name="checkmark-circle" />
32           </View>
33         ) : (
34           <Portal>
35             {visible ? (
36               <View style={Styles.btn}>
37                 <Icon size={50} source="image" color={"#A594F9"} />
38                 <Text style={Styles.textBtn}>Escolha uma imagem</Text>
39               </View>
40             ) : (
41               <Modal>
42                 visible={visible}
43                 onDismiss={hideModal}
44                 contentContainerStyle={Styles.model}>
45                 <Text>Escolha da galeria</Text>
46                 <Button mode="contained-tonal" onPress={handleTakeGalleryImage}>
47                   <View style={Styles.icon}>
48                     <Icon size={48} color={"#A594F9"} name="image" />
49                   </View>
50                 </Button>
51               </Modal>
52             )}
53           </Portal>
54         )}
55       </View>
56     </Provider>
57   );
58 });

```

13:27

← Cadastro

Cadastre o fornecedor

nome

cnpj

email

telefone

Escolha da galeria

Cannot connect to Metro....