

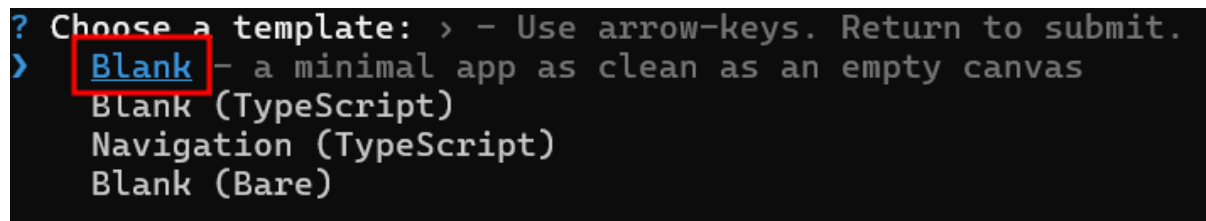
## Criação de tela de Login/Registro

Nessa aula vamos criar telas de login e registro e para isso vamos criar um projeto. Abra o VS Code na pasta onde você deseja criar o seu projeto e em seguida abra o terminal do Vs Code e digite o comando:

```
npx create-expo-app --template
```

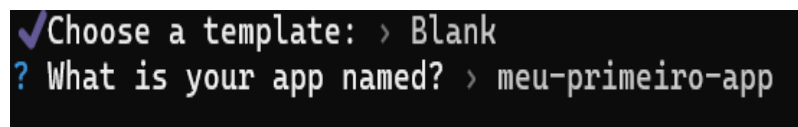
Na hora de colar no terminal, certifique-se que tem dois traços (--) antes da palavra "template", para não dar erro na hora da instalação.

Em seguida deverá mostrar a seguinte mensagem: Escolha a opção "blank" (Primeira opção)



```
? Choose a template: > - Use arrow-keys. Return to submit.  
> Blank - a minimal app as clean as an empty canvas  
Blank (TypeScript)  
Navigation (TypeScript)  
Blank (Bare)
```

Depois dê um nome ao seu projeto. Exemplo: meu-primeiro-app:



```
✓ Choose a template: > Blank  
? What is your app named? > meu-primeiro-app
```

Em seguida será criado o seu projeto, quando finalizado entre na pasta do seu projeto para realizar a instalação das dependências que serão utilizadas no projeto:

```
npm install @react-navigation/native
```

```
npm install @react-navigation/native-stack
```

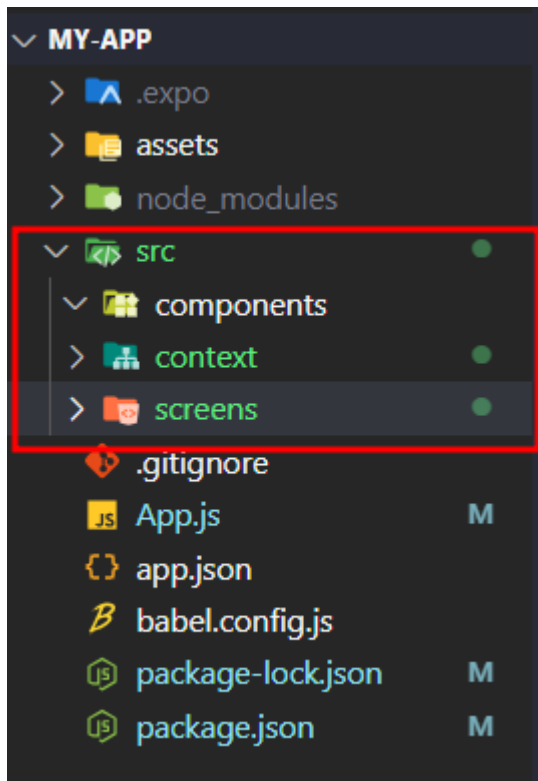
```
npx expo install react-native-screens
```

```
npm install react-native-safe-area-context
```

Após finalizado a instalação das dependências vamos iniciar nosso projeto:

```
npm start
```

Também inicie o emulador no android studio, e após o dispositivo ligar, pressione "a" no terminal para conectar o projeto ao dispositivo. Feito isso, vamos estruturar o nosso código. Na raiz do projeto crie a pasta "src" e dentro dela crie mais 3 pastas: components, context e screens. Deverá ficar assim:



Agora dentro da pasta context crie o arquivo “createContext.js”

### **createContext.js**

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

export default (reducer, actions, initialValue) => {
  const Context = React.createContext();

  const Provider = ({ children }) => {
    const [state, dispatch] = useReducer(reducer, initialValue);

    const customFunctions = {};
    Object.keys(actions).forEach(
      (key) => (customFunctions[key] = actions[key](dispatch))
    );

    return (
```

```
    <Context.Provider value={{ state, ...customFunctions }}>
      {children}
    </Context.Provider>
  );
};
return { Context, Provider };
};
```

Ainda dentro da mesma pasta crie outro arquivo “authContext.js”

### **authContext.js**

```
import createContext from "./createContext";

const initialState = {};

const reducer = (state, action) => {
  switch (action.type) {
    default:
      return state;
  }
};

const teste = (dispatch) => {
  return (args) => {
    console.log(args);
  };
};
```

```
export const { Context, Provider } = createContext(
  reducer,
  { teste },
  initialState
);
```

Esses arquivos vão nos auxiliar para fazer a autenticação no login mais a frente. Agora vamos editar o arquivo App.js

### **App.js**

```
import React from "react";
import { NavigationContainer } from "@react-navigation/native";
import { createNativeStackNavigator } from "@react-navigation/native-stack";
import SignIn from "../src/screens/SignIn";
import SignUp from "../src/screens/SignUp";

import { Provider } from "../src/context/authContext";
import { SafeAreaProvider } from "react-native-safe-area-context";

const Stack = createNativeStackNavigator();

const App = () => {
  return (
    <NavigationContainer>
      <Stack.Navigator screenOptions={{ headerShown: false }}>
        <Stack.Screen name="SignIn" component={SignIn} />
        <Stack.Screen name="SignUp" component={SignUp} />
      </Stack.Navigator>
    </NavigationContainer>
  );
};
```

```
);  
};  
  
export default () => {  
  return (  
    <Provider>  
      <SafeAreaProvider>  
        <App />  
      </SafeAreaProvider>  
    </Provider>  
  );  
};
```

E agora, dentro da pasta screens vamos criar o arquivo “SignIn” e nela vamos criar nossa tela de login.

### **SignIn.js**

```
import React, { useState, useContext } from "react";  
import { View, StyleSheet, TouchableOpacity } from "react-native";  
import { SafeAreaView } from "react-native-safe-area-context";  
import { Button, Text, TextInput } from "react-native-paper";  
import { Context } from "../context/authContext";  
  
const SignIn = ({ navigation }) => {  
  const { state, teste } = useContext(Context);  
  
  const [email, setEmail] = useState("");  
  const [password, setPassword] = useState("");
```

```
const [showPassword, setShowPassword] = useState(true);

return (
  <SafeAreaView style={styles.container}>
    <Text style={styles.login}>Login</Text>
    <TextInput
      style={styles.textInput}
      label="E-mail"
      value={email}
      onChangeText={(text) => setEmail(text)}
      left={<TextInput.Icon icon="at" size={25} color="black" />}
    />
    <TextInput
      style={styles.textInput}
      label="Senha"
      value={password}
      onChangeText={(text) => setPassword(text)}
      left={<TextInput.Icon icon="lock" size={25} color="black" />}
      secureTextEntry={showPassword}
      right={
        showPassword ? (
          <TextInput.Icon
            icon="eye"
            size={25}
            color="black"
            onPress={() => setShowPassword(!showPassword)}
          />
        ) : (
```

```

        <TextInput.Icon
          icon="eye-off"
          size={25}
          color="black"
          onPress={() => setShowPassword(!showPassword)}
        />
      )
    }
  />

  <Button mode="contained" style={styles.loginButton}>
    Login
  </Button>

  <TouchableOpacity onPress={() => navigation.navigate("SignUp")}>
    <Text>
      Não tem uma conta?{" "}
      <Text style={styles.createAccountText}>Crie uma</Text>
    </Text>
  </TouchableOpacity>
</SafeAreaView>

);
};

const styles = StyleSheet.create({
  container: {
    marginTop: "40%",
    alignSelf: "center",
    width: "80%",
  },
});

```

```
login: {
  fontSize: 40,
  fontWeight: "bold",
  marginBottom: 10,
},
textInput: {
  marginBottom: 10,
},
loginButton: {
  padding: 5,
  margin: 20,
  width: "50%",
  alignSelf: "center",
},
createAccountText: {
  fontWeight: "bold",
  color: "#6200ee",
},
});

export default SignIn;
```

Note que as tags `Button`, `TextInput` e `Text` vem do nosso import do `react-native-paper`. São elas que aplicam alguns dos estilos do nosso projeto, assim como também colocamos os ícones no código. Agora vamos criar a nossa tela de registro, para isso crie na mesma pasta o arquivo “`SignUp`”.

### **SignUp.js**

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



```
import { StyleSheet, TouchableOpacity } from "react-native";
import { Text, TextInput, Button } from "react-native-paper";
import { SafeAreaView } from "react-native-safe-area-context";

const SignUp = ({ navigation }) => {
  const [name, setName] = useState("");
  const [text, setText] = useState("");
  const [password, setPassword] = useState("");

  const [showPassword, setShowPassword] = useState(true);

  return (
    <SafeAreaView style={styles.container}>
      <Text style={styles.createAccount}>Criar conta</Text>
      <TextInput
        style={styles.textInput}
        label="Nome"
        mode="flat"
        left={<TextInput.Icon icon="account" size={25} color="black" />}
        value={name}
        onChangeText={(text) => setName(text)}
      />
      <TextInput
        style={styles.textInput}
        label="E-mail"
        mode="flat"
        left={<TextInput.Icon icon="at" size={25} color="black" />}
        value={text}
      />
    </SafeAreaView>
  );
};
```

```
onChangeText={(text) => setText(text)}
```

```
/>
```

```
<TextInput
```

```
style={styles.textInput}
```

```
label="Senha"
```

```
mode="flat"
```

```
secureTextEntry={showPassword}
```

```
left={<TextInput.Icon icon="lock" size={25} color="black" />}
```

```
right={
```

```
showPassword ? (
```

```
<TextInput.Icon
```

```
icon="eye"
```

```
size={25}
```

```
color="black"
```

```
onPress={() => setShowPassword(!showPassword)}
```

```
/>
```

```
): (
```

```
<TextInput.Icon
```

```
icon="eye-off"
```

```
size={25}
```

```
color="black"
```

```
onPress={() => setShowPassword(!showPassword)}
```

```
/>
```

```
)
```

```
}
```

```
value={password}
```

```
onChangeText={(text) => setPassword(text)}
```

```
    />

    <Button style={styles.createButton} mode="contained">

      Criar

    </Button>

    <TouchableOpacity onPress={() => navigation.navigate("SignIn")}>

      <Text>

        Já tem uma conta? <Text style={styles.loginText}>Faça o login</Text>

      </Text>

    </TouchableOpacity>

  </SafeAreaView>

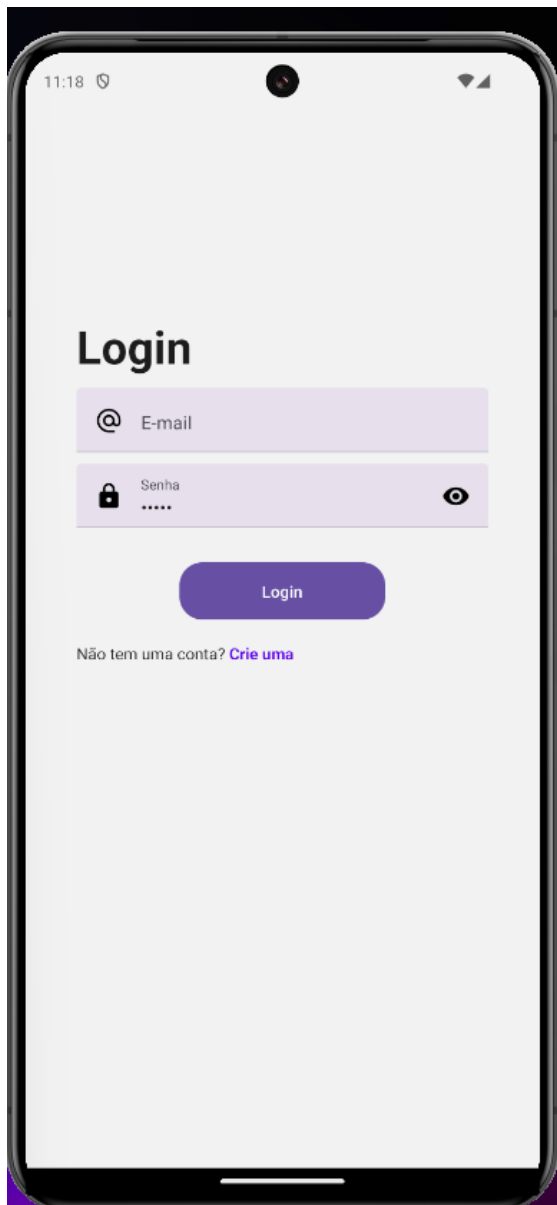
);

};

const styles = StyleSheet.create({
  container: {
    marginTop: "40%",
    alignSelf: "center",
    width: "80%",
  },
  createAccount: {
    fontSize: 40,
    fontWeight: "bold",
    marginBottom: 10,
  },
  textInput: {
    marginBottom: 10,
  },
});
```

```
createButton: {  
  padding: 5,  
  margin: 20,  
  width: "50%",  
  alignSelf: "center",  
},  
loginText: {  
  fontWeight: "bold",  
  color: "#6200ee",  
},  
});  
  
export default SignUp;
```

Com isso, nossas páginas de login já estão prontas, a sua página de Login deve estar assim:



E a sua página de Registro deverá estar assim:

