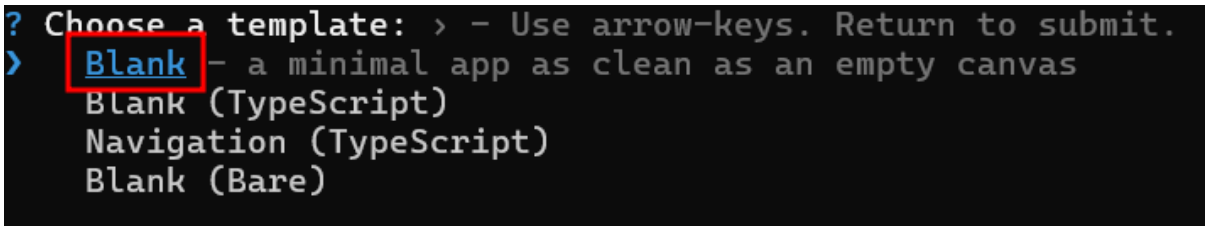
Exemplo de Projeto com Cards

Nessa aula vamos criar um app com um menu lateral com algumas funcionalidades e um dashboard com exemplo de cards e para isso vamos criar um novo 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:

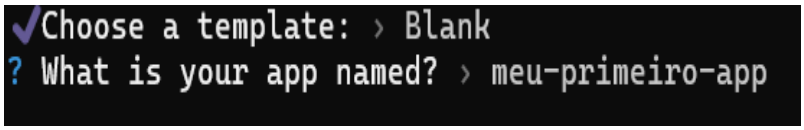
```
npx create-expo-app --template
```

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)



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



Em seguida será criado o seu projeto, quando finalizado entre na pasta do seu projeto e abra o arquivo package.json

package.json

```
{
 "name": "awesomeproject",
 "version": "1.0.0",
 "main": "node_modules/expo/AppEntry.js",
 "scripts": {
  "start": "expo start",
  "android": "expo start --android",
  "ios": "expo start --ios",
  "web": "expo start --web"
 },
 "dependencies": {
```

```
    "@expo/vector-icons": "^13.0.0",
    "@react-native-community/datetimepicker": "6.7.3",
    "@react-navigation/drawer": "^6.6.3",
    "@react-navigation/native": "^6.1.7",
    "@react-navigation/stack": "^6.3.17",
    "axios": "^1.4.0",
    "expo": "~48.0.18",
    "expo-status-bar": "~1.4.4",
    "react": "18.2.0",
    "react-native": "0.71.8",
    "react-native-gesture-handler": "~2.9.0",
    "react-native-paper": "^5.9.1",
    "react-native-reanimated": "~2.14.4",
    "react-native-safe-area-context": "4.5.0"
  },
  "devDependencies": {
    "@babel/core": "^7.20.0"
  },
  "private": true
}
```

E agora no seu terminal digite "**npm start**" para instalar as dependências que adicionamos.

Em seguida abra o seu arquivo "**babel.config**"

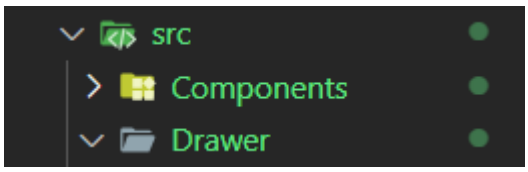
babel.config

```
module.exports = function (api) {
  api.cache(true);
  return {
    presets: ["babel-preset-expo"],
    plugins: ["react-native-reanimated/plugin"],
```

```
 };
};
```

Em seguida no seu terminal digite “npx react-native start --reset-cache” para limpar o cache e percerber a mudança do arquivo “babel.config”

Agora você pode iniciar seu emulador no android studio. E vamos começar a criar nosso app. Primeiro crie a pasta “src” e dentro dela a pasta “Components”. Também dentro de src crie a pasta Drawer. Deverá ficar assim:



Agora dentro da pasta Components crie o arquivo Dashboard.js

Dashboard.js

```
import React from "react";
import { ScrollView, StyleSheet, Text, View } from "react-native";
import { Avatar, Button, Card } from "react-native-paper";

const LeftContent = (props) => <Avatar.Icon {...props} icon="folder" />;

export default function Dashboard() {
 return (
  <ScrollView style={styles.container}>
   <Card style={styles.cardBox}>
    <Card.Title
     title="Card Title"
     subtitle="Card Subtitle"
     left={LeftContent}
    />
    <Card.Content>
     <Text variant="titleLarge">Card title</Text>
```

```
    <Text variant="bodyMedium">Card content</Text>
  </Card.Content>
  <Card.Cover
    source={{ uri: "https://picsum.photos/700" }}
    style={styles.img}
  />
  <Card.Actions>
    <Button>Cancel</Button>
    <Button>Ok</Button>
  </Card.Actions>
</Card>

<Card style={styles.cardBox}>
  <Card.Title
    title="Card Title"
    subtitle="Card Subtitle"
    left={LeftContent}
  />
  <Card.Content>
    <Text variant="titleLarge">Card title</Text>
    <Text variant="bodyMedium">Card content</Text>
  </Card.Content>
  <Card.Cover
    source={{ uri: "https://picsum.photos/700" }}
    style={styles.img}
  />
  <Card.Actions>
    <Button>Cancel</Button>
```

```
    <Button>Ok</Button>
  </Card.Actions>
</Card>

<Card style={styles.cardBox}>
  <Card.Title
    title="Card Title"
    subtitle="Card Subtitle"
    left={LeftContent}
  />
  <Card.Content>
    <Text variant="titleLarge">Card title</Text>
    <Text variant="bodyMedium">Card content</Text>
  </Card.Content>
  <Card.Cover
    source={{ uri: "https://picsum.photos/700" }}
    style={styles.img}
  />
  <Card.Actions>
    <Button>Cancel</Button>
    <Button>Ok</Button>
  </Card.Actions>
</Card>

<Card style={styles.cardBox}>
  <Card.Title
    title="Card Title"
    subtitle="Card Subtitle"
```

```
        left={LeftContent}
      />
      <Card.Content>
        <Text variant="titleLarge">Card title</Text>
        <Text variant="bodyMedium">Card content</Text>
      </Card.Content>
      <Card.Cover
        source={{ uri: "https://picsum.photos/700" }}
        style={styles.img}
      />
      <Card.Actions>
        <Button>Cancel</Button>
        <Button>Ok</Button>
      </Card.Actions>
    </Card>
  </ScrollView>
 );
}

const styles = StyleSheet.create({
 cardBox: {
  margin: 10,
 },
 img: {
  padding: 10,
 },
});
```

Agora crie o arquivo Help.js

Help.js

```
import React, { useState } from "react";
import { StyleSheet, Text, View } from "react-native";
import {
  Modal,
  Portal,
  Button,
  PaperProvider,
  TextInput,
} from "react-native-paper";
export default function Help() {
  const [visible, setVisible] = React.useState(false);
  const [note, setNote] = useState("");
  const [notes, setNotes] = useState([]);

  const showModal = () => setVisible(true);
  const hideModal = () => setVisible(false);
  const containerStyle = { backgroundColor: "white", padding: 20, margin: 10 };

  const handleSubmit = () => {
    if (note.trim() !== "") {
      const noteId = Date.now();
      setNotes([...notes, { id: noteId, content: note }]);
      setNote("");
      hideModal();
    }
```

```
};

const handleDelete = (noteId) => {
 setNotes(notes.filter((note) => note.id !== noteId));
};
return (
 <PaperProvider>
  <Portal>
   <Modal
    visible={visible}
    onDismiss={hideModal}
    contentContainerStyle={containerStyle}
   >
    <Text>Add Your Notes</Text>
    <TextInput
     value={note}
     onChangeText={setNote}
     placeholder="Enter Your Notes..."
     style={styles.textInpput}
    />
    <View style={{ width: 150 }}>
     <Button mode="contained" onPress={handleSubmit}>
      Add Node
     </Button>
    </View>
   </Modal>
  </Portal>
```

```
    <View style={styles.container}>
     <Button
      style={{ marginTop: 30, backgroundColor: "#f0f0f0" }}
      onPress={showModal}
     >
      ADD Notes
     </Button>
     {notes.map((note, index) => (
      <View key={index} style={styles.noteContainer}>
       <Text
        style={{
         borderLeftColor: "blue",
         borderLeftWidth: 2,
         color: "black",
         padding: 10,
         borderRadius: 10,
        }}
       >
        {note.content}
       </Text>

       <Button onPress={() => handleDelete(note.id)}>Delete</Button>
      </View>
     ))}
    </View>
   </PaperProvider>
  );
}
```

```
const styles = StyleSheet.create({
 container: {
  flex: 1,
  backgroundColor: "#fff",
 },
 textInpput: {
  height: 40,
  borderColor: "grey",
  borderWidth: 1,
  paddingHorizontal: 10,
  marginVertical: 10,
 },
 noteContainer: {
  flexDirection: "row",
  alignItems: "center",
  justifyContent: "space-between",
  margin: 10,
  padding: 10,
 },
});
```

Em seguida crie o arquivo Profile.js

Profile.js

```
import React, { useEffect, useState } from "react";
import { StyleSheet, Text, View, FlatList } from "react-native";
import axios from "axios";
export default function Profile() {
```

```
const [users, setUsers] = useState([]);


useEffect(() => {
  axios
    .get("https://jsonplaceholder.typicode.com/users")
    .then((response) => setUsers(response.data))
    .catch((err) => console.log(err));
}, []);


const resderUserCard = ({ item }) => {
  return (
    <View style={styles.card}>
      <Text style={styles.title}>{item.name}</Text>
      <Text style={styles.email}>{item.email}</Text>
      <Text style={styles.username}>{item.username}</Text>
      <Text style={styles.website}>{item.website}</Text>
    </View>
  );
};
return (
  <View style={styles.container}>
    <FlatList
      data={users}
      keyExtractor={(item) => item.id.toString()}
      renderItem={resderUserCard}
    />
  </View>
);
```

```
}

const styles = StyleSheet.create({
 container: {
  flex: 1,
  backgroundColor: "#f0f0f0",
  padding: 30,
  paddingHorizontal: 10,
 },
 card: {
  backgroundColor: "#fff",
  borderRadius: 8,
  padding: 15,
  marginBottom: 10,
 },
 title: {
  fontSize: 18,
  fontWeight: "bold",
  marginBottom: 5,
 },
 email: {
  color: "#666",
  marginBottom: 5,
 },
 username: {
  fontStyle: "italic",
  marginBottom: 5,
 },
```

```
  website: {
    color: "blue",
  },
});
```

E por último vamos criar o arquivo Profit.js

Profit.js

```
import React, { useState } from "react";
import { StyleSheet, Text, View, SafeAreaView, Button } from "react-native";
import DateTimePicker from "@react-native-community/datetimepicker";
export default function Profit() {
  const [date, setDate] = useState(new Date(1598051730000));
  const [mode, setMode] = useState("date");
  const [show, setShow] = useState(false);

  const onChange = (event, selectedDate) => {
    const currentDate = selectedDate;
    setShow(false);
    setDate(currentDate);
  };

  const showMode = (currentMode) => {
    setShow(true);
    setMode(currentMode);
  };

  const showDatepicker = () => {
    showMode("date");
```

```
};

const showTimepicker = () => {
 showMode("time");
};
return (
 <View style={styles.container}>
  <SafeAreaView>
   <View>
    <Button onPress={showDatepicker} title="Show date picker!" />
   </View>
   <View style={{ marginTop: 30 }}>
    <Button onPress={showTimepicker} title="Show time picker!" />
   </View>

   <Text
    style={{
     marginTop: 20,
     backgroundColor: "blue",
     padding: 20,
     color: "white",
     borderRadius: 10,
    }}
   >
    selected: {date.toLocaleString()}
   </Text>
   {show && (
    <DateTimePicker
```

```
        testID="dateTimePicker"
        value={date}
        mode={mode}
        is24Hour={true}
        onChange={onChange}
       />
      )}
    </SafeAreaView>
  </View>
 );
}

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

Com isso temos as nossas telas do nosso menu prontas. Agora vamos criar o nosso menu para poder chamar elas. Para isso, na pasta Drawer crie o arquivo MainDrawer.js. Ele irá chamar as nossas páginas que acabamos de criar.

MainDrawer.js

```
import React from "react";
import { createDrawerNavigator } from "@react-navigation/drawer";
import { createStackNavigator } from "@react-navigation/stack";
import { NavigationContainer } from "@react-navigation/native";
```

```
import Dashboard from "../Components/Dashboard";
import Profile from "../Components/Profile";
import Help from "../Components/Help";
import Profit from "../Components/Profit";
import Ionicons from "@expo/vector-icons/Ionicons";
import { View } from "react-native";
const Drawer = createDrawerNavigator();
const Stack = createStackNavigator();

const DashboardIcon = ({ focused, color, size }) => (
  <Ionicons name="md-speedometer" size={size} color={color} />
);
const ProfileIcon = ({ focused, color, size }) => (
  <Ionicons name="md-person" size={size} color={color} />
);
const HelpdIcon = ({ focused, color, size }) => (
  <Ionicons name="md-refresh-circle" size={size} color={color} />
);
const ProfitIcon = ({ focused, color, size }) => (
  <Ionicons name="md-cart" size={size} color={color} />
);

const MainDrawer = () => {
  return (
    <NavigationContainer>
      <Stack.Navigator screenOptions={{ headerShown: false }}>
        <Stack.Screen name="MainDashboard">
          {() => (
```

```
<Drawer.Navigator
 screenOptions={{
  drawerStyle: {
   backgroundColor: "#6b3fa0", //mudar a cor
   width: 230, //mudar a largura do sidebar
  },
 }}
>
 <Drawer.Screen
  name="Dashboard"
  component={Dashboard}
  options={{ drawerIcon: DashboardIcon }}
 />
 <Drawer.Screen
  name="Profile"
  component={Profile}
  options={{ drawerIcon: ProfileIcon }}
 />
 <Drawer.Screen
  name="Notes"
  component={Help}
  options={{ drawerIcon: HelpdIcon }}
 />
 <Drawer.Screen
  name="Profit"
  component={Profit}
  options={{ drawerIcon: ProfitIcon }}
 />
```

```
        </Drawer.Navigator>
       )}
      </Stack.Screen>
    </Stack.Navigator>
  </NavigationContainer>
 );
};

export default MainDrawer;
```

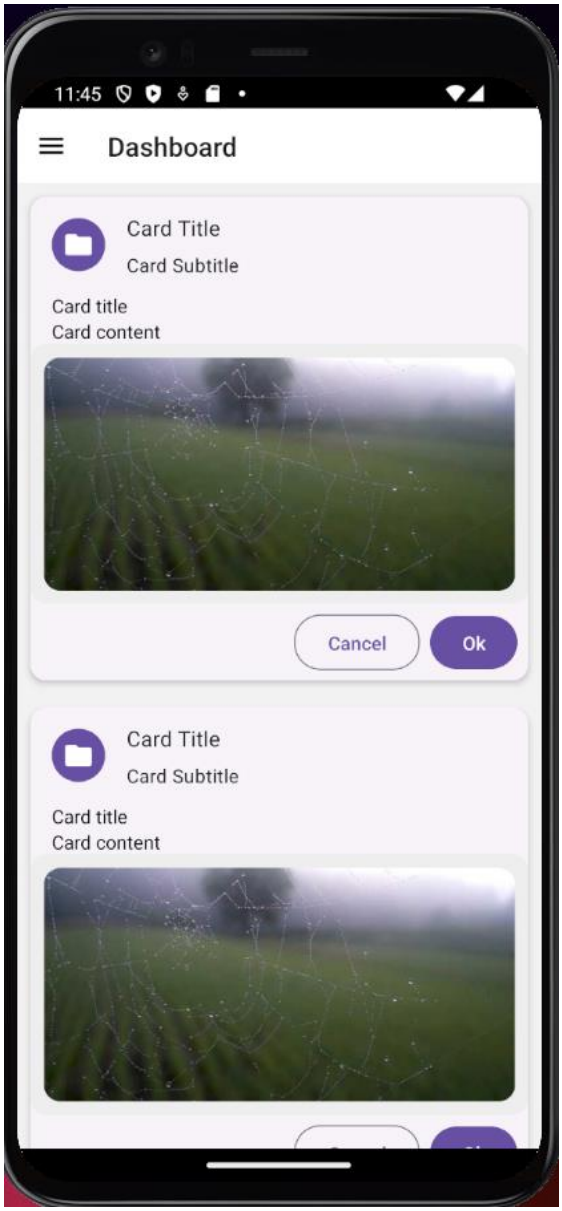
Com isso, agora vamos alterar o nosso App.js para que chame o MainDrawer.js

App.js

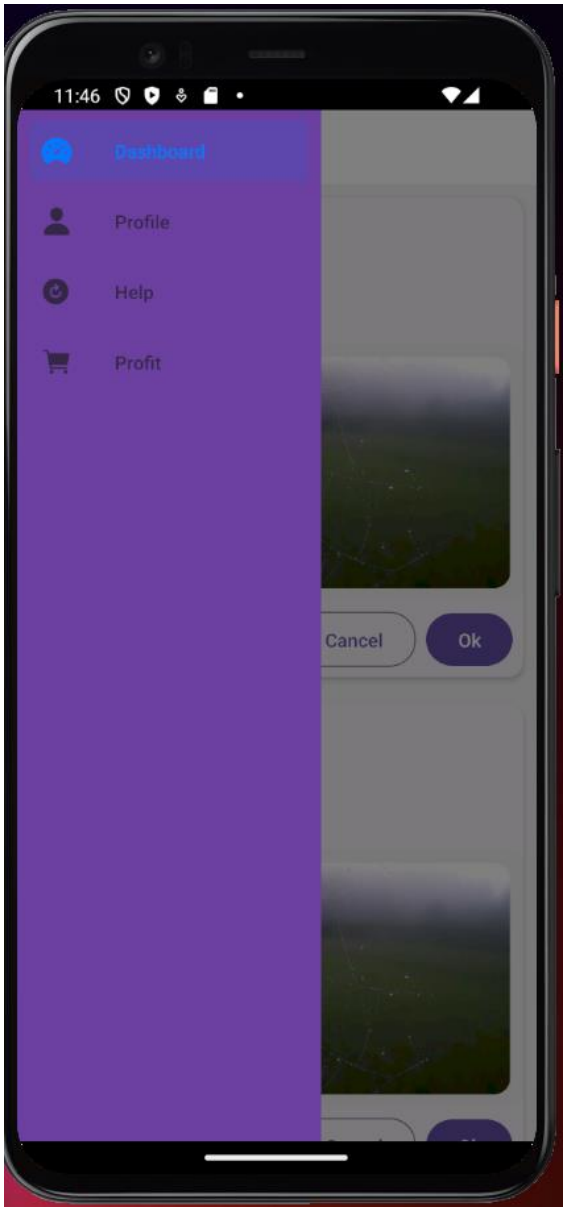
```
import MainDrawer from "./src/Drawer/MainDrawer";

export default function App() {
 return (
  <>
   <MainDrawer />
  </>
 );
}
```

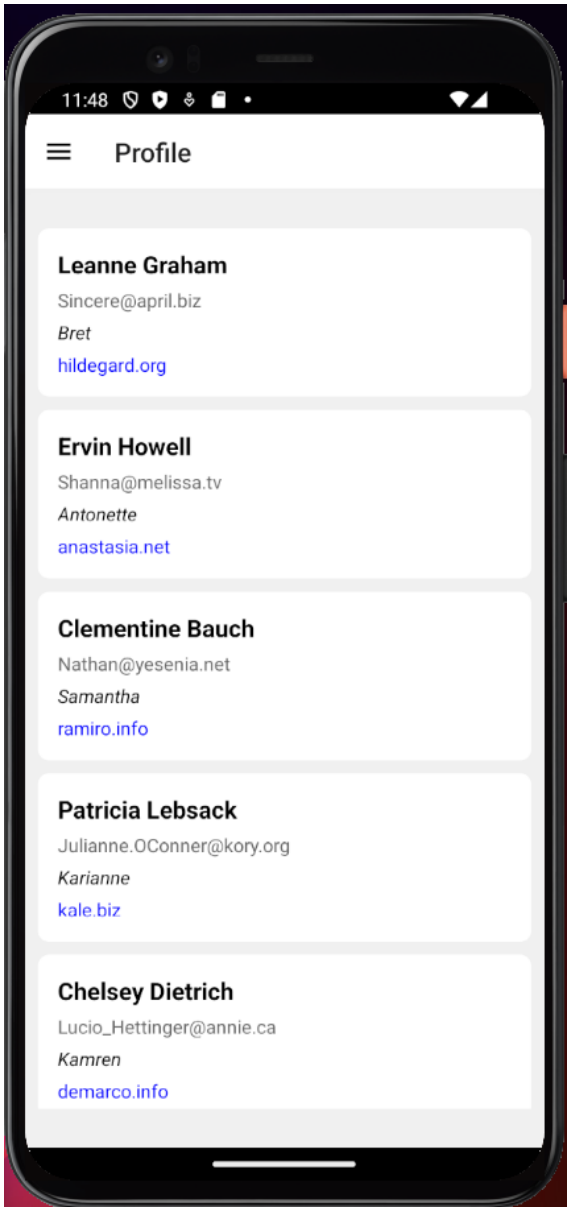
Depois disso, você deverá estar vendo as telas do seu app já funcionando.



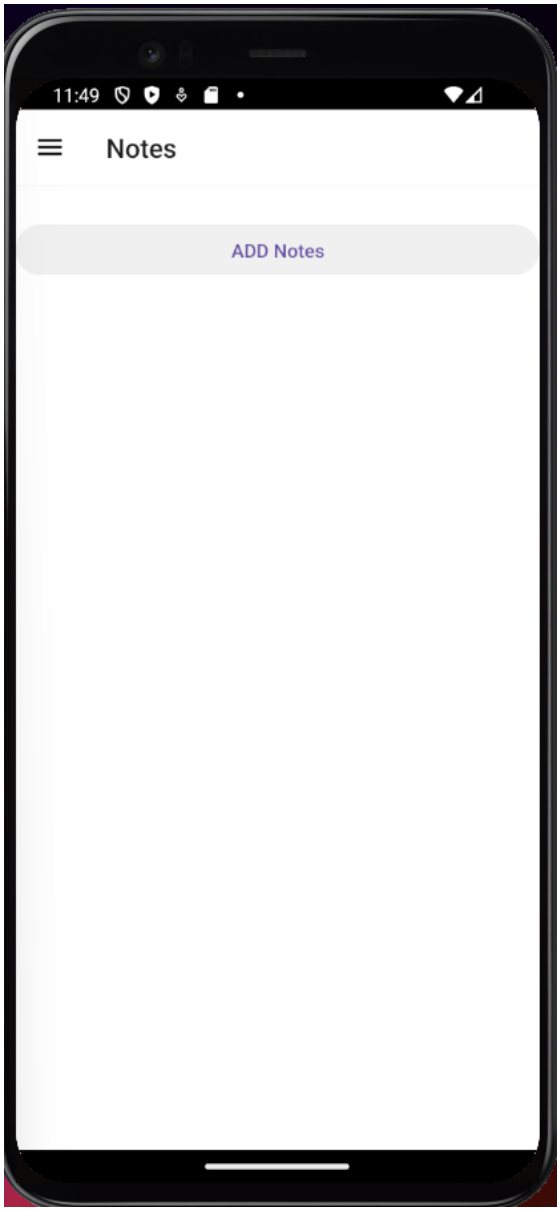
Ao clicar no menu superior esquerdo deverá mostrar as páginas disponíveis.



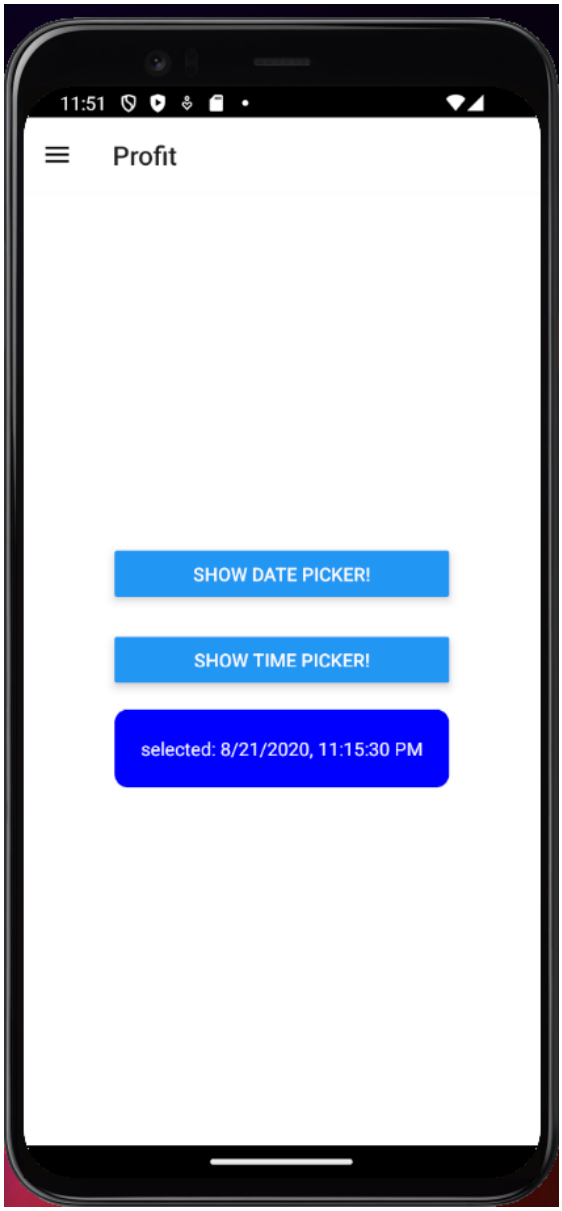
Página de Perfil:



Página de notas:



Pagina de profit:


11:51
Profit
SHOW DATE PICKER!
SHOW TIME PICKER!
selected: 8/21/2020, 11:15:30 PM