Aula 69
Leitor de QR code

Objetivos.

- Obter permissões para usar a câmera no aplicativo.
- Usar o componente BarCodeScanner (leitor de código de barras) para digitalizar um QR code.
- Exibir dados do QR code dentro de um componente Text.

Vamos criar um botão (usando TouchableOpacity) que acionará o leitor de QR code e exibir o resultado digitalizado como texto.

```
    render(){
        return(
            <View style={styles.container}>

                <TouchableOpacity style={styles.btqr}>
                <Text style={styles.texto}>QR Code</Text>
                </TouchableOpacity>

            </View>
        );
    }
}

const styles= StyleSheet.create({
    container:{
        flex:1,
        justifyContent:"center",
        alignItems:"center",
        backgroundColor:"#5653D4",
    },
    texto:{
        color:"#ffffff",
        fontSize:30,
    },
    btqr:{
        width:"43%",
        height:55,
        justifyContent:"center",
        alignItems:"center",
        backgroundColor:"#f48d20",
        borderRadius:15,
    },
})
```

Usaremos um pacote expo (biblioteca) que nos ajudará a construir o scanner de QR code em nosso aplicativo.

expo install expo-barcode-scanner
expo install expo-permissions

Vamos importar o componente e as permissões do leitor de código de barras.

```
import React, { Component } from "react";
import { View,Text,StyleSheet,TouchableOpacity } from "react-native";
import { BarCodeScanner } from "expo-barcode-scanner";
import * as Permissions from "expo-permissions"
export default class RetiradaScreen extends Component{
```

Vamos definir quatro estados em nosso aplicativo:
● domState : “ (estado do modo) => Isso dirá se o aplicativo está no modo digitalizar ou no modo digitalizado.
● hasCameraPermissions: null (câmera tem permissões) => Isso dirá se o usuário concedeu permissão de câmera para o aplicativo.
● scanned: false (digitalizado) => Isso dirá se a digitalização foi concluída ou não.
● scannedData: " (dado digitalizado) => Isso manterá os dados digitalizados que obtemos após digitalizar.

```
export default class RetiradaScreen extends Component{

  constructor(props){
      super(props);
      this.state={
          domState:"normal",
          hasCameraPermissions:null,
          scanned:false,
          scannedData:""
      }
  }
```

Vamos escrever uma função **getCameraPermission** para solicitar a permissão da câmera.

```
getCameraPermissions=async domState=>{
    const {status}=await Permissions.askAsync(Permissions.CAMERA);

    this.setState({
       hasCameraPermissions:status=== "grated",
        domState:domState,
        scanned:false
    });
};
```

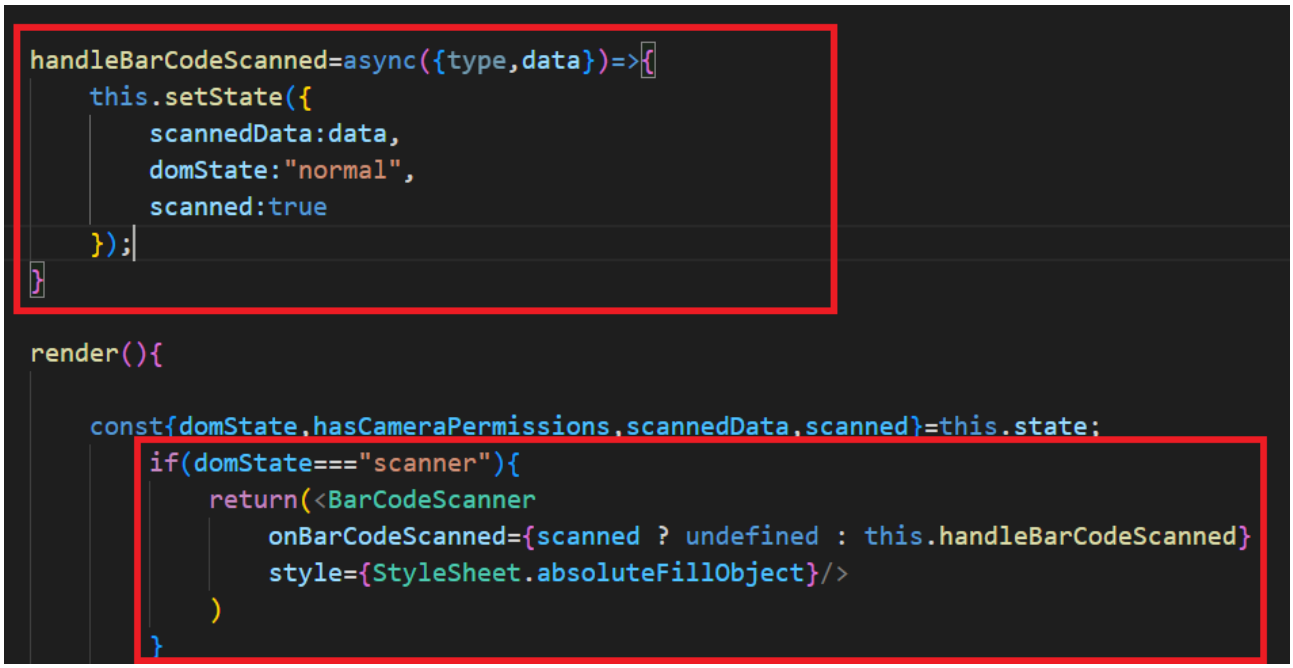
Vamos chamar a função getCameraPermissions quando o botão TouchableOpacity for pressionado usando sua props onPress.

```
<TouchableOpacity style={styles.btqr}
onPress={()=>this.getCameraPermissions("scanner")}
>
<Text style={styles.texto}>QR Code</Text>
</TouchableOpacity>
```

Precisamos dizer ao nosso aplicativo o que fazer se hasCameraPermissions for 'null' (nulo) ou 'false' (falso).

```
render(){
    const{domState,hasCameraPermissions,scannedData,scanned}=this.state;
    return(
        <View style={styles.container}>
            <Text style={styles.textms}>
                {hasCameraPermissions? scannedData:"Solicitar Permissão para a câmera"}
            </Text>
            <TouchableOpacity style={styles.btqr}
            onPress={()=>this.getCameraPermissions("scanner")}
            >
            <Text style={styles.texto}>QR Code</Text>
            </TouchableOpacity>
```

Agora, queremos exibir nosso BarCodeScanner (leitor de código de barras) quando nosso botão Scan (digitalizar) for clicado.



```
handleBarCodeScanned=async({type,data})=>{
    this.setState({
        scannedData:data,
        domState:"normal",
        scanned:true
    });
}

render(){

    const{domState,hasCameraPermissions,scannedData,scanned}=this.state;
        if(domState==="scanner"){
            return(<BarCodeScanner
                onBarCodeScanned={scanned ? undefined : this.handleBarCodeScanned}
                style={StyleSheet.absoluteFillObject}/>
            )
        }
```