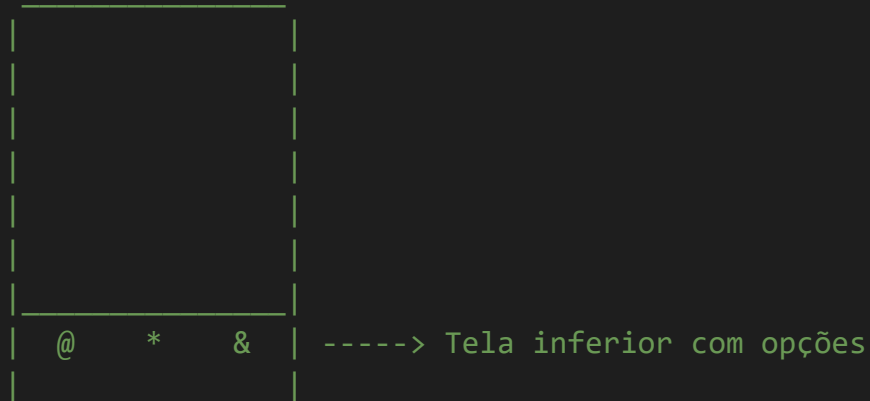


```
import React from "react";
import { createBottomTabNavigator } from "@react-navigation/bottom-tabs";
import TelaA from "../views/TelaA";
import TelaB from "../views/TelaB";
import TelaC from "../views/TelaC";
```

/* TAB NAVIGATION:

Uma das navegações que o React Navigation usa é a navegação em Tab ou Bottom Tab, esse tipo de navegação consiste em fazer com que o aplicativo possua uma barra inferior com opções, que ao serem clicadas naveguem entre diferentes tela.



Para usar o Tab precisamos seguir um passo a passo importante:

- 1º Criação de um Componente que terá um wrapper onde todas as telas serão comportadas;
- 2º Dentro deste componente, teremos que importar o método "createBottomTabNavigator" da biblioteca "bottom-tabs", conforme podemos ver nas linhas de cima. Para facilitar temos o costume de atribuir esse método a uma variável intitulada Tab;
- 3º Criar um wrapper de navegação usando o componente "Navigator" a partir da variável Tab, é esse

wrapper que irá comportar as telas;

4º Todos os Componentes deverão ser passados por através de um componente "Screen", esse componente também é nativo da variável "Tab", os componentes podem ser passados tanto por através da propriedade "component", embutida no próprio componente Screen, quanto por através de wrapper, onde o componente Screen será o wrapper do componente passado (Nesse caso não poderemos ter a propriedade "component" declarada sobre o componente Screen);

5º Todo esse componente, com suas devidas telas, deverá ser exportado para um componente externo que irá renderizar a nossa tela de navegação Tab. O componente externo deverá importar o componente "NavigationContainer" a partir da biblioteca "@react-navigation/native". Esse componente que irá comportar o nosso componente Tab.

Veja um exemplo detalhado abaixo...

```
*/  
  
//Veja que atribuímos o método "createBottomTabNavigator" à constante "Tab"  
const Tab = createBottomTabNavigator()  
  
export default props => {  
  
  return (  
    /* Usamos um wrapper Navigator, esse wrapper pode receber algumas propriedades que nos ajudam a  
       renderizar as opções, temos muitas outras que podem nos ajudar com estilização e imagens de  
       ícones...  
    */  
    <Tab.Navigator  
      screenOptions={{  
        tabBarActiveTintColor: 'tomato',  
        tabBarInactiveTintColor: 'gray',
```

```

    }}
    //Note que escolhemos a tela B como a tela inicial...
    initialRouteName="TelaB"
  >
    { /* Não há necessidade de passar para onde a tela irá, visto que todas as telas ficam expostas na tab
inferior... */ }
    <Tab.Screen name="TelaA" component={TelaA} />
    <Tab.Screen name="TelaB" component={TelaB} />
    <Tab.Screen name="TelaC" component={TelaC} />
  </Tab.Navigator>
)
}

```

ARQUIVO TELA A...

```

import React from "react";
import TextoCentral from "../componentes/TextoCentral";

export default props => {
  return (
    <TextoCentral children={'TELA A'} corFundo="red" />
  )
}

```

ARQUIVO INDEX.JS...

```

import React from "react";
import { SafeAreaView, StyleSheet } from "react-native";
//Aqui temos a importação do componente NavigationContainer...
import { NavigationContainer } from "@react-navigation/native";
import Tab from "../Tab";

```

```
export default props => {  
  return (  
    <SafeAreaView style={styles.safeArea}>  
      <NavigationContainer>  
        <Tab />  
      </NavigationContainer>  
    </SafeAreaView>  
  )  
}  
  
const styles = StyleSheet.create({  
  safeArea: {  
    flex: 1,  
  },  
})
```

RESULTADO NO APP...

