	<p align="center"><b>Centro Federal de Educação Tecnológica de Minas Gerais</b>  <b>Campus VIII – Varginha</b>  <b>Curso Técnico em Informática</b></p>	
<p><i>Disciplina</i>  <b>Lab. Aplicações Móveis</b></p>	<p align="center"><b>React Navigation</b>  <b>Biblioteca de Navegação</b></p>	<p><i>Professor</i>  <b>Lázaro Eduardo da Silva</b></p>

Nessa aula vamos trabalhar com uma biblioteca da comunidade chamada React Navigation. A documentação desta ferramenta está no link abaixo:

<https://reactnavigation.org/>

Vou mostrar a implementação em um projeto novo, mas você pode acrescentar as bibliotecas no seu projeto e encaminhar o aplicativo para a navegação. Para iniciar o primeiro pacote que precisa ser instalado é o `@react-navigation/native` para isso, execute o comando abaixo:

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

Após finalizada a instalação, instale o pacote `react-native-screens`, responsável por informar a biblioteca os componentes que são telas e o pacote `react-native-safe-area-context`, responsável por calcular a área visível do celular que está executando o aplicativo e posicionar o menu de navegação. Para isso, execute o comando abaixo.

```
npx expo install react-native-screens react-native-safe-area-context
```

Com isso, temos as instalações iniciais da biblioteca, o que já permite utilizar o container de navegação. Vamos criar uma pasta chamada `navigation`, dentro do `src`, criar um arquivo `index.tsx` e colocar o seguinte código.

```
src > navigation > index.tsx > ...
1  import * as React from 'react';
2  import { NavigationContainer } from '@react-navigation/native';
3
4  export default function Navigation() {
5    return (
6      <NavigationContainer>
7
8      </NavigationContainer>
9    );
10 }
```

Dentro do container, carregaremos o componente específico da navegação que quisermos criar.

Agora precisamos decidir quais formas de navegação iremos trabalhar. Vamos começar pela navegação Stack. Ela empilha as

telas e desempilha ao voltar as telas carregadas. O local específico da documentação dela está no link abaixo:

<https://reactnavigation.org/docs/stack-navigator>

Primeiro vamos instalar os pacotes necessários. O primeiro é o específico desta navigator.

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

Para utilizá-la, ainda é necessário instalar o pacote `react-native-gesture-handler`, que permite mover as telas empilhadas pelo arrastar do dedo na tela.

```
npx expo install react-native-gesture-handler
```

O próximo pacote é necessário para manter a compatibilidade das ações na tela com o SO IOS.

```
npx expo install @react-native-masked-view/masked-view
```

Com os pacotes instalados, podemos criar o arquivo MenuStack que iremos colocar a navegação entre as telas da aplicação. O código para navegação entre duas telas chamadas Slide1 e Slide2 é:

```
src > navigation > MenuStack.navigation.tsx > ...
1  import { StackNavigationProp, createStackNavigator } from '@react-navigation/stack';
2  import { ScreenSlide1, ScreenSlide2 } from '../screens';
3
4  type MenuStackParam = {
5    Slide1: undefined
6    Slide2: undefined
7  }
8  type MenuScreenNavigation = StackNavigationProp<MenuStackParam, "Slide1">
9  export type MenuStackTypes = {
10    navigation: MenuScreenNavigation
11  }
12
13  export function MenuStack() {
14    const Stack = createStackNavigator<MenuStackParam>();
15    return (
16      <Stack.Navigator screenOptions={{
17        animationEnabled: true,
18        gestureEnabled: true
19      }}>
20        <Stack.Screen name="Slide1" component={ScreenSlide1} />
21        <Stack.Screen name="Slide2" component={ScreenSlide2} />
22      </Stack.Navigator>
23    );
24  }
```

Neste arquivo foi criado o tipo MenuStackParam que define o nome das telas para a navegação. O tipo MenuScreenNavigation guarda a tipagem criada, aplicada nas propriedades da StackNavigationProp. O tipo MenuStackTypes foi exportado, porque ele irá tipar os parâmetros das telas para permitir o uso das funções que realizam a navegação.


Dentro da function component MenuStack criamos a constante Stack que será o componente responsável por

```
src > screens > Slide1 > index.tsx > ...
1  import { Text, TouchableOpacity, View } from "react-native";
2  import { styles } from "../styles"
3  import { MenuStackTypes } from "../../navigation/MenuStack.navigation";
4
5  export function Slide1({ navigation }: MenuStackTypes) {
6    return (
7      <View style={styles.container}>
8        <Text style={styles.texto}>Slide 1</Text>
9        <TouchableOpacity style={styles.botao}
10          onPress={() => navigation.push("Slide2")}
11        >
12          <Text>Slide 2</Text>
13        </TouchableOpacity>
14      </View>
15    )
16  }
```

definir a navegação e as telas.

O código das telas Slides1 e Slides2 são simples. Vamos fazer com que através de um botão, podemos navegar do Slide1 para o Slide 2 e vice-versa.

Para isso, basta trocar o nome da tela dentro da chamada da função push.

	<p align="center"><b>Centro Federal de Educação Tecnológica de Minas Gerais</b>  <b>Campus VIII – Varginha</b>  <b>Curso Técnico em Informática</b></p>	
<p><i>Disciplina</i>  <b>Lab. Aplicações Móveis</b></p>	<p align="center"><b>React Navigation</b>  <b>Biblioteca de Navegação</b></p>	<p><i>Professor</i>  <b>Lázaro Eduardo da Silva</b></p>

Criado o arquivo MenuStack, vamos chamá-lo no arquivo index.tsx do navigation.

```
src > navigation > index.tsx > ...
1  import * as React from 'react';
2  import { NavigationContainer } from '@react-navigation/native';
3  import { MenuStack } from './MenuStack.navigation';
4
5  export default function Navigation() {
6    return (
7      <NavigationContainer>
8        <MenuStack />
9      </NavigationContainer>
10   );
11 }
```

Por fim, para que a navegação seja carregada, vamos realizar a chamada da navegação no arquivo App.tsx

```
App.tsx > ...
You, há 31 minutos | 1 author (You)
1  import 'react-native-gesture-handler';
2  import Navigation from './src/navigation';
3
4  export default function App() {
5    return (
6      <Navigation />
7    );
8  }
```

Vamos aproveitar a atividade para construir a navegação Bottom Tabs também. A documentação específica dela fica no link abaixo:

<https://reactnavigation.org/docs/bottom-tab-navigator>

Para utilizá-la, inicialmente, vamos instalar o pacote dela executando o comando abaixo:

```
npm install @react-navigation/bottom-tabs
```

Feito isso, podemos criar o arquivo MenuBottomTab. Este menu é localizado no rodapé do aplicativo e os itens clicáveis são ícones. Para colocarmos estes ícones, podemos utilizar um conjunto de itens da comunidade disponibilizado no pacote @expo/vector-icons. A documentação dele fica no link abaixo:

<https://docs.expo.dev/guides/icons/#expovector-icons>

Para começarmos a utilizá-lo, inicialmente precisamos instalar o pacote executando o comando.

```
npx expo install @expo/vector-icons
```

Feito isso, podemos consultar o ícone que desejamos colocar no link abaixo:


<https://icons.expo.fyi/>

Escolhidos os ícones, podemos criar nosso menu Bottom Tab com o seguinte conteúdo:

```
src > navigation > MenuBottomTab.navigarion.tsx > ...
1  import { createBottomTabNavigator, BottomTabNavigationProp } from '@react-navigation/bottom-tabs';
2  import { ScreenSlide1, ScreenSlide2 } from '../screens';
3  import { Entypo, Feather } from '@expo/vector-icons';
4
5  type MenuTabParam = {
6    Slide1: undefined
7    Slide2: undefined
8  }
9  type MenuScreenNavigation = BottomTabNavigationProp<MenuTabParam, "Slide1">
10 export type MenuTabTypes = {
11   navigation: MenuScreenNavigation
12 }
13
14 export function MenuTabs() {
15   const Tab = createBottomTabNavigator<MenuTabParam>();
16   return (
17     <Tab.Navigator>
18       <Tab.Screen name="Slide1" component={ScreenSlide1}
19         options={{
20           tabBarIcon: () => (
21             <Entypo name="slideshare" size={24} color="black" />
22           )
23         }}
24       />
25       <Tab.Screen name="Slide2" component={ScreenSlide2}
26         options={{
27           tabBarIcon: () => (
28             <Feather name="sliders" size={24} color="black" />
29           )
30         }}
31       />
32     </Tab.Navigator>
33   );
34 }
```

Para verificarmos como ficou a nossa navegação, será necessário incluir o componente dela no index.tsx do navigation.

```
src > navigation > index.tsx > ...
1  import * as React from 'react';
2  import { NavigationContainer } from '@react-navigation/native';
3  import { MenuStack } from './MenuStack.navigation';
4  import { MenuTabs } from './MenuBottomTab.navigarion'
5
6  export default function Navigation() {
7    return (
8      <NavigationContainer>
9        <MenuTabs />
10      </NavigationContainer>
11    );
12 }
```

	<p align="center"><b>Centro Federal de Educação Tecnológica de Minas Gerais</b>  <b>Campus VIII – Varginha</b>  <b>Curso Técnico em Informática</b></p>	
<p><i>Disciplina</i>  <b>Lab. Aplicações Móveis</b></p>	<p align="center"><b>React Navigation</b>  <b>Biblioteca de Navegação</b></p>	<p><i>Professor</i>  <b>Lázaro Eduardo da Silva</b></p>

Por fim, a última navegação que estudaremos é a Drawer. Ela monta o menu lateral conhecido como menu sanduíche. A documentação específica dela está no link abaixo:

<https://reactnavigation.org/docs/drawer-navigator>

Para começarmos a trabalhar com ela, precisamos instalar o seu pacote principal executando o comando abaixo:

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

Posteriormente, precisamos instalar os pacotes react-native-gesture-handler e o react-native-reanimated. Observe que já instalamos o primeiro na Stack, mas vamos seguir o passo a passo completo. Caso já tenha instalado, não precisa instalar novamente, porém, caso fique na dúvida é melhor repetir a instalação.

```
npx expo install react-native-gesture-handler react-native-reanimated
```

Para utilizarmos o segundo pacote instalado, precisamos habilitar ele como plugin no arquivo babel.config.js conforme imagem abaixo.



```

babel.config.js > ...
You, há 8 minutos | 1 author (You)
1  module.exports = function (api) {
2    api.cache(true);
3    return {
4      presets: ['babel-preset-expo'],
5      plugins: ['react-native-reanimated/plugin']
6    };
7  };

```

Após realizar esta configuração você terá que para a execução do seu projeto, limpar as telas abertas no seu celular para executá-lo novamente.


Feito isso, podemos criar o arquivo MenuDrawer para colocar o código da implementação desta navegação.




```

src > navigation > MenuDrawer.navigation.tsx > ...
1  import { DrawerNavigationProp, createDrawerNavigator } from '@react-navigation/drawer';
2  import { ScreenSlide1, ScreenSlide2 } from '../screens';
3
4  type MenuDrawerParam = {
5    Slide1: undefined
6    Slide2: undefined
7  }
8  type MenuScreenNavigation = DrawerNavigationProp<MenuDrawerParam, "Slide1">
9  export type MenuDrawerTypes = {
10   navigation: MenuScreenNavigation
11 }
12
13 export function MenuDrawer() {
14   const Drawer = createDrawerNavigator<MenuDrawerParam>();
15   return (
16     <Drawer.Navigator>
17       <Drawer.Screen name="Slide1" component={ScreenSlide1} />
18       <Drawer.Screen name="Slide2" component={ScreenSlide2} />
19     </Drawer.Navigator>
20   );
21 }

```

	<p align="center"><b>Centro Federal de Educação Tecnológica de Minas Gerais</b>  <b>Campus VIII – Varginha</b>  <b>Curso Técnico em Informática</b></p>	
<p><i>Disciplina</i>  <b>Lab. Aplicações Móveis</b></p>	<p align="center"><b>React Navigation</b>  <b>Biblioteca de Navegação</b></p>	<p><i>Professor</i>  <b>Lázaro Eduardo da Silva</b></p>

Para verificarmos como ficou a nossa navegação, será necessário incluir o componente dela no index.tsx do navigation conforme imagem abaixo.

```
src > navigation >  index.tsx > ...
1   import * as React from 'react';
2   import { NavigationContainer } from '@react-navigation/native';
3   import { MenuStack } from './MenuStack.navigation';
4   import { MenuTabs } from './MenuBottomTab.navigation';
5   import { MenuDrawer } from './MenuDrawer.navigation';
6
7   export default function Navigation() {
8     return (
9       <NavigationContainer>
10      |   <MenuDrawer />
11      | </NavigationContainer>
12    );
13  }
```

Testados os 3 formatos de navegação mais utilizados nos aplicativos atuais, você pode escolher qual se adequa melhor ao seu projeto.

Bom trabalho!