

Roteiro Didático de Aula
Aula 02 – React Native
11/11/2020

Assuntos que serão abordados:

Criando um novo App “NavegApp”

Navegando por Pilha

Navegação em abas:

Referências

Criando um novo App “NavegApp”

Executar esse comando no terminal, já no diretório onde será a raiz do projeto

```
yarn global add expo-cli
```

Após a instalação executar

```
expo init NavegApp
```

Após isso escolher o “blank (typescript)” pois é um template em banco que vem com typescript configurado. Esperar ele terminar de carregar...

Então navegar para pasta do projeto NavegApp, executar o comando `code .` para abrir o vsCode

Pronto, seu app já está iniciado, agora temos que começar a editar!

Navegando por Pilha

Para navegar por pilha, pages ou tabs devemos instalar o React-Navigation:

```
yarn add @react-navigation/native
```

E como estamos trabalhando com o expo, instalar mais essas:

```
expo install react-native-gesture-handler react-native-reanimated react-native-screens react-native-safe-area-context @react-native-community/masked-view
```

Vamos iniciar o servidor e começar a brincar, e então executar no terminal:

```
yarn start
```

E aguardar a aplicação carregar...

Então podemos editar o arquivo `App.tsx`

Alterar o texto para ver que está funcionando

Agora para usar a navegação em pilha(stack)

Instalar, pelo terminal, dentro do diretório do projeto:

```
yarn add react
```

```
yarn add @react-navigation/stack
```

```
yarn add @react-navigation/native
```

Reinicie o servidor, encerrando o servidor e iniciando de novo

OBS: O VSCODE AS VEZES PEDE ALGUNS PACKS DE TIPAGENS, SE NECESSARIO INSTALAR SEGUNDO O QUE ELE TA PEDINDO

E podemos deletar todo o código do arquivo app, e começarmos a escrever um novo código!

Iniciamos pelas importações:

```
import React from 'react';  
//pois toda aplicação que utiliza react precisa dessa importação no inicio.  
import { Button, View, Text } from 'react-native';  
//pois para a navegação vamos usar botões, e para usa-  
los, se faz necessario a View w o Text  
import { NavigationContainer } from '@react-navigation/native';  
//pois sempre que vamos fazer navegação em pilha, é necessário essa tag por volta, como ve  
remos a frente  
import { createStackNavigator } from '@react-navigation/stack';  
//pois é ela que controla a navegação em pilha
```

E para começar, vamos criar a Home page:

```
function HomeScreen(NavigationContainer: { navigation: { navigate: (arg0: string) => void;  
}; }) {  
  return (  
    <View style={{ flex: 1, alignItems: 'center', justifyContent: 'center' }}>  
      <Text>Home Screen</Text>  
      <Button  
        title="Ir para Detalhes"  
        onPress={() => NavigationContainer.navigation.navigate('Detalhes')}  
      />  
    </View>  
  );  
}
```

Agora vamos criar a page de Details:

```
function DetailsScreen(NavigationContainer: { navigation: { navigate: (arg0: string) => void; }; }) {  
  return (  
    <View style={{ flex: 1, alignItems: 'center', justifyContent: 'center' }}>  
      <Text>Details Screen</Text>  
      <Button  
        title="Ir para Home"  
        onPress={() => NavigationContainer.navigation.navigate('Home')}  
      />  
    </View>  
  );  
}
```

E agora vamos criar a funcionalidade da navegação:

```
const Stack = createStackNavigator();  
//Vamos criar essa constante para o manuseio da função  
  
function App() {  
  //nessa função podemos ver como deve ser estruturado as rotas de um projeto react-  
  native usando react-navigation  
  
  NÃO COPIAR OS COMENTÁRIOS ABAIXO NO CÓDIGO SE NÃO DA PAU  
  
  return (  
    <NavigationContainer>  
      //essa tag SEMPRE deve estar envolvendo as rotas na aplicação  
      <Stack.Navigator initialRouteName="Home">  
        //essa irá definir qual é a page inicial  
        <Stack.Screen name="Home" component={HomeScreen} />  
        //esses mostram quais os nomes, lembrando que, assim como no react, TUDO aqui é um compo-  
        nte, e como no react tbm, o nome em 'component' é o nome da função  
        <Stack.Screen name="Detalhes" component={DetailsScreen} />  
      </Stack.Navigator>  
    </NavigationContainer>  
  );  
}  
  
export default App;  
// exportamos no final, não esquecer disso, se não o app não abre
```

Quando compilado, veremos que a aplicação já veio com uma status bar, que já possui uma opção de voltar, porém e se quisermos fazê-la de forma manual?

Para isso que temos o "goBack()"

Usando a tela de DetailsScreen mesmo:

```
function DetailsScreen(NavigationContainer: { navigation: { navigate: (arg0: string) => void; }}) {  
  return (  
    <View style={{ flex: 1, alignItems: 'center', justifyContent: 'center' }}>  
      <Text>Details Screen</Text>  
      <Button  
        title=" Ir para Home... de novo"  
        onPress={() => NavigationContainer.navigation.navigate('Detalhes')}  
      />  
      <Button title="Ir para Home" onPress={() => NavigationContainer.navigation.navigate('Home')} />  
      <Button title="Go back" onPress={() => NavigationContainer.navigation.goBack()} />  
    </View>  
  );  
}
```

//aqui vemos o goBack funcionando, ele irá voltar para a page anterior

Navegação em abas:

Para a navegação em abas, vamos instalar:

```
yarn add @react-navigation/bottom-tabs
```

E então começar a escrever o código:

```
import React from 'react';  
//pois toda aplicação que utiliza react precisa dessa importação no início.  
  
import { View, Text } from 'react-native';  
//pois para a navegação se faz necessário a View w o Text  
  
import { NavigationContainer } from '@react-navigation/native';  
//pois sempre que vamos fazer navegação em pilha, é necessário essa tag por volta, como veremos a frente  
  
import { createBottomTabNavigator } from '@react-navigation/bottom-tabs';  
//pois é ela que controla a navegação em abas
```

Então vamos começar com o conteúdo que terá dentro das abas:

```
function HomeScreen() {  
  return (  
    <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>  
      <Text>Home!</Text>  
    </View>  
  );  
}  
  
function SettingsScreen() {  
  return (  
    <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>  
      <Text>Settings!</Text>  
    </View>  
  );  
}
```

O funcionamento é o mesmo que antes.

Agora a parte em que algo muda:

```
const Tab = createBottomTabNavigator();  
//Vamos criar essa constante para o manuseio da função  
  
export default function App() {  
  //criamos a função e já a exportamos.  
  return (  
    <NavigationContainer>  
      //essa tag SEMPRE deve estar envolvendo as rotas na aplicação  
      <Tab.Navigator  
        tabBarOptions={{style:{height:100, paddingBottom:50}}}>  
        //apenas para vermos mais fácil e ver que possível estilizar, um pouco a tab  
        //essa tab mostra que vamos usar tabs  
  
        <Tab.Screen name="Home" component={HomeScreen} />  
      //define as tabs  
      <Tab.Screen name="Settings" component={SettingsScreen} />  
    </Tab.Navigator>  
  </NavigationContainer>  
  );  
}  
//e como já exportamos no começo, não é preciso agora.
```

Menu Lateral

yarn add @react-navigation/drawer

```
import React from 'react';
import { Button, Text, View } from 'react-native';
import { createDrawerNavigator } from '@react-navigation/drawer';
import { NavigationContainer } from '@react-navigation/native';

function HomeScreen({ navigation }) {
  return (
    <View style={{ flex: 1, alignItems: 'center', justifyContent: 'center' }}>
      <Button
        onPress={() => navigation.navigate('Notifications')}
        title="Go to notifications"
      />
    </View>
  );
}

function NotificationsScreen({ navigation }) {
  return (
    <View style={{ flex: 1, alignItems: 'center', justifyContent: 'center' }}>
      <Button onPress={() => navigation.goBack()} title="Go back home" />
    </View>
  );
}

const Drawer = createDrawerNavigator();

export default function App() {
  return (
    <NavigationContainer>
      <Drawer.Navigator initialRouteName="Home">
        <Drawer.Screen name="Home" component={HomeScreen} />
        <Drawer.Screen name="Notifications" component={NotificationsScreen} />
      </Drawer.Navigator>
    </NavigationContainer>
  );
}
```

Docs typeorm:

<https://typeorm.io/#/>

Docs React-native:

<https://reactnative.dev/docs/getting-started>

Docs Expo:

<https://docs.expo.io/>

Docs React-Navigation

<https://reactnavigation.org/docs/getting-started>

Referências

Rocketseat. (s.d.). *Expo*. Fonte: <https://www.youtube.com/watch?v=ZaDpDIPr25M>

Yarn. (s.d.). Fonte: <https://yarnpkg.com/getting-started/install>