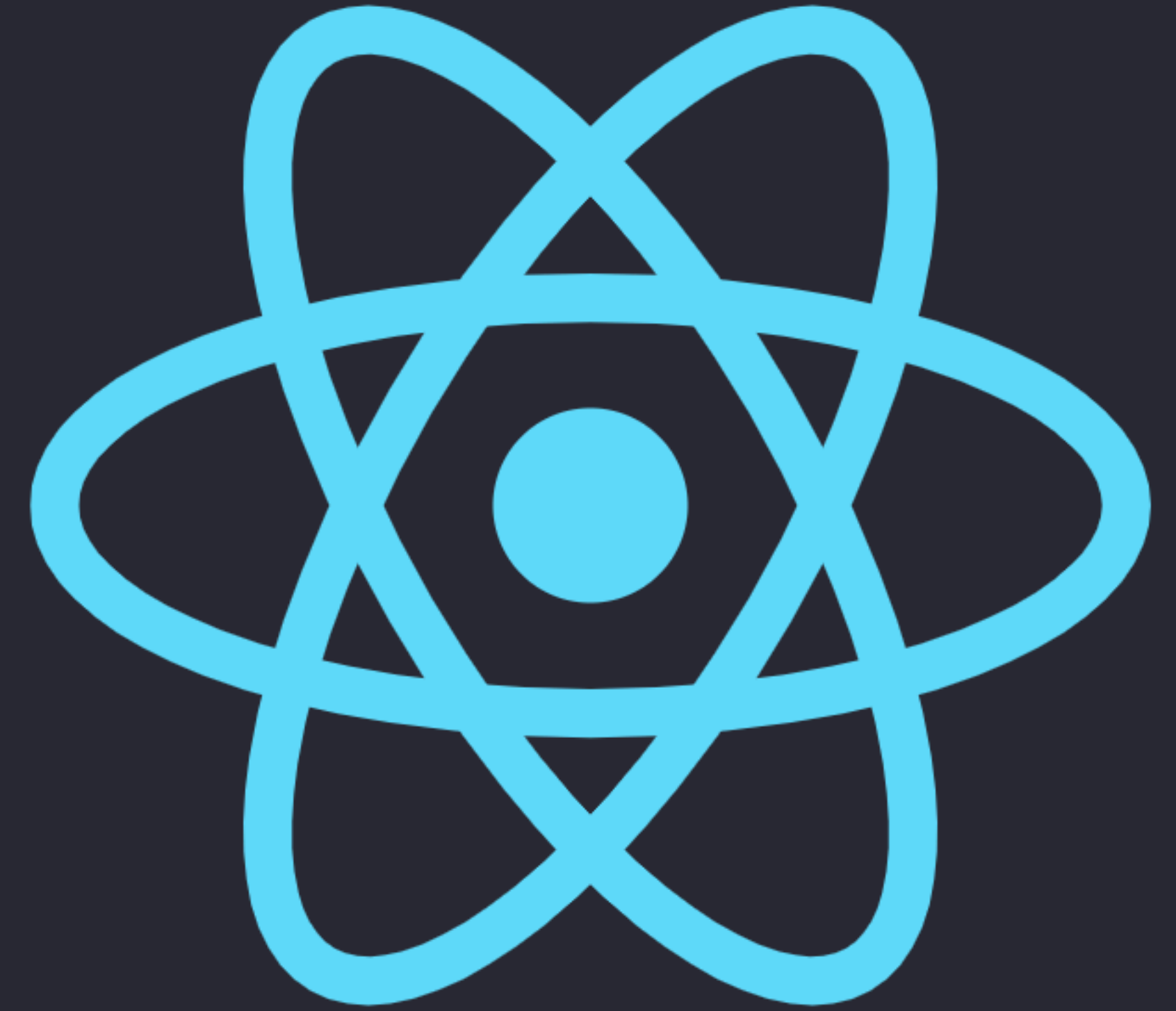


# React Native

Aprenda uma vez, escreva em qualquer lugar.



# React Native

## Hooks de forma resumida

*Hooks* são uma nova adição no React 16.8. Eles permitem que você use o state e outros recursos do React sem escrever uma classe.

## February 16, 2019

On **February 16, 2019**, React 16.8 was released to the public. The release introduced React Hooks.

## Motivação

*Hooks* resolvem uma variedade de problemas aparentemente separados em React que encontramos ao longo de cinco anos escrevendo e mantendo milhares de componentes.

# React Native

## Porque Hooks?

**Classes Confundem tanto Pessoas quanto Máquinas**

**Componentes Complexos se Tornam Difíceis de Entender**

**É Difícil Reutilizar Lógica com Estado entre Componentes**

## **Estratégia Gradual de Adoção**

**TLDR: Não há planos de remover classes de React.**

# React Native

## State Hook

Este exemplo renderiza um contador. Quando você clica no botão, ele incrementa o valor:

```
import React, { useState } from 'react';

function Example() {
  // Declara uma nova variável de state, que chamaremos de "count"
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}
```

# React Native

## Hook de Efeito

```
import React, { useState, useEffect } from 'react';

function Example() {
  const [count, setCount] = useState(0);

  // Similar a componentDidMount e componentDidUpdate:
  useEffect(() => {
    // Atualiza o título do documento utilizando a API do navegador
    document.title = `You clicked ${count} times`;
  });

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}
```

Versão em RN!

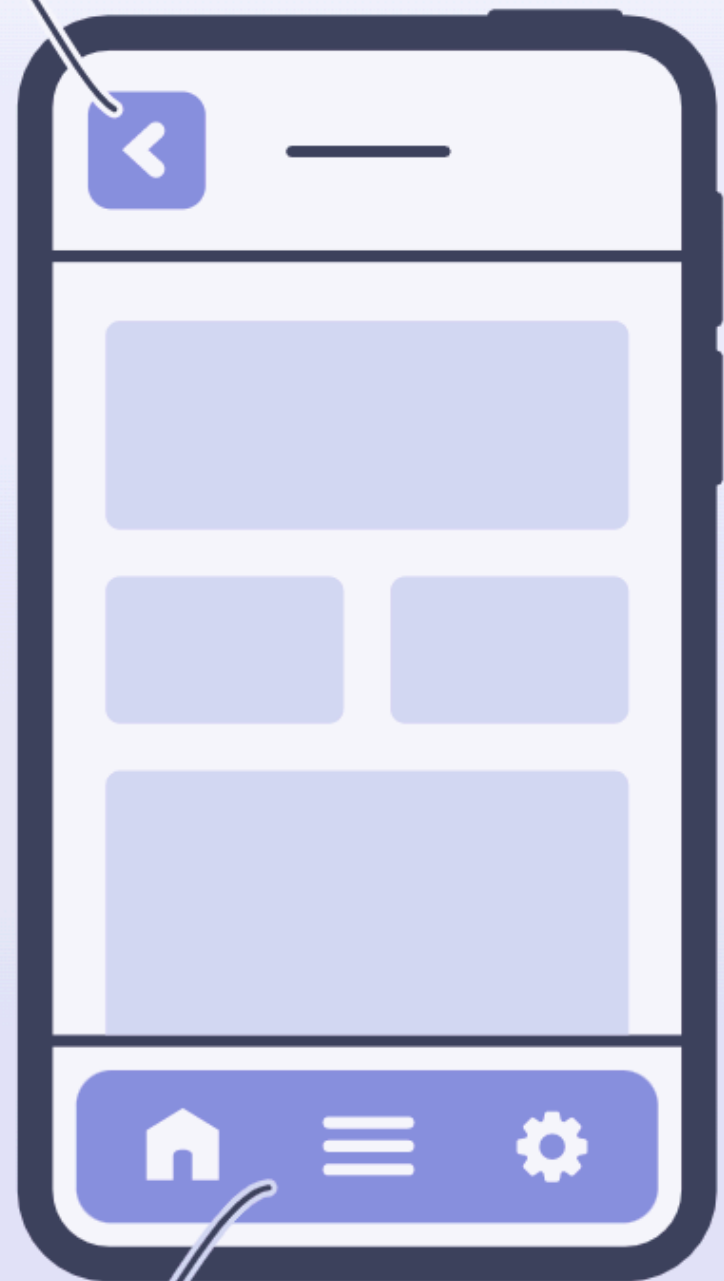


hooks

# Navegação entre Telas



Stack



Tabs

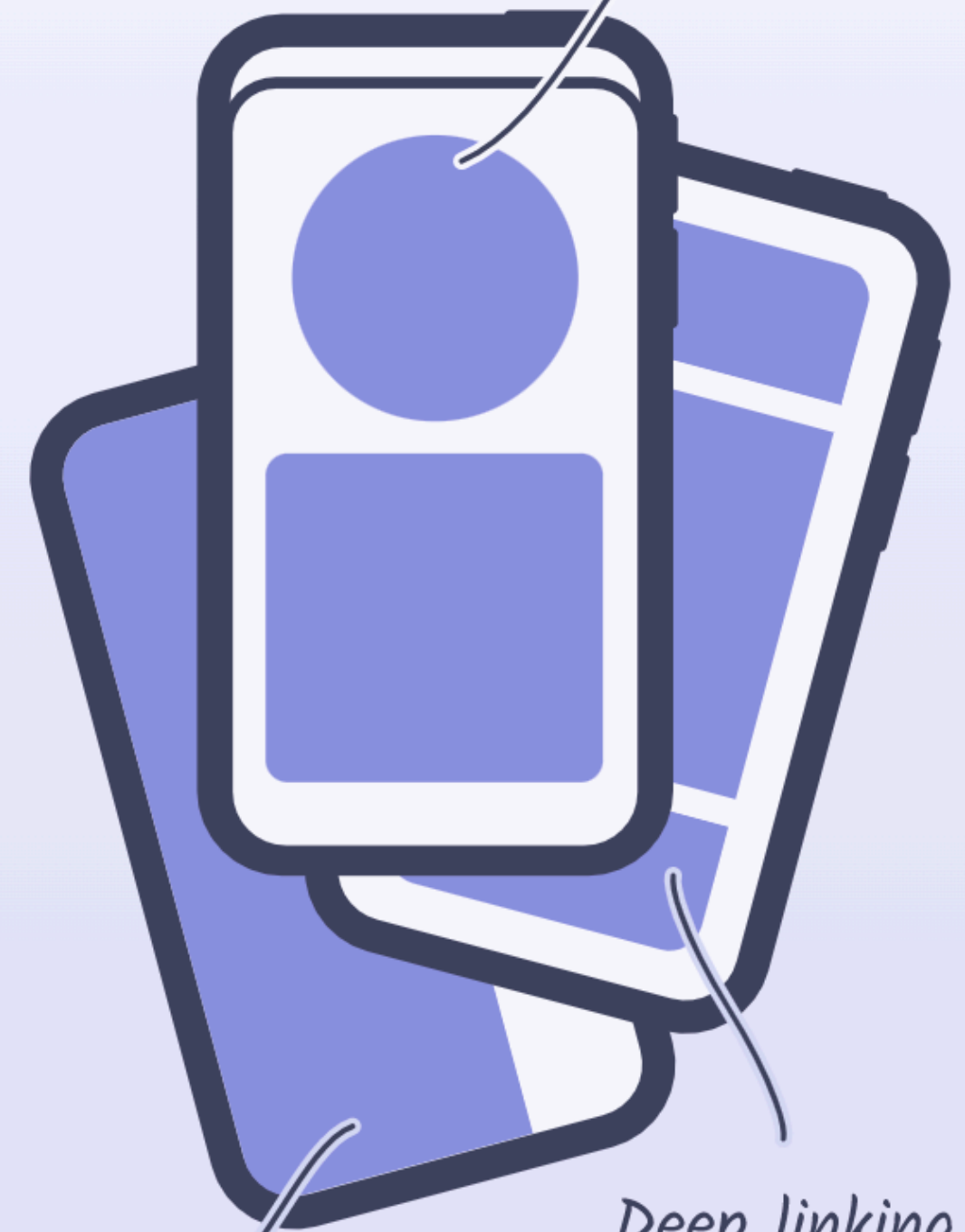
# React Navigation

Routing and navigation for Expo and React Native apps.

Read Docs

Try It

Modal



Drawer

Deep linking

# React Native

## Stack Navigator

O navegador de pilha nativo do React Navigation fornece uma maneira de seu aplicativo fazer a transição entre telas e gerenciar o histórico de navegação. Se seu aplicativo usa apenas um navegador de pilha, é conceitualmente semelhante a como um navegador da web lida com o estado de navegação - seu aplicativo empurra e retira itens da pilha de navegação conforme os usuários interagem com ele, e isso resulta no usuário vendo telas diferentes. Uma diferença importante entre como isso funciona em um navegador da web e no React Navigation é que o navegador de pilha nativo do React Navigation fornece os gestos e animações que você esperaria no Android e iOS ao navegar entre as rotas na pilha.



# React Native

## Stack Navigator

npm   **Yarn**

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

💡 `@react-navigation/native-stack` depends on `react-native-screens` and the other libraries that we installed in [Getting started](#). If you haven't installed those yet, head over to that page and follow the installation instructions.

# React Native

*// In App.js in a new project*

```
import * as React from 'react';
import { View, Text } from 'react-native';
import { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from '@react-navigation/native-stack';

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

const Stack = createNativeStackNavigator(); ←

function App() {
  return (
    <NavigationContainer>
      <Stack.Navigator>
        <Stack.Screen name="Home" component={HomeScreen} />
      </Stack.Navigator>
    </NavigationContainer>
  );
}

export default App;
```

```
<Stack.Screen
  name="Home"
  component={HomeScreen}
  options={{ title: 'Overview' }}
/>
```



navigation

# React Native

```
import * as React from 'react';
import { Button, View, Text } from 'react-native';
import { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from '@react-navigation/native-stack';

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

// ... other code from the previous section
```

Copy

Utilizar o **push**  
para empilhar!

Utilizar o **goBack**  
para voltar!

Utilizar o  
**popToTop** para  
voltar para a  
primeira tela!

# React Native

```
function StackScreen() {  
  return (  
    <Stack.Navigator>  
      <Stack.Screen  
        name="Home"  
        component={HomeScreen}  
        options={{  
          title: 'My home',  
          headerStyle: {  
            backgroundColor: '#f4511e',  
          },  
          headerTintColor: '#fff',  
          headerTitleStyle: {  
            fontWeight: 'bold',  
          },  
        }}  
      />  
    </Stack.Navigator>  
  );  
}
```

Copy

Configurando a  
barra superior!

# React Native

```
function StackScreen() {  
  return (  
    <Stack.Navigator  
      screenOptions={{  
        headerStyle: {  
          backgroundColor: '#f4511e',  
        },  
        headerTintColor: '#fff',  
        headerTitleStyle: {  
          fontWeight: 'bold',  
        },  
      }}  
    >  
    <Stack.Screen  
      name="Home"  
      component={HomeScreen}  
      options={{ title: 'My home' }}  
    />  
  </Stack.Navigator>  
);  
}
```

Utilizando a mesma  
estilização para todas  
as telas na navegação!



# React Native

## Tab Navigator

Possivelmente, o estilo mais comum de navegação em aplicativos móveis é a navegação baseada em guias. Podem ser guias na parte inferior da tela ou na parte superior abaixo do cabeçalho (ou mesmo em vez de um cabeçalho).

Este guia cobre `createBottomTabNavigator`. Você também pode usar `createMaterialBottomTabNavigator` e `createMaterialTopTabNavigator` para adicionar guias ao seu aplicativo.

Antes de continuar, primeiro instale `@react-navigation/bottom-tabs`:

npm   Yarn

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

# React Native

```
import * as React from 'react';
import { Text, View } from 'react-native';
import { NavigationContainer } from '@react-navigation/native';
import { createBottomTabNavigator } from '@react-navigation/bottom-tabs';

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>
  );
}

const Tab = createBottomTabNavigator();

export default function App() {
  return (
    <NavigationContainer>
      <Tab.Navigator>
        <Tab.Screen name="Home" component={HomeScreen} />
        <Tab.Screen name="Settings" component={SettingsScreen} />
      </Tab.Navigator>
    </NavigationContainer>
  );
}
```

cópia de

tabs

# React Native

```
// You can import Ionicons from @expo/vector-icons/Ionicons if you use Expo or  
// react-native-vector-icons/Ionicons otherwise.
```

```
import Ionicons from 'react-native-vector-icons/Ionicons';
```

```
// (...)
```

```
export default function App() {  
  return (  
    <NavigationContainer>
```

```
    <Tab.Navigator
```

```
      screenOptions={({ route }) => ({
```

```
        tabBarIcon: ({ focused, color, size }) => {
```

```
          let iconName;
```

```
          if (route.name === 'Home') {
```

```
            iconName = focused
```

```
              ? 'ios-information-circle'
```

```
              : 'ios-information-circle-outline';
```

```
          } else if (route.name === 'Settings') {
```

```
            iconName = focused ? 'ios-list-box' : 'ios-list';
```

```
          }
```

```
// You can return any component that you like here!
```

```
          return <Ionicons name={iconName} size={size} color={color} />;
```

```
        },
```

```
        tabBarActiveTintColor: 'tomato',
```

```
        tabBarInactiveTintColor: 'gray',
```

```
      })}
```

```
    >
```

```
    <Tab.Screen name="Home" component={HomeScreen} />
```

```
    <Tab.Screen name="Settings" component={SettingsScreen} />
```

```
  </Tab.Navigator>
```

```
</NavigationContainer>
```

```
);
```

```
}
```

cópia de

## Personalização dos ícones da tabs

# React Native

```
<Tab.Screen name="Home" component={HomeScreen} options={{ tabBarBadge: 3 }} />
```

Da perspectiva da IU, este componente está pronto para uso, mas você ainda precisa encontrar uma maneira de transmitir a contagem de crachás corretamente de outro lugar, como usar [React Context](#), [Redux](#), [MobX](#) ou [emissores de evento](#).

Camera 14:25 100%

Home!

Home

Settings



# React Native

## Um navegador pilha nativa para cada guia

Normalmente, as guias não exibem apenas uma tela - por exemplo, em seu feed do Twitter, você pode tocar em um tweet e ele o leva a uma nova tela dentro dessa guia com todas as respostas. Você pode pensar nisso como existindo pilhas de navegação separadas dentro de cada guia, e é exatamente assim que iremos modelá-lo no React Navigation.

tabs-stack

```
const HomeStack = createNativeStackNavigator();

function HomeStackScreen() {
  return (
    <HomeStack.Navigator>
      <HomeStack.Screen name="Home" component={HomeScreen} />
      <HomeStack.Screen name="Details" component={DetailsScreen} />
    </HomeStack.Navigator>
  );
}

const SettingsStack = createNativeStackNavigator();

function SettingsStackScreen() {
  return (
    <SettingsStack.Navigator>
      <SettingsStack.Screen name="Settings" component={SettingsScreen} />
      <SettingsStack.Screen name="Details" component={DetailsScreen} />
    </SettingsStack.Navigator>
  );
}

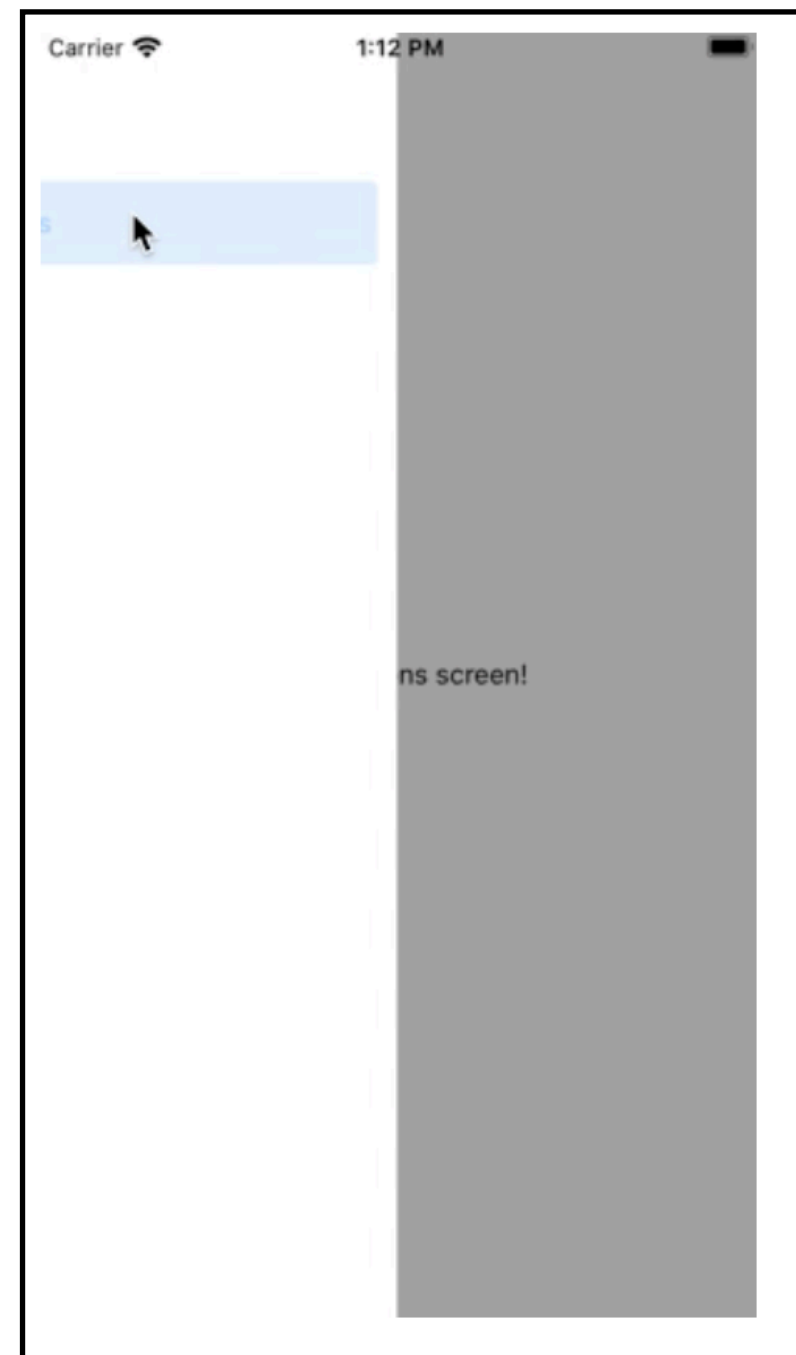
const Tab = createBottomTabNavigator();

export default function App() {
  return (
    <NavigationContainer>
      <Tab.Navigator>
        <Tab.Screen name="Home" component={HomeStackScreen} />
        <Tab.Screen name="Settings" component={SettingsStackScreen} />
      </Tab.Navigator>
    </NavigationContainer>
  );
}
```



# React Native

## Drawer Navigator



Para usar este navegador de gaveta, importe-o de `@react-navigation/drawer`:

# React Native

```
import * as React from 'react';
import { Button, 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>
  );
}
```

Separação dos  
componentes em  
pastas

full-nav

# React Native

## React Native

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

## React Navigation

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

## Full Navigation

<https://dev.to/easybuoy/combining-stack-tab-drawer-navigations-in-react-native-with-react-navigation-5-da>

## Deep link

<https://dev.to/techtalks/deep-linking-in-react-native-app-with-react-navigation-v5-41id>