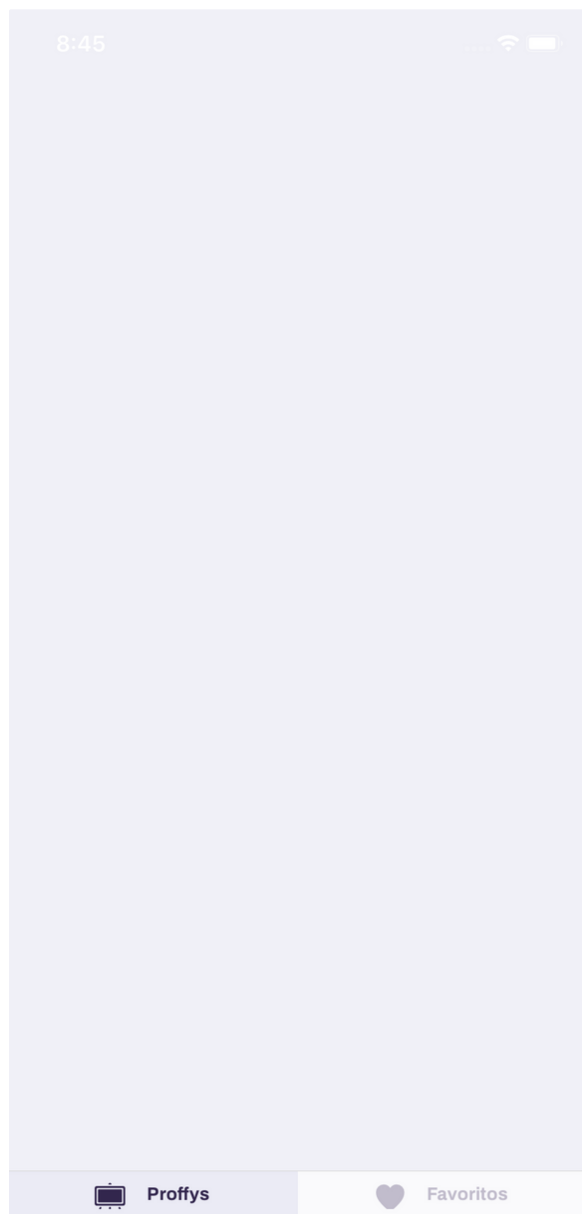


The Proffy logo is centered on a purple background. It features the word "Proffy" in a white, sans-serif font. The background is decorated with various green geometric shapes: circles, squares, and crosses, some of which are larger and more prominent than others.

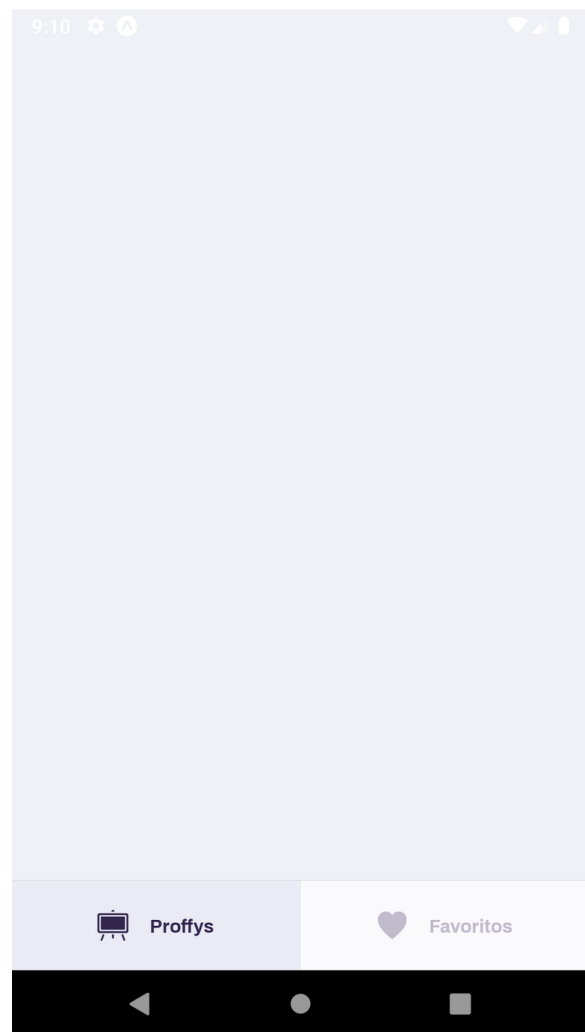
Ajustando estilo das abas no iOS

No trecho da aula onde é criada a **Bottom Tab Bar**, a estilização funciona perfeitamente no Android, porém são necessários alguns ajustes para o **iOS**, visto que os iPhones mais atuais possuem algumas partes que não são utilizáveis no ponto de vista dos componentes mas temos que considera-las na customização do Layout.

Abaixo está o exemplo de como ficou a estilização no **iOS** e no **Android** seguindo a aula:



Simulador do iOS



Emulador do Android

E para corrigir esse problema de estilização vamos utilizar uma funcionalidade do **React Native**, o **Platform**, que consiste em um módulo que disponibiliza a informação de qual plataforma a aplicação está rodando.

Platform Specific Code · React Native

When building a cross-platform app, you'll want to re-use as much code as possible. Scenarios may arise where it makes sense for the



<https://reactnative.dev/docs/platform-specific-code#platform-...>



Ele é comumente utilizado para mudar a estilização de componentes dependendo da plataforma, segue um exemplo da própria documentação:

```
import { Platform, StyleSheet } from 'react-native'; const styles =  
StyleSheet.create({ height: Platform.OS === 'ios' ? 200 : 100 });
```

JavaScript ▾



Como você pode perceber dentro do **Platform** tem uma constante, a **OS** que é obtida diretamente do dispositivo que está rodando a aplicação, e vai retornar ou **ios** ou **android**.

Dada a explicação sobre esse módulo que vamos utilizar, vamos modificar a Bottom Tab bar, que está em `src/routes/StudyTabs.tsx` !

A primeira coisa que deve ser feita é a importação do **Platform**, para isso basta adicionar logo depois da importação do **React** a seguinte linha:

```
import { Platform } from 'react-native';
```

JavaScript ▾

Feito isso vamos ter que fazer 3 estilizações condicionais, a primeira delas é na propriedade **height** dentro de **styles** no **tabBarOptions**, utilizando o **Platform**, o **style** vai ficar assim:

```
tabBarOptions={{ style: { elevation: 0, shadowOpacity: 0, height: Platform.OS  
=== 'ios' ? 84 : 64, // Linha modificada }, tabStyle: { // ... }}
```

JavaScript ▾

No código acima, a linha que foi modificada vai garantir que quando o sistema do dispositivo for **iOS**, o **height** tenha o valor **84** e se for **Android** o **height** vai ter o valor **64**.

A segunda modificação que vamos fazer é no **tabStyle** dentro do **tabBarOptions**, nesse trecho vamos adicionar uma nova propriedade para aplicar um **paddingBottom** para **iOS**.

O código vai ficar assim:

```
tabBarOptions={{ // ... tabStyle: { flexDirection: 'row', alignItems:  
'center', justifyContent: 'center', paddingBottom: Platform.OS === 'ios' ? 20  
: 0, // Linha modificada }, // ... }}
```

JavaScript ▾

Essa modificação vai fazer com que seja aplicado um **paddingBottom** apenas no **iOS**, pois no **Android** não precisamos de nenhum espaçamento extra, assim como você viu na aula.

A terceira modificação é no **height** dentro do **iconStyle** no **tabBarOptions**, essa modificação vai fazer com que o ícone não tenha um pedaço da parte de baixo cortada, ficando assim:

```
tabBarOptions={{ // ... iconStyle: { flex: 0, width: 20, height: Platform.OS  
=== 'ios' ? 24 : 20, // Linha modificada }, // ... }}
```

JavaScript ▾

Para finalizar vamos ter que adicionar mais uma opção no **tabBarOptions** para que o estilo fique correto no **iOS**, essa opção é o **safeAreaInsets**.

React Navigation

A simple tab bar on the bottom of the screen that lets you switch between different routes. Routes are lazily initialized -- their screen components are not mounted until they are first focused.



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

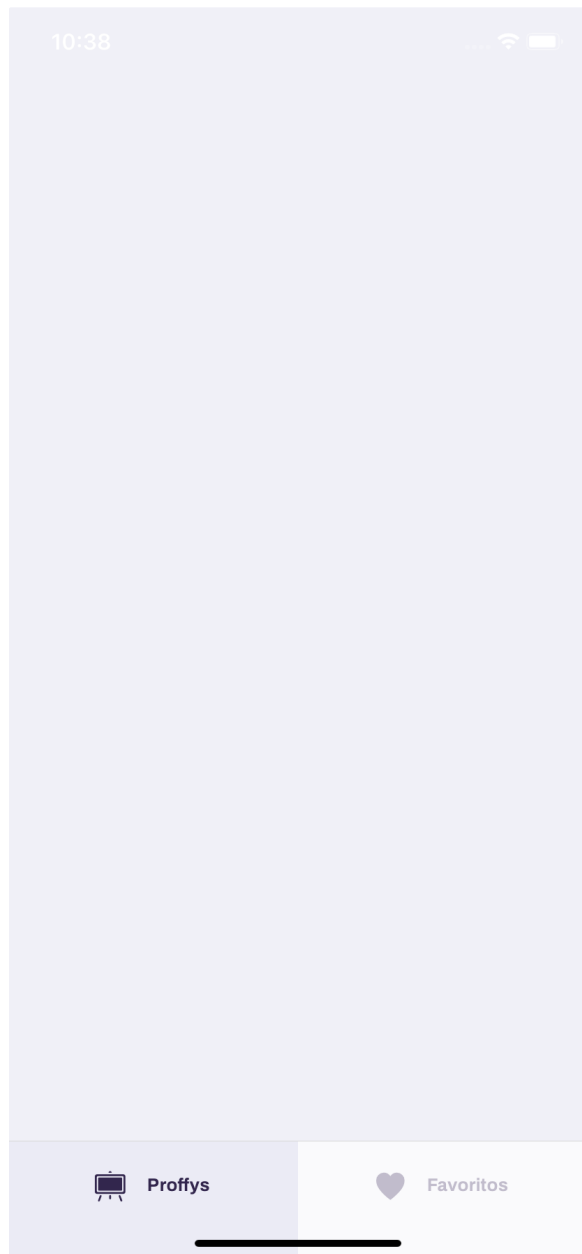
Ela garante que a **Bottom Tab Bar** não dê nenhum espaçamento da "área de segurança" na parte de baixo do **iPhone**, você pode adicionar essa opção depois do `tabStyle` , ficando assim:

```
tabBarOptions={{ tabStyle: { // ... }, safeAreaInsets: { bottom: 0, },  
iconStyle: { // ... }, }}
```

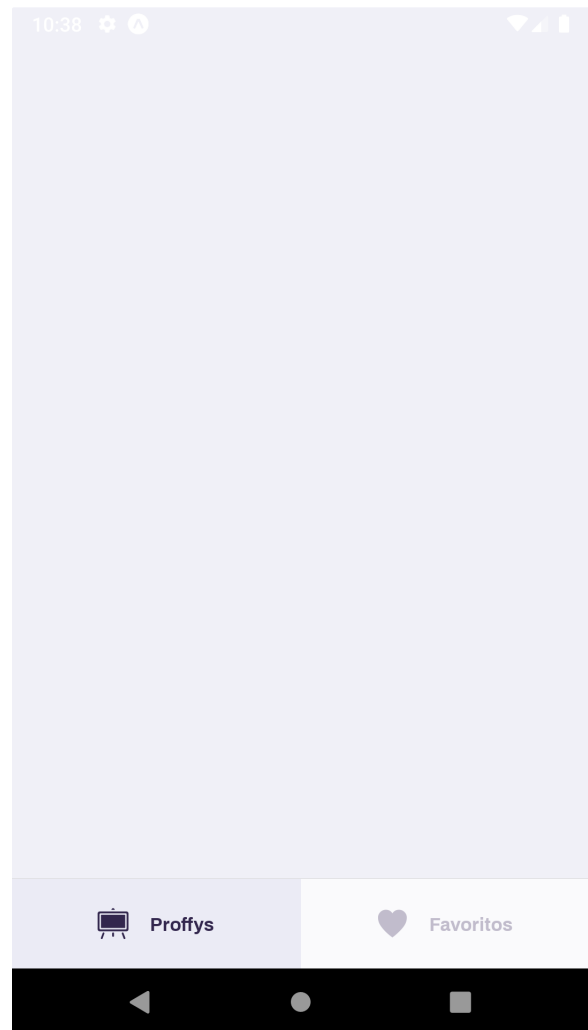
JavaScript ▾

Essa opção não vamos utilizar o Platform, pois no Android esse valor já corresponde à 0 (zero) então nenhum efeito visual será alterado.

E assim finalizamos o ajuste na Bottom Tab bar para que ela fique com o mesmo visual igual em ambas as plataformas.



Simulador do iOS - Resultado Final



Emulador do Android - Resultado Final

Bora codar! 🚀