React Native

Funcionamento

<https://tableless.com.br/o-que-voce-deve-saber-sobre-funcionamento-react-native/>

Guia

<https://facebook.github.io/react-native/>

Instalação

<https://facebook.github.io/react-native/docs/getting-started>

FlexBox no React Native

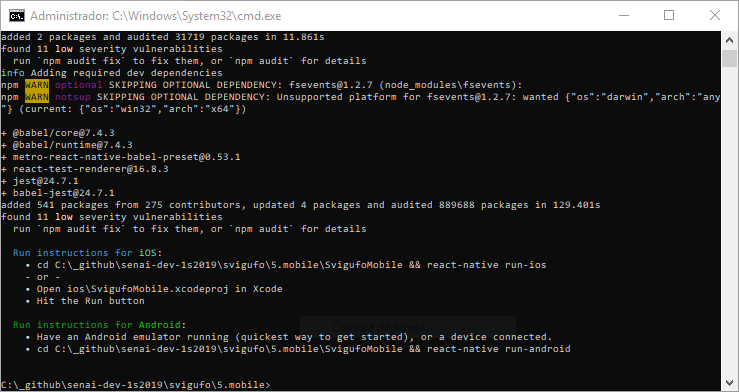
<https://www.youtube.com/watch?v=dUkkOdhyO1w>

JSX

Produzindo a interface nativa

Criar um projeto

react-native init SvigufoMobile



react-native run-android

Bundler

Lê tudo o que escrevemos

Transpila com o Babel

Joga o js unificado para o dispositivo

Para que ele renderize

Depois de rodar uma vez e gerar o bundler, basta apenas digitar o comando abaixo para iniciar novamente o projeto:

react-native start

react-native run-android somente quando instalar alguma outra dependência.

index.js – é o arquivo que vai carregar assim que a nossa aplicação for montada

AppRegistry – arquivo inicial da aplicação

Componentização – componente – conjunto de código de visualização + lógica + estilização

View – caixa sem estilização

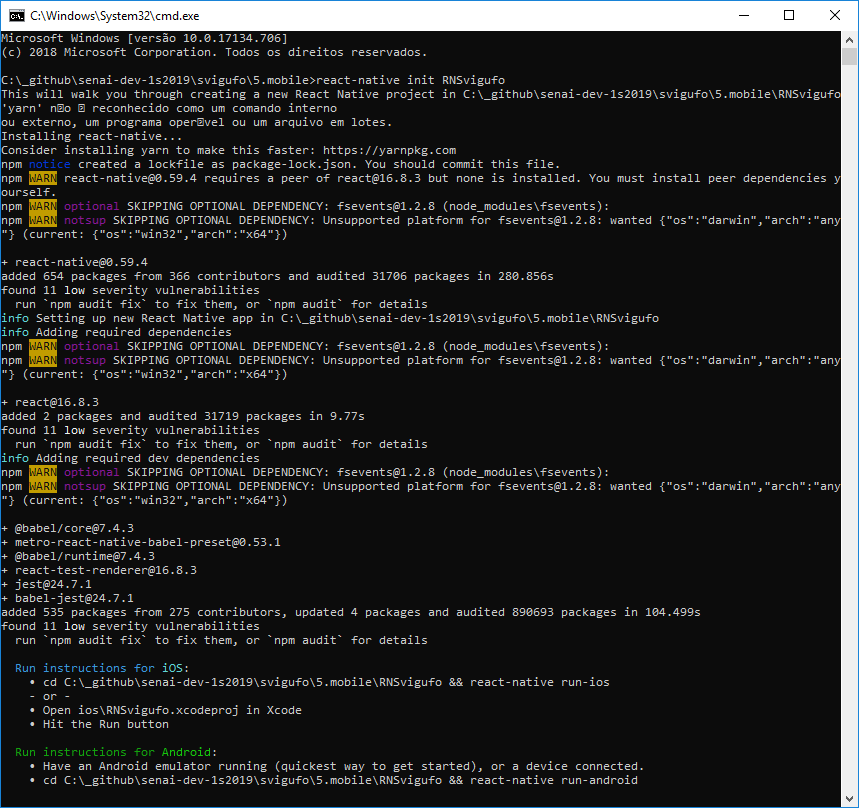
Não utilizamos classes no React Native (css)

render() é o único obrigatório pq é o que renderiza para o usuário

StyleSheet – não tem utilização de hífen, é com cammelcase e aspas para strings – é utilizado flexbox – Todos os elementos já estão com display: flex;

**ROTEIRO**

react-native init RNSvigufo



cd RNSvigufo

C:\Users\helena.strada\Documents\android\emulator

emulator -list-avds

emulator -avd avd\_name [ {-option [value]} … ]

avdmanager.bat create avd --force --name avdRN --abi google\_apis\_playstore/x86 --package "system-images;android-27;google\_apis\_playstore;x86"

emulator -list-avds

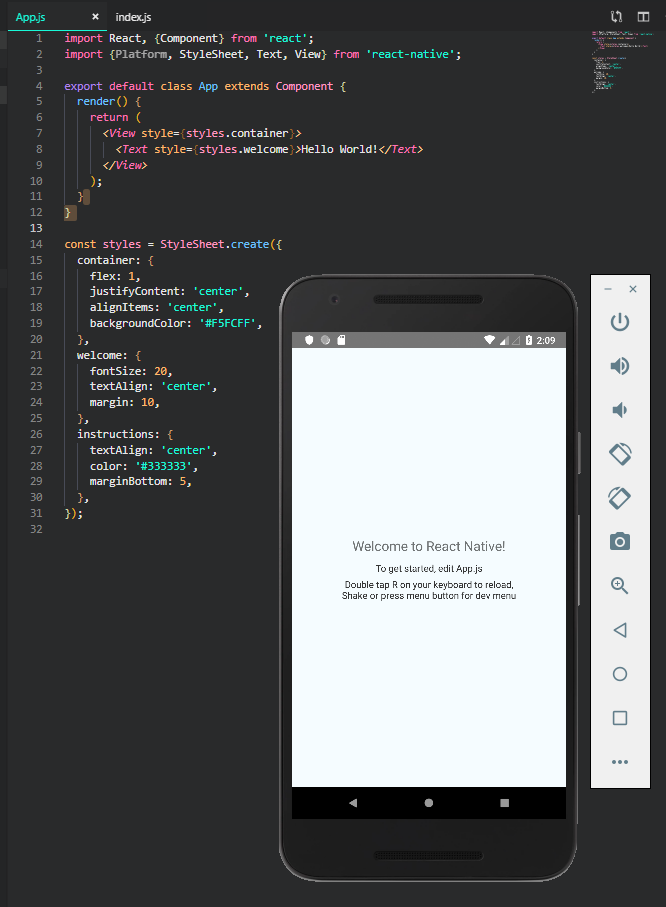
emulator -avd avdRN

Abrir uma AVD (Android Virtual Device) para realizar os testes durante o desenvolvimento.

react-native run-android

Início abaixo para criação do projeto

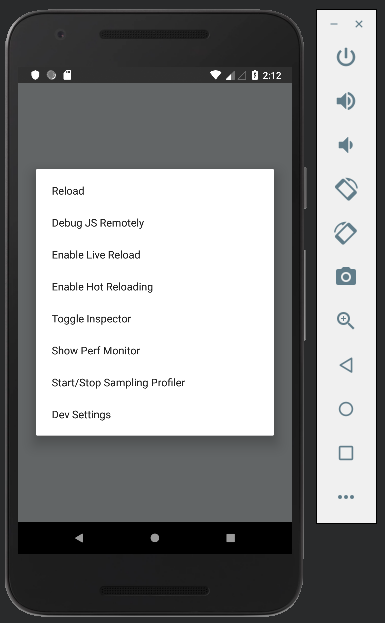
Remover os props e alterar o valor do Text.



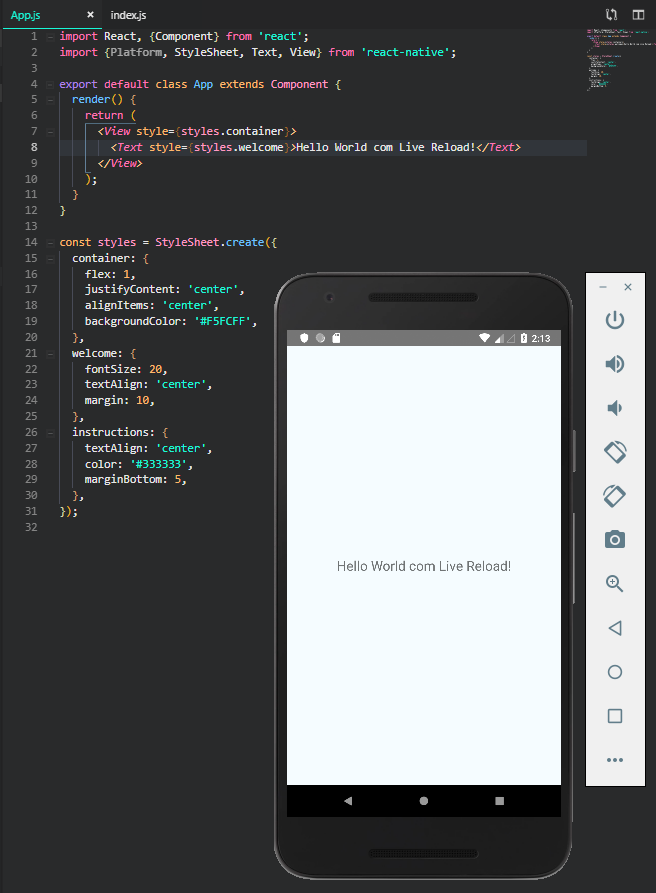
Para atualizar o valor na tela, basta ir na avd e apertar R 2x. Há uma outra opção para habilitar o live reload.

Para abrir o menu, aperte CTRL+M.

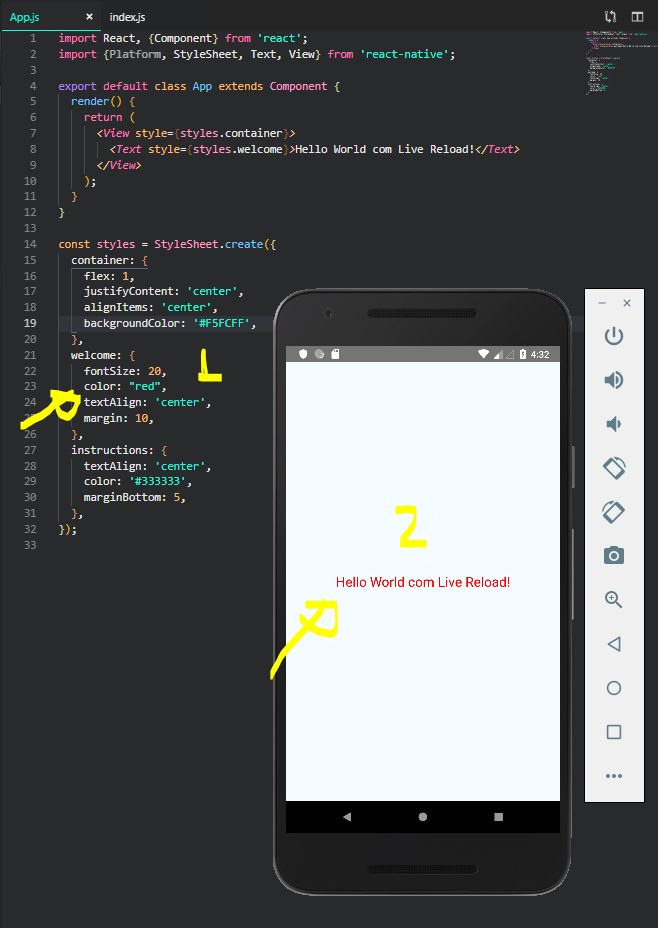
Coloque a opção Enable Live Reload.



Altere o texto do conteúdo e visualize as informações alteradas no mobile.



Alterar a cor do texto



**TÓPICOS**

Criar src

Criar as pastas : assets, config, pages, services

Arquivo index.js na pasta ‘src’

No arquivo index.js da raiz do projeto, alterar a navegação para ser chamada para o index.js

Criar o arquivo index.js na src, colocar os itens referentes a bottomTabNavigator e no AppContainer, incluir somente este navigator criado

Criar a main com os tópicos selecionados

Incluir axios

Criar profile

Incluir os itens associados

Rodar a aplicação e testar a lista e o perfil

Criar no index.js da src, a stackNavigator e incluir um switchNavigator no appContainer, incluindo a que foi criada

Criar a tela de signin, incluir os itens selecionados

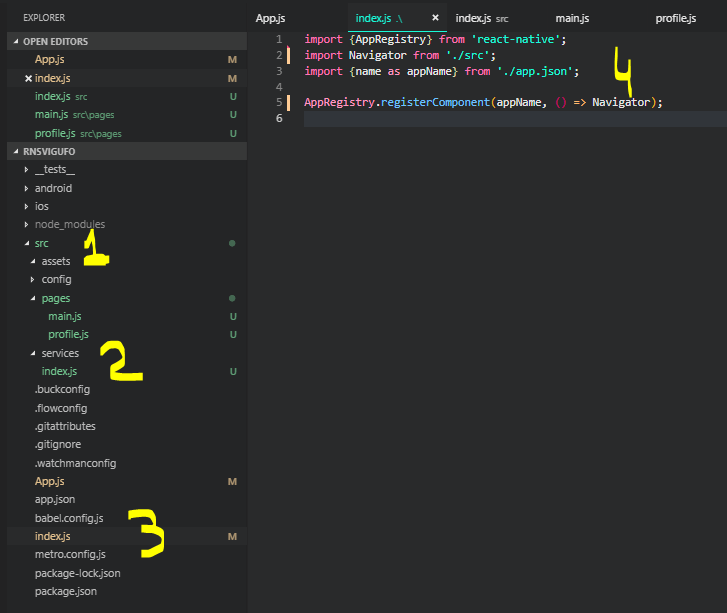
Alterar a tela de profile, para buscar os valores do token

npm install jwt-decode

Criar a pasta src

assets, config, pages, services

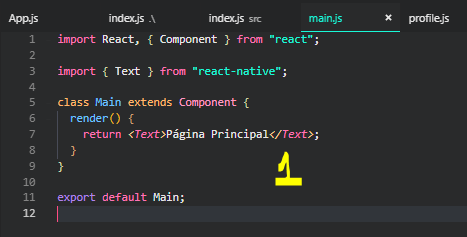
index.js



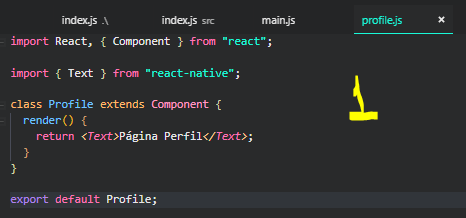
Criar dois novos arquivos dentro da pasta de pages (main.js e profile.js)

Alterar o arquivo index.js na raiz do projeto para importar o arquivo Navigator da pasta ./src/index.js

Conteúdo da main.js



Conteúdo da profile.js



Como realizar a navegação entre as telas?

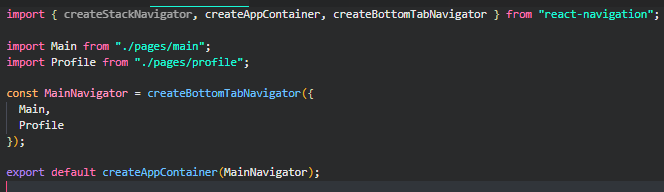
<https://facebook.github.io/react-native/docs/navigation>

npm install --save react-navigation

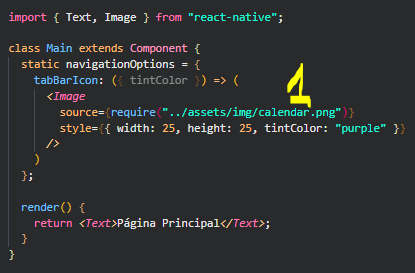
npm install --save react-native-gesture-handler

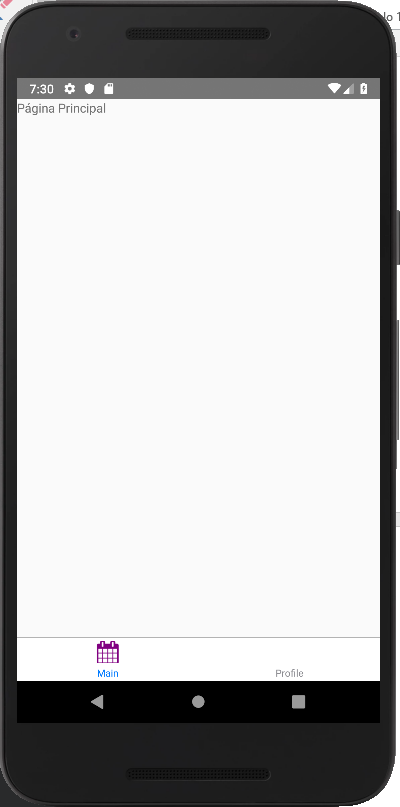
react-native link react-native-gesture-handler

Alterar o arquivo index.js dentro da pasta src.

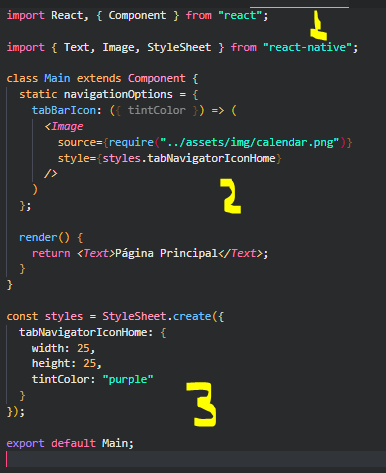


Alterar o main.js para adicionar um ícone na navegação

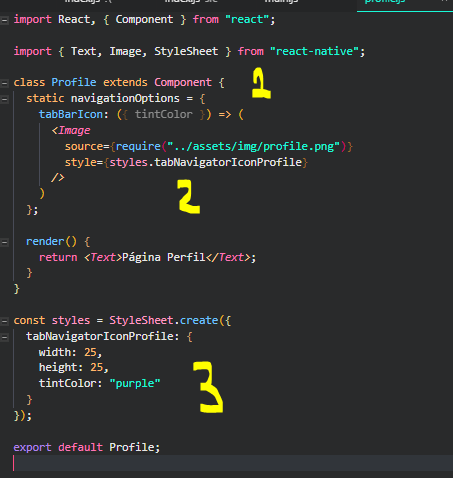




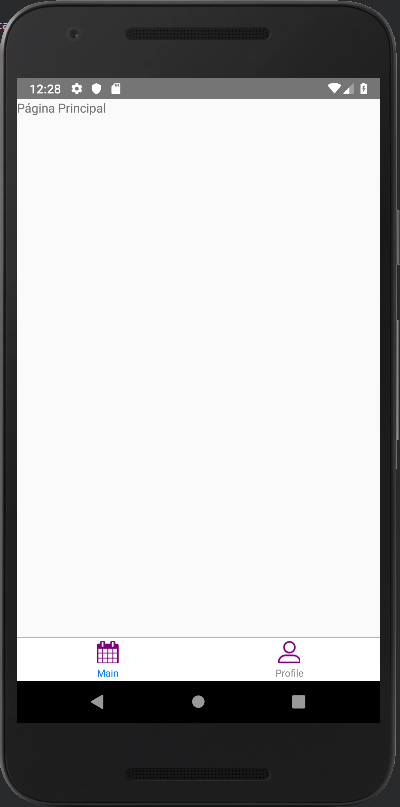
Importar o StyleSheet e incluir o estilo na main.js.



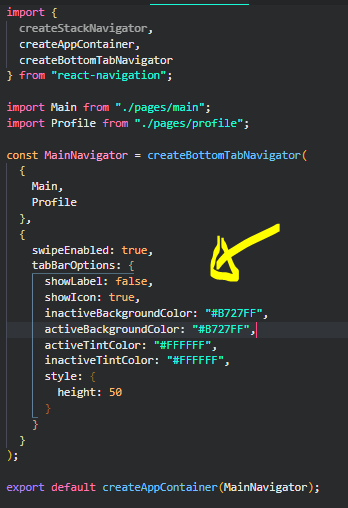
Alterar o estilo no profile.js.



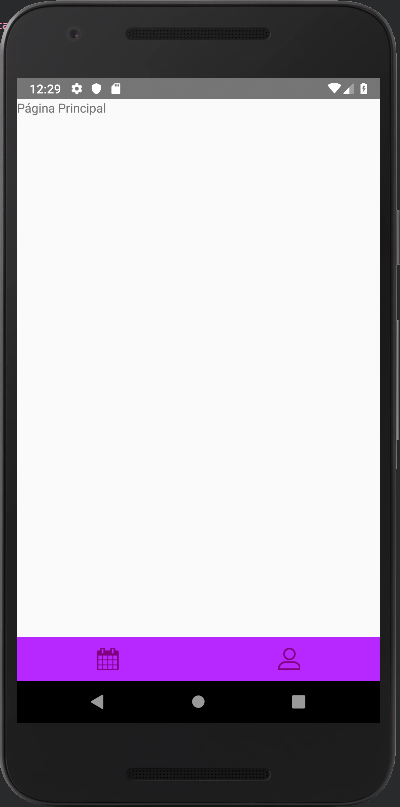
Resultado da alteração entre main e profile.js.



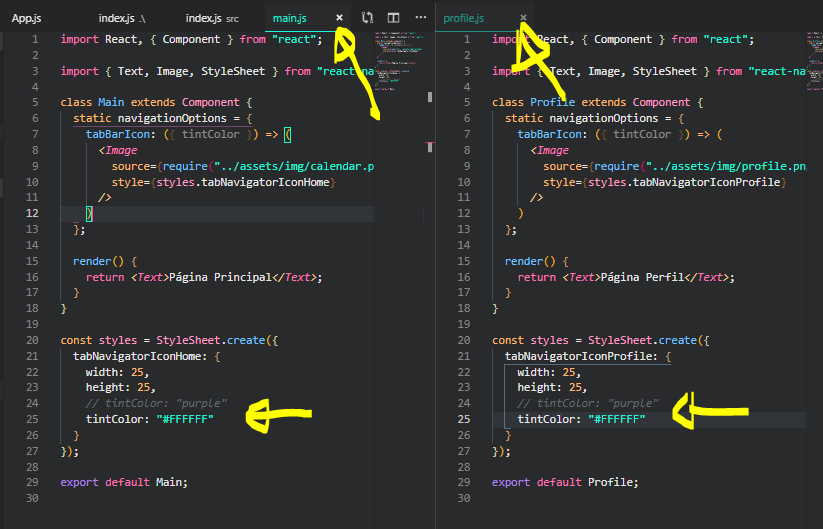
Alterando o index.js da pasta src.js para realizar a mudança na navegação.

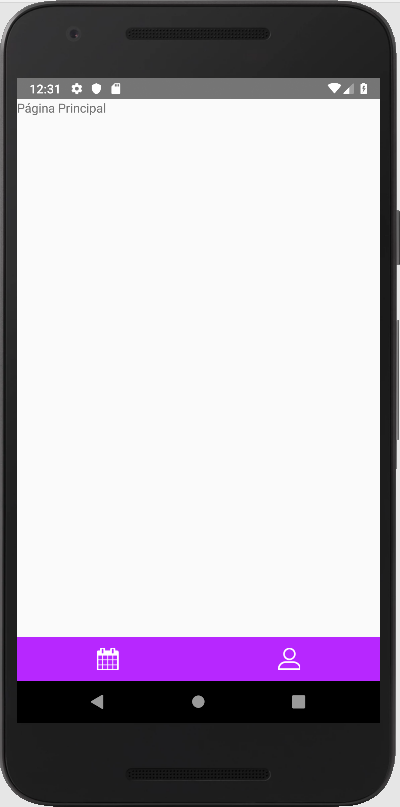


Resultado ao alterar o estilo



Alterar novamente o main.js e o profile.js para deixar os ícones brancos.



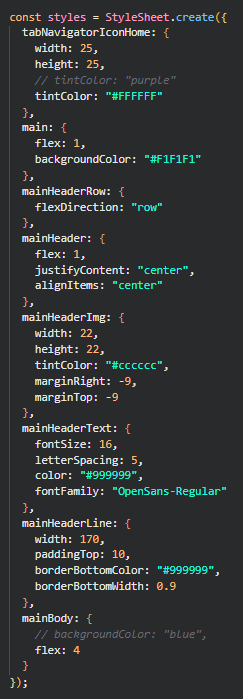


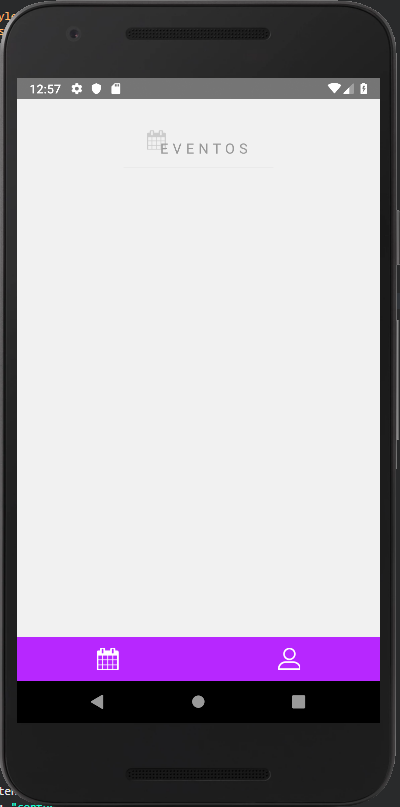
Alterando o main.js.

Alterar a construção do layout.

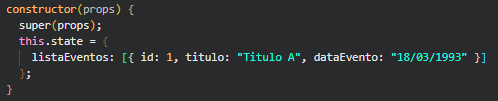


Alterar o conteúdo dos estilos.





Incluir a lista de eventos na main.js.



Incluir a FlatList

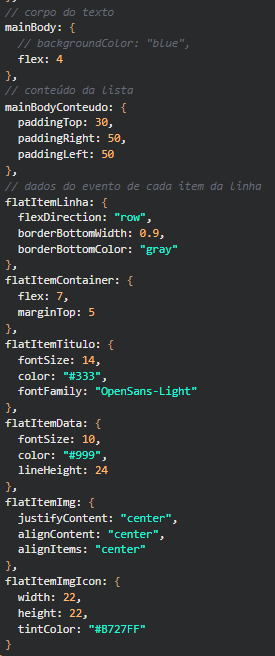
Possui alguns parâmetros: estilo, informação que será mostrada, a chave para cada item da linha, e qual método será chamado para renderizar os itens disponíveis.



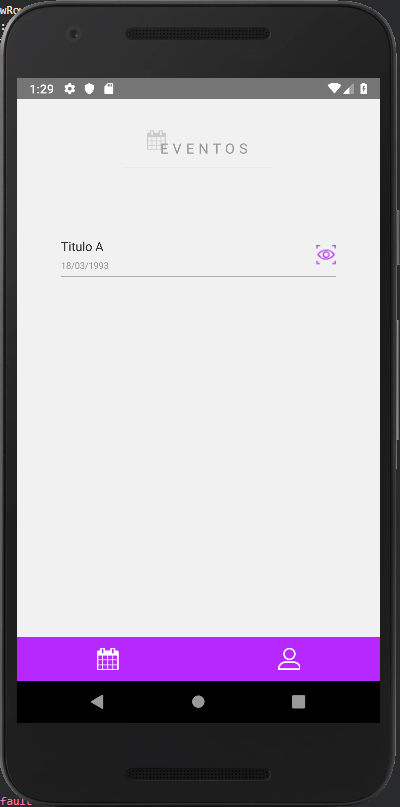
Criando cada item da linha.



Adicionando todos os estilos.



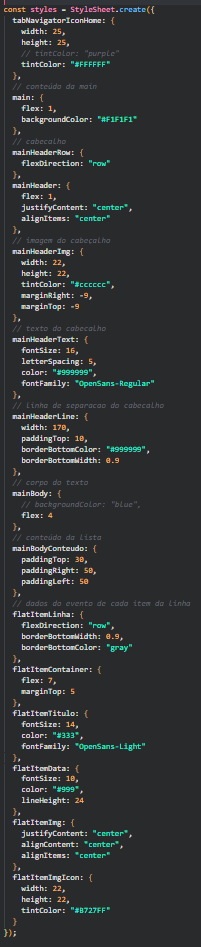
Resultado da Lista criada



Classe main.js completa



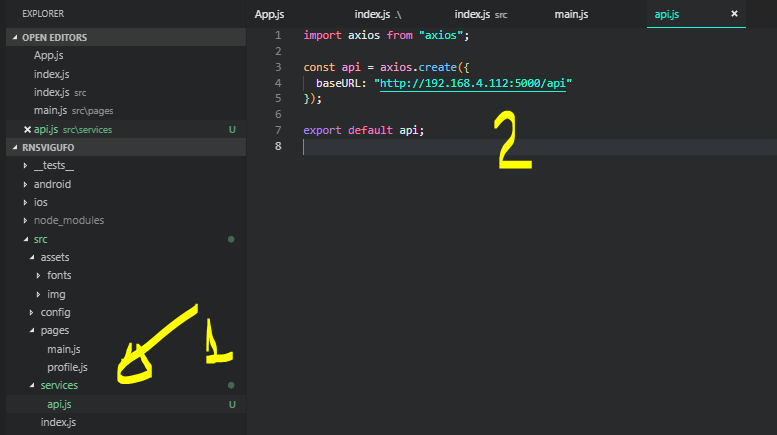
Estilo da main.js completo.



Realizado a chamada a API

npm install --save axios

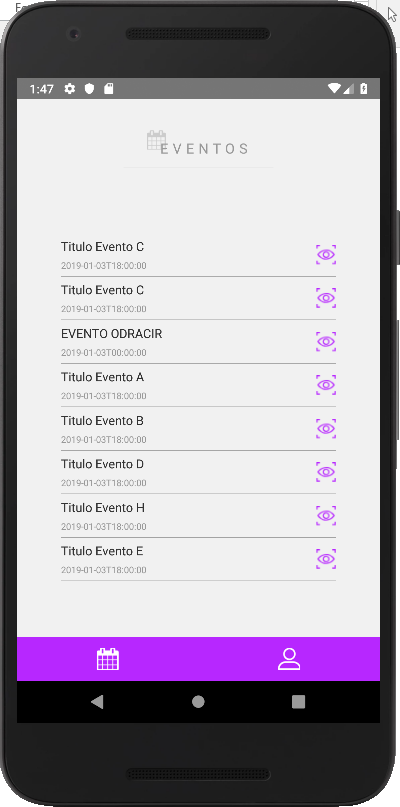
Criar um arquivo chamado api.js dentro da pasta src.



Realizar a alteração no main.js para realizar a chamada à API.



Resultado da alteração.

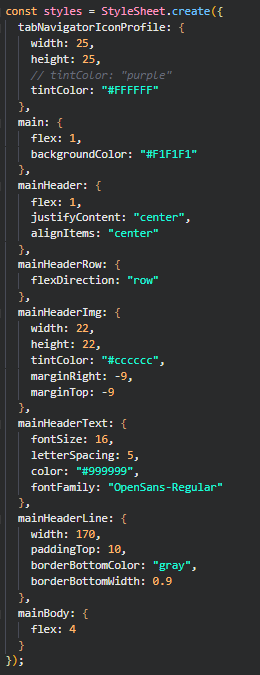


Atualizando o header da página de perfil (profile).

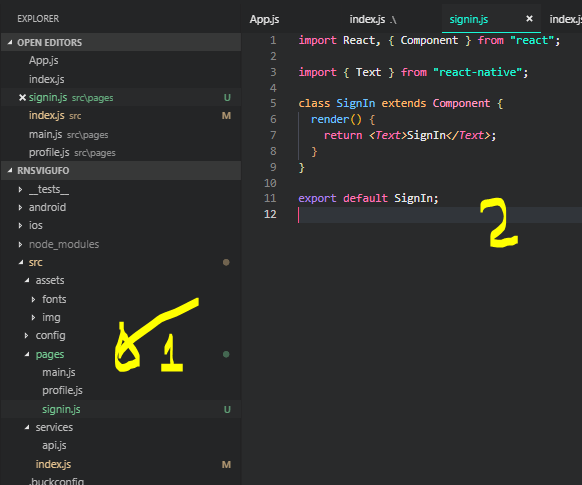
profile.js



Criando um estilo no profile.js.



Criando tela de login.



Alterar o index.js dentro da pasta src.



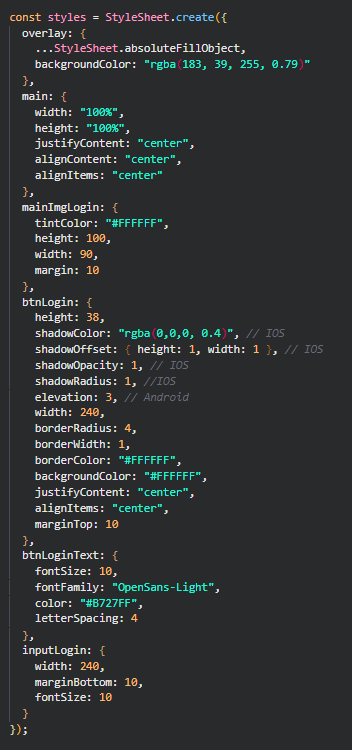
Resultado da tela de login.



Realizando as alterações na tela de login (signin.js).



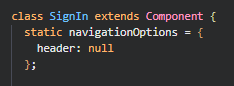
Adicionar o estilo ao signin.js.

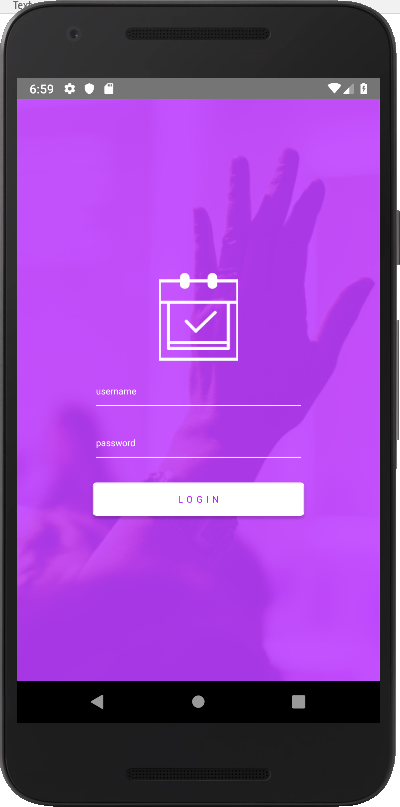


Resultado



Removendo a barra de cima de navegação.

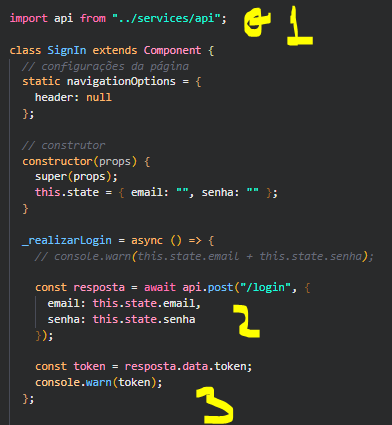




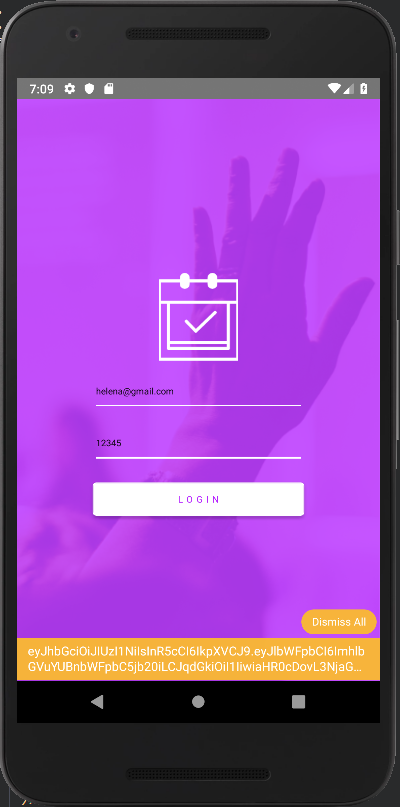
Buscando os valores dos inputs e mostrando um warn na tela.



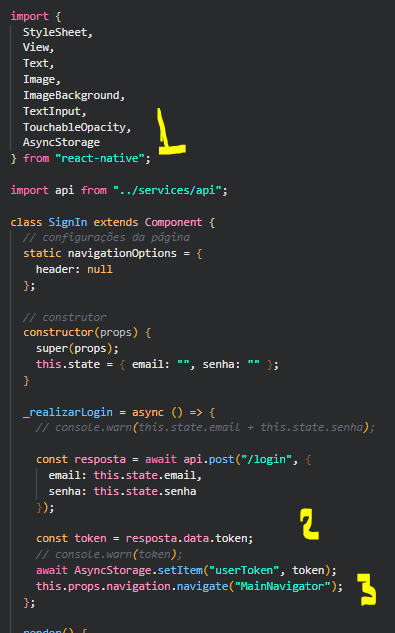
Realizar o envio dos dados de login para a API



Resultado



Guardando no AsyncStorage



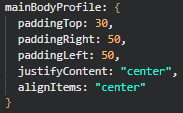
Buscando o valor do token

npm install --save jwt-decode

Não esquecer de importar o AsyncStorage







Adicionar o estilo no perfil.

Resultado

