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

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

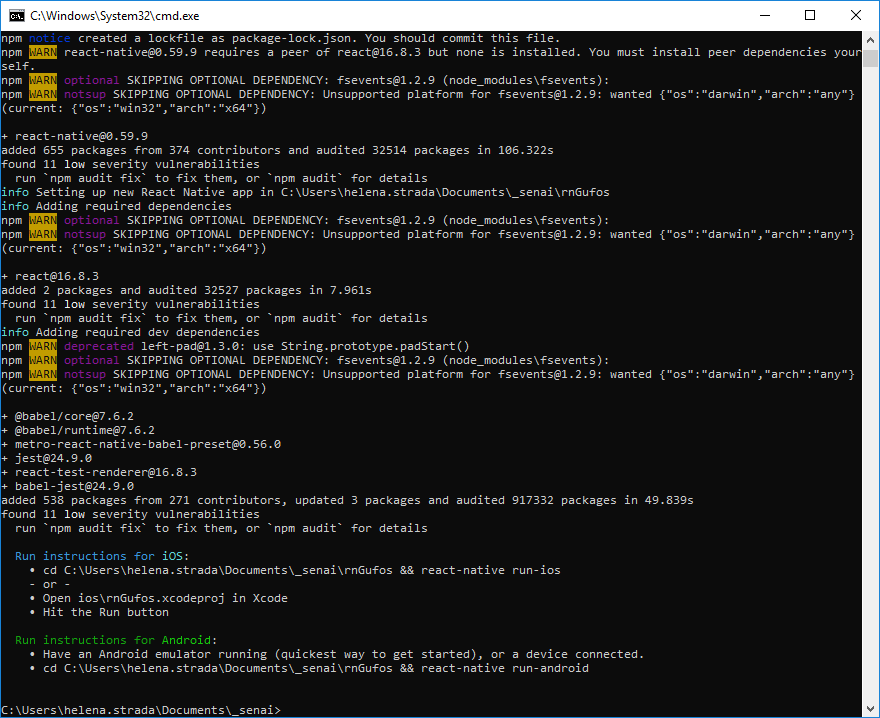
StyleSheet – não tem utilização de hífen, é com cammelcase e aspas para strings – é utilizado flexbox – Todos os elementos já estão com display: flex;

npm install -g react-native-cli

**ROTEIRO**

C:\Users\helena.strada\AppData\Roaming\npm\react-native init --version 0.60.6 Gufos

C:\Users\helena.strada\AppData\Roaming\npm\react-native init Gufos



cd Gufos

**Emuladores**

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

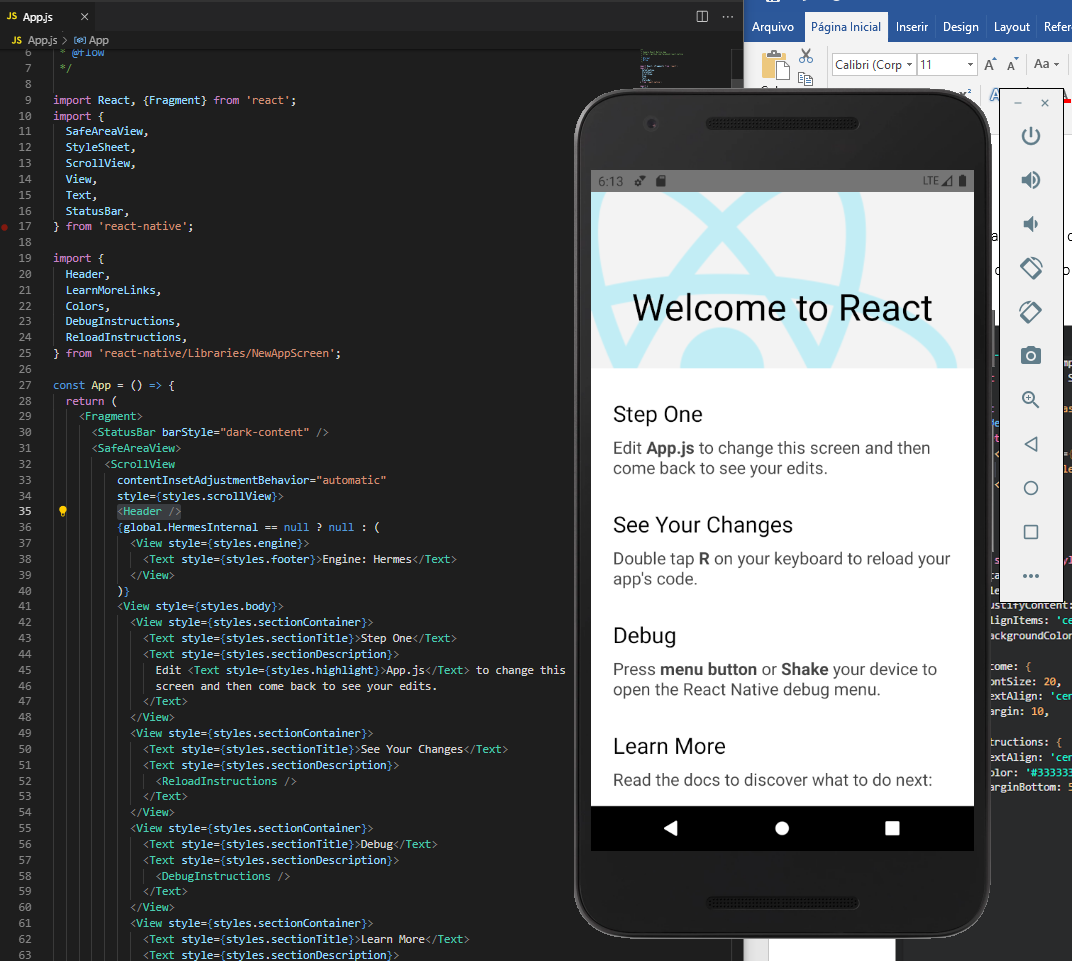
emulator -avd avdRN

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

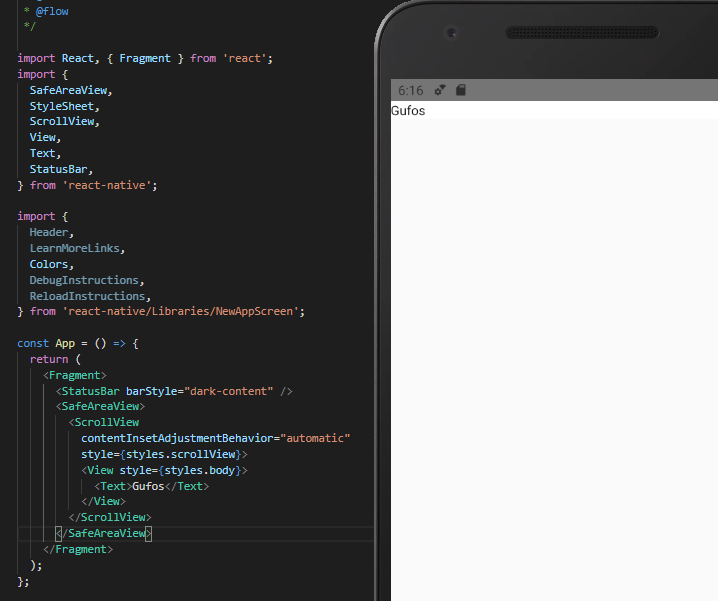
**Executar o Projeto**

react-native run-android

Rodar o projeto com o comando acima e apresentar o resultado inicial.



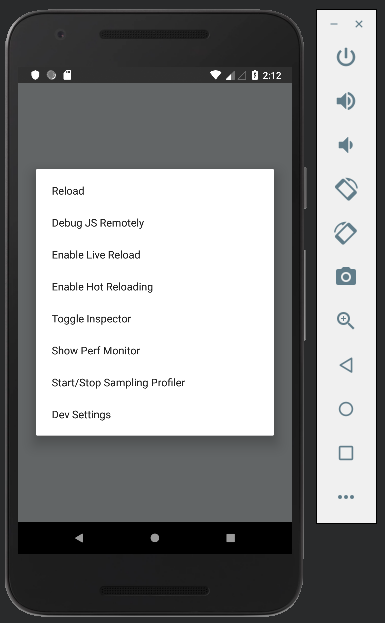
Remover todo o conteúdo do arquivo App.js e incluir somente um título de exemplo (Text).



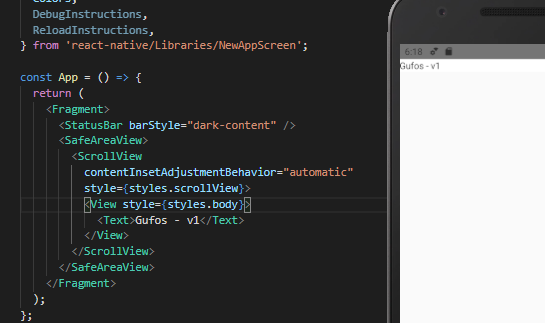
Para atualizar o valor na tela, basta ir na avd e apertar R 2x. Há uma outra opção para habilitar o live reload.

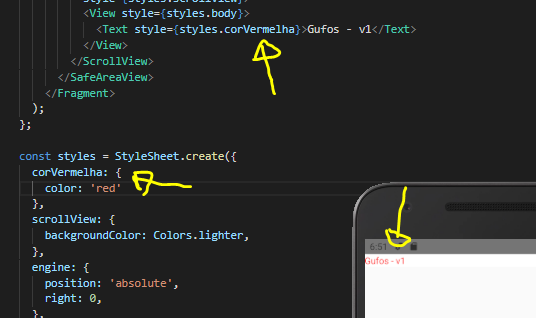
Para abrir o menu, aperte CTRL+M.

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

Criar os dois arquivos de main.js e profile.js.

Adicionar as rotas de navegação.

Criar um arquivo dentro da pasta src.

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

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

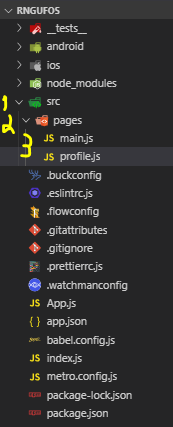
npm install jwt-decode

**Organização das pastas**

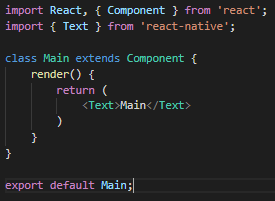
Criar a pasta src

Criar a pasta pages

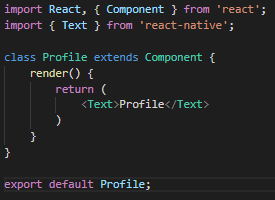
Criar os arquivos main.js e profile.js.



Conteúdo da main.js



Conteúdo da profile.js



**Navegação entre as telas**

Como realizar a navegação entre as telas?

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

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

necessário

npm install react-navigation

npm install react-native-reanimated react-native-gesture-handler [react-native-screens@^1.0.0-alpha.23](mailto:react-native-screens@%5e1.0.0-alpha.23)

android/app/build.gradle

dependencies

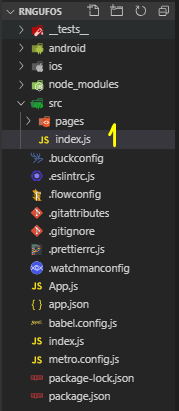
implementation 'androidx.appcompat:appcompat:1.1.0-rc01'

implementation 'androidx.swiperefreshlayout:swiperefreshlayout:1.1.0-alpha02'

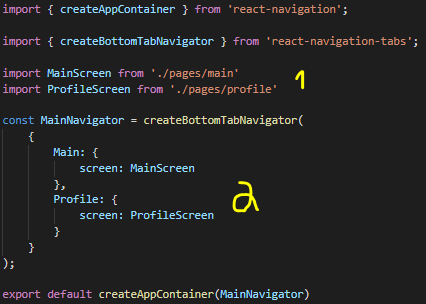
adicionar a dependência para ter acesso à navegação

npm install react-navigation-tabs

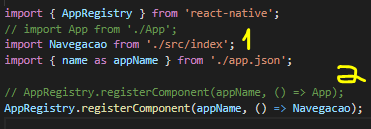
Criar um arquivo index.js dentro da pasta src e alterá-lo com o seguinte conteúdo.



index.js



Alterar o index.js do roteamento principal para dizer que é esta rota que queremos trabalhar.



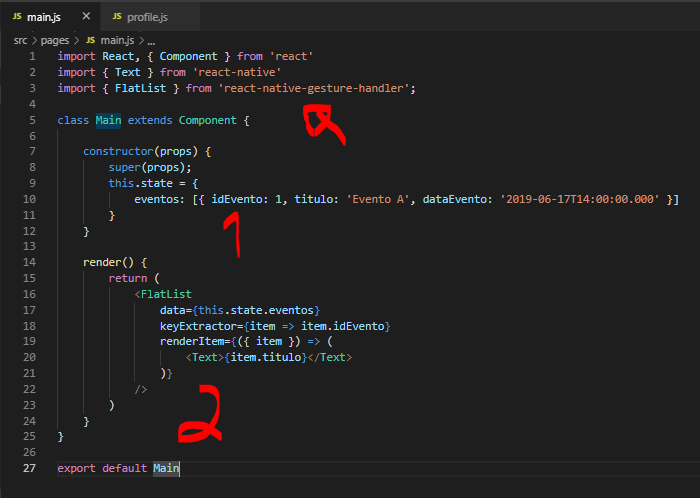
**Eventos**

**Apresentando a lista**

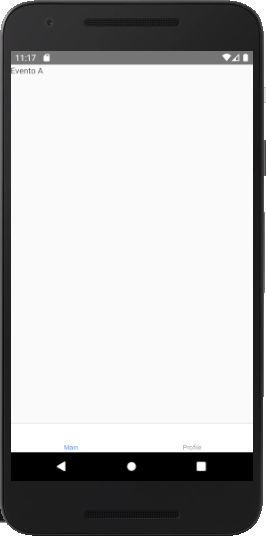
FlatList

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

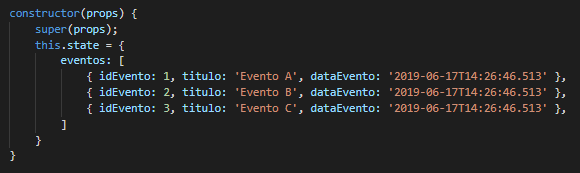
Incluir uma lista fixa e criar uma flatlist. Possui alguns parâmetros: estilo, informação que será mostrada, a chave para cada item da linha, e qual método será chamado para renderizar os itens disponíveis.



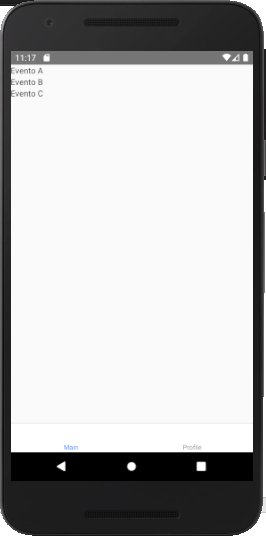
Visualizando o resultado na lista.



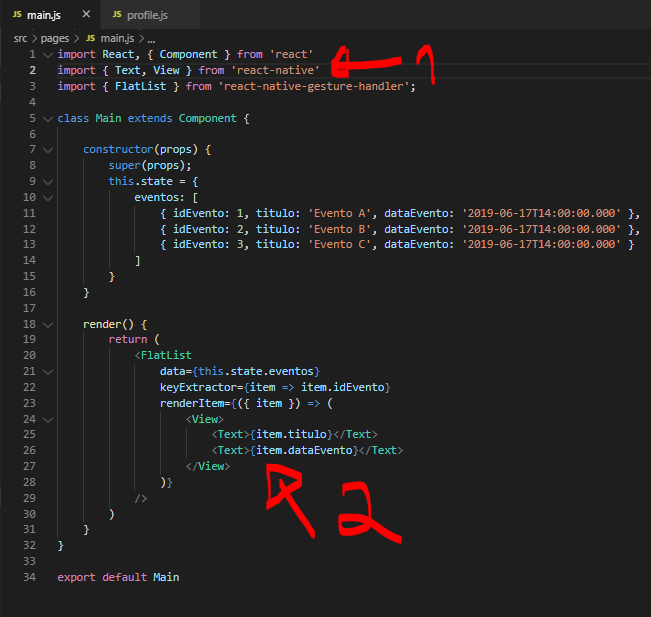
Incluindo mais registros.



Visualizar a apresentação.



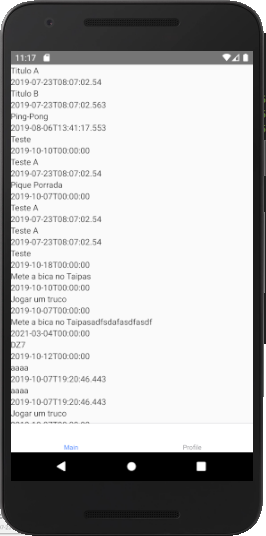
Alterando os itens a serem apresentados.



**Buscando as informações da API**



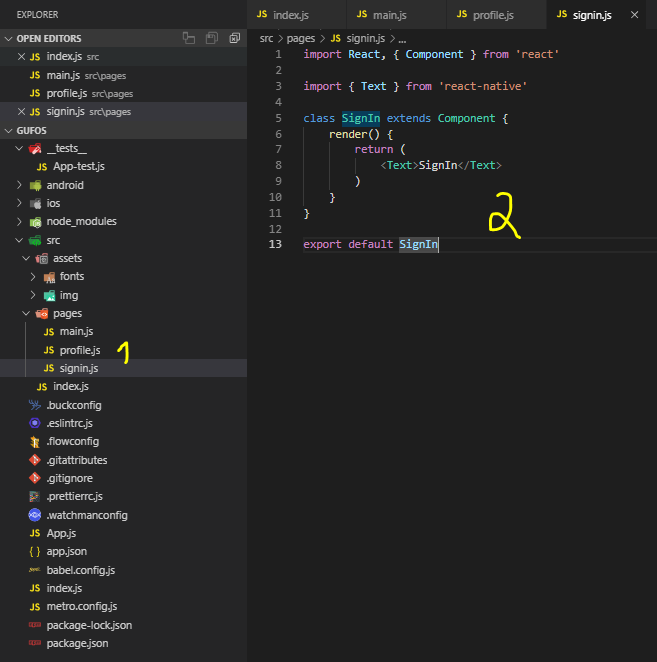
Visualizando os resultados.



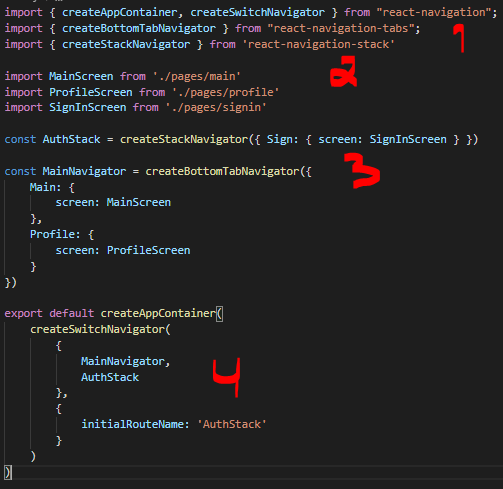
**Incluindo a tela de login**

npm install --save react-navigation-stack

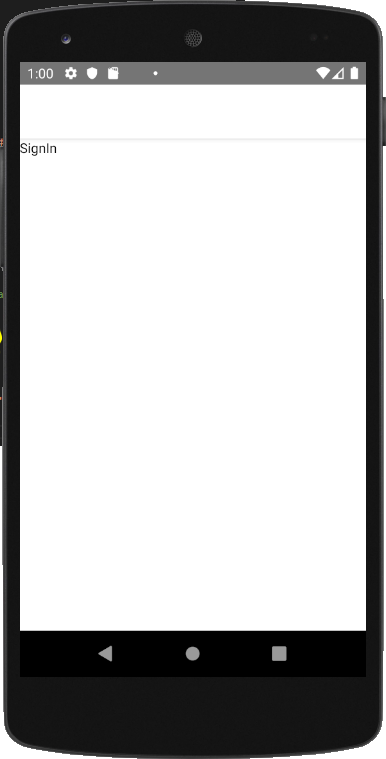
Criar o arquivo signin.js.



Alterar o arquivo index.js de dentro do src.

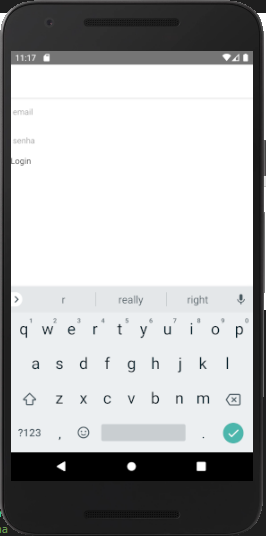


Visualizando o resultado.



Adicionando dois inputs na tela de login(signin.js) para email e senha e um botão para realizar a ação.

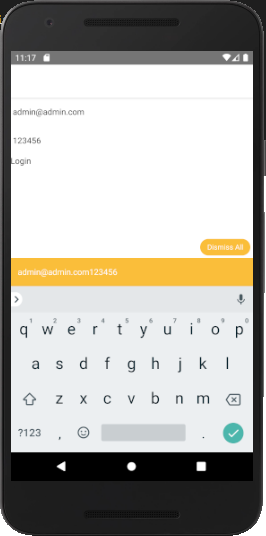




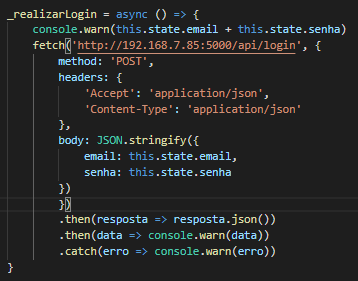
Criando a interação com o usuário.



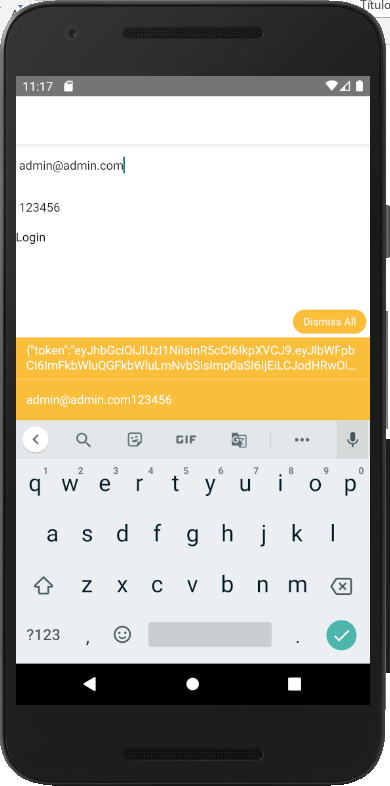
Visualizando o resultado



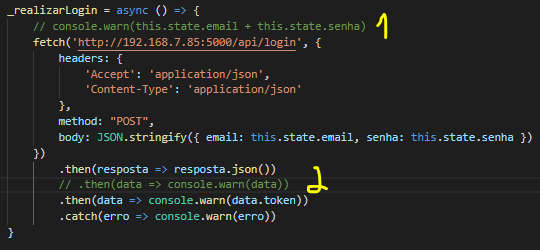
Realizando a chamada à API.



Visualizando o resultado.



Alterando para buscar somente o token.

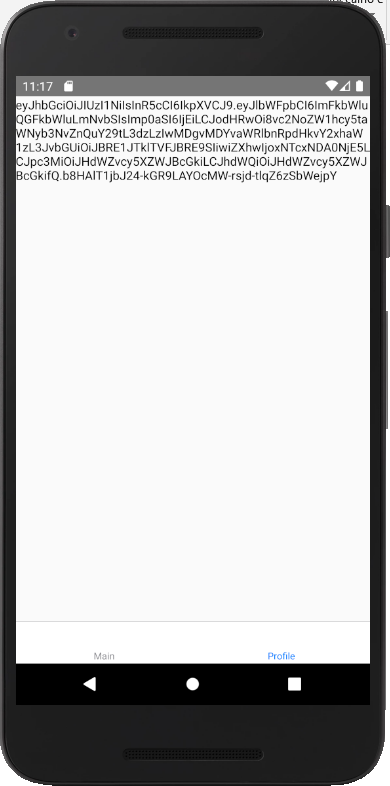


Guardando no AsyncStorage.



Buscando do AsyncStorage



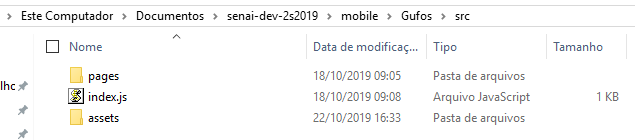


Instalar a dependência para realizar o decode do JWT.

npm install --save jwt-decode

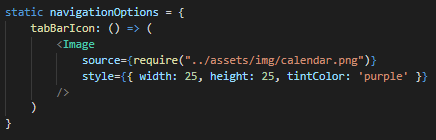
**Como estilizar os itens**

Antes de iniciar, copiar os itens da pasta assets para dentro da pasta src do projeto.



Para alterar os ícones da barra de navegação.

Ir até o main.js e alterar as opções de navegação.



Isto fará com que o ícone da home de eventos seja alterado para o indicado.

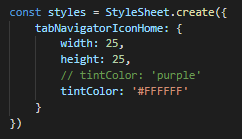


Uma dica: é não trabalhar com o valor inline. E sim, criar um estilo.





E alterar main.js e o profile.js para alterar o estilo dele.



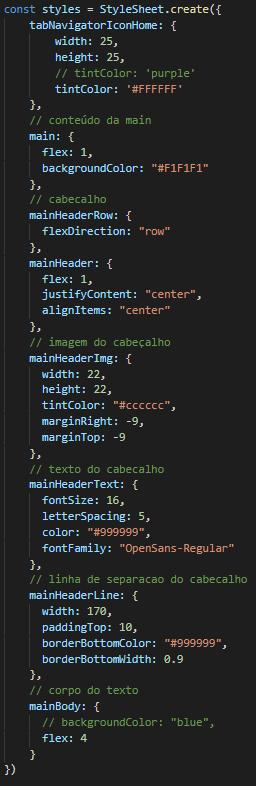
Visualizando o resultado.



Alterar o layout da lista.



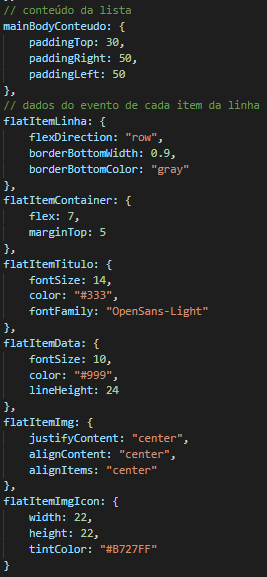
Criar o estilo da main.js. Adicionar os itens que faltam.



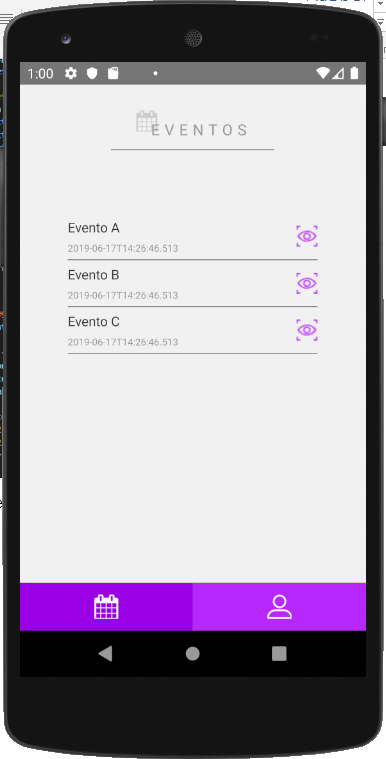
Criar um item customizado.



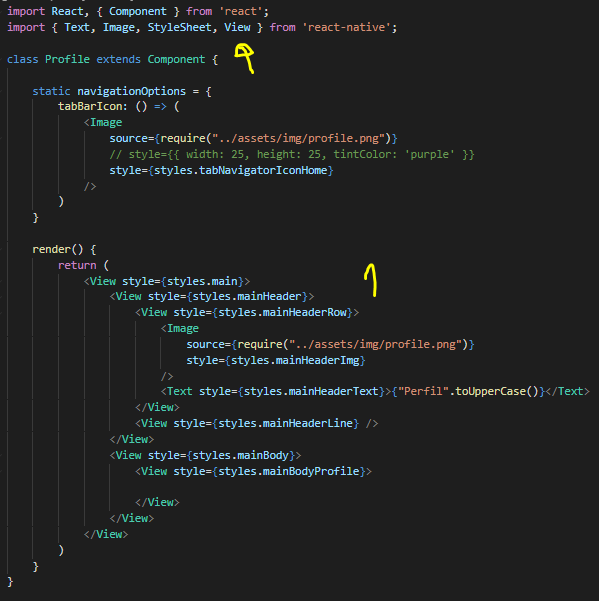
Ajustar layout no styles.



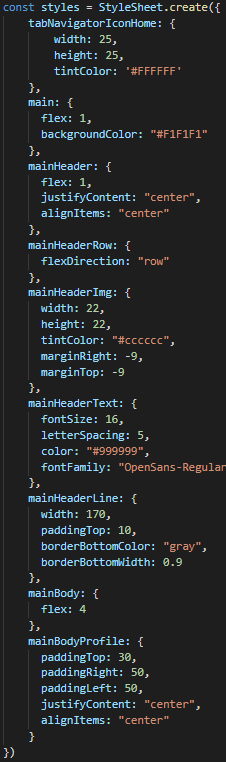
Resultado da lista apresentada com um item customizado.



**Estilizando o profile.js.**



Alterando o estilo do profile.js.





Adicionando o estilo.



Visualizando o resultado.

