**Utilizando a propriedade navigationOptions**

O navigationOptions é uma propriedade que o react-navigation busca em cada rota para ele setar algumas configurações.

**Definindo o título da página Main com o navigationOptions**

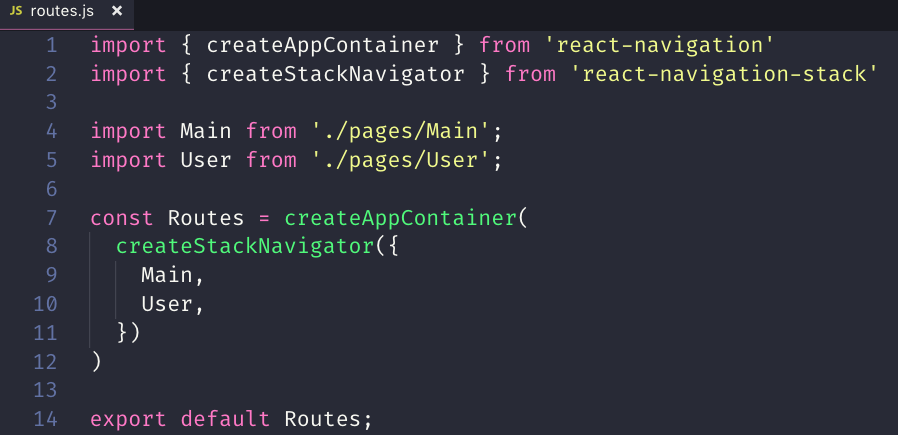
**Edite o arquivo src/pages/Main/index.js**

* Insira as linhas 10, 11 e 12.

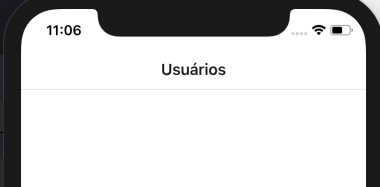


Edite o arquivo src/routes.js e altere o tipo de navegação para createStackNavigator, conforme print a seguir.

* Altere as linhas 2 e 8



Inicie a aplicação digitando react-native run-android e observe que o título usuários no android fica do lado esquerdo, porém no ios fica ao centralizado. Veja abaixo o print da tela no ios.



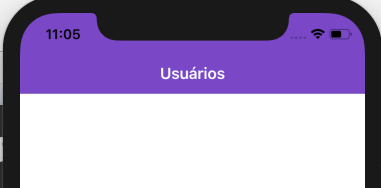
O CreateStackNavigator, aceita um segundo objeto onde você pode definir várias propriedades para o header da aplicação. Veja a seguir.

Insira da linha 11 até 19.



* Na linha 11, por padrão no android, o título fica do lado esquerdo, utilizando o comando headerLayoutPreset o conteúdo fica centralizado.
* Na linha 12 a propriedade headerBackTitleVisible permite que você navegue para outra tela no IOS e oculte a palavra voltar, desde que você defina o conteúdo false. A setinha ficará sempre visível.
* Na linha 13 a propriedade defaultNavigationOptions permite definir configurações padrões em todas as telas.
* Na linha 15 a propriedade backgroundColor define a cor de fundo do header.
* Na linha 17 a propriedade headerTintColor define a cor do titulo do header

Atualize a tela e veja o resultado conforme print a seguir,



**Configurando o status bar**

O Status bar mostra o horário, a bateria. No ios ela está com a cor preta e no android provavelmente com a cor cinza, dependendo do modelo do aparelho.

**Edite o arquivo src/index.js conforme print abaixo.**

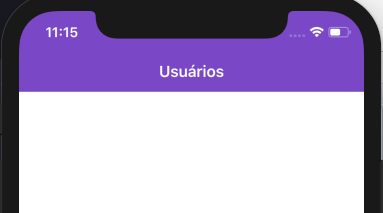
Altere a linha 2

Insira da linha 7 até a 10.

****

* Na linha 2 importa o StatusBar do react-native.
* Na linha 7 cria um fragment para não gerar informação no html.
* Na linha 8 inseri o componente StatusBar e a propriedade barStyle com o conteúdo light-content para definir a cor do texto como branco. A propriedade backgrounColor no android define define a cor do status bar como roxo.

Veja abaixo o resultado.



**Styled Components**

É a lib utilizada para estilizar a página. Para instalá-la digite:

**yarn add styled-components**

**Crie o arquivo src/pages/Main/styles.js**

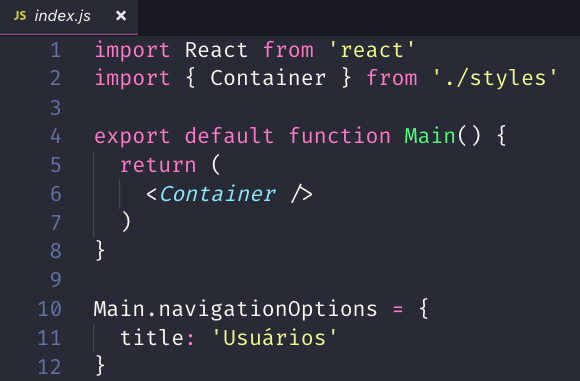
Neste styles.js será configurado o container para a página Main, conforme código a seguir.



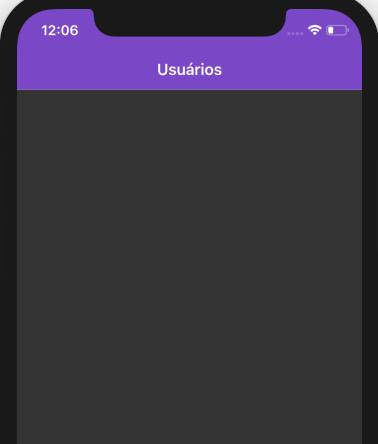
* Na linha 1 importa o styles-components do native e o armazena na variável styled.
* Na linha 3 cria uma função Container com a tag View.
* Na linha 4 define o flex com o valor 1 para que ele ocupe todo o espaço da tela.
* Na linha 5 define o padding como 30px para o espaçamento superior, inferior, esquerda e direita.
* Na linha 6 define a cor de fundo com um cinza mais escuro.

**Edite o arquivo src/pages/Main/index.js**

* Altere a linha 2 para importar o Container do arquivo styles
* Altere a linha 6 para executar o Container.



Veja a seguir o resultado da importação do Container.



**Estilizando o formulário**

Edite o arquivo src/pages/Main/index.js conforme print a seguir.



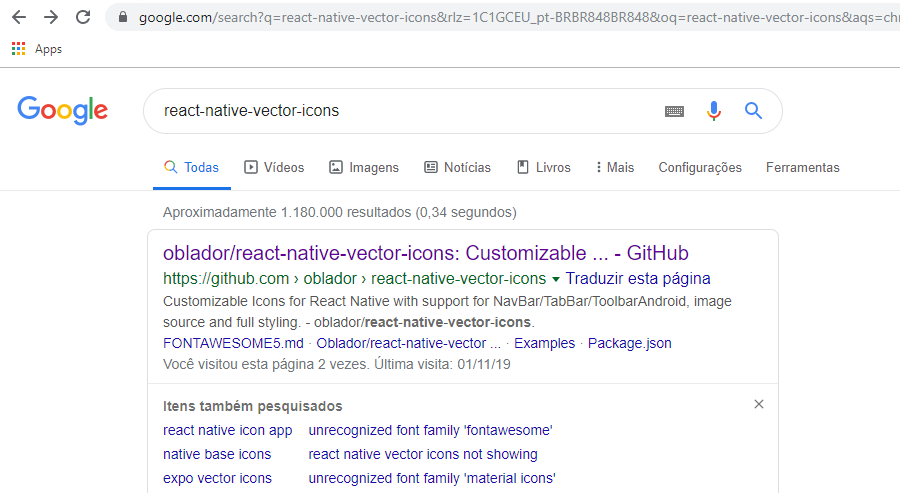
Para definir um button, instale a biblioteca react-native-vector-icons digitando:

**yarn add react-native-vector-icons**

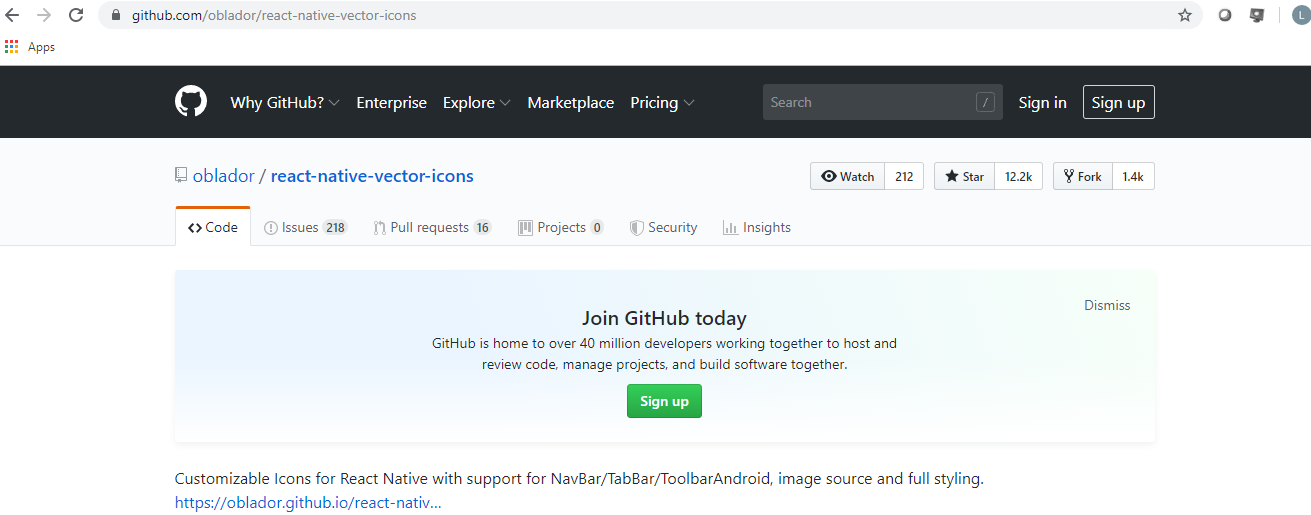
Esta biblioteca serve para instalar ícones em formato de vetor e contém fontes de icones famosos como: materialIcons, fontAwesome e outras.

Para ele funcionar tem que executar o comando:

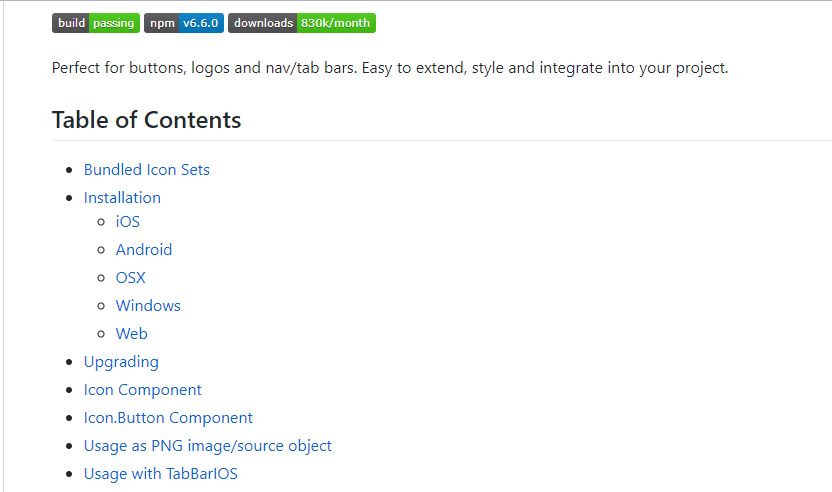
Abra o google e digite react-native-vector-icons conforme print abaixo.



Clique no link da janela acima para abrir o GitHub e será exibida a janela abaixo.

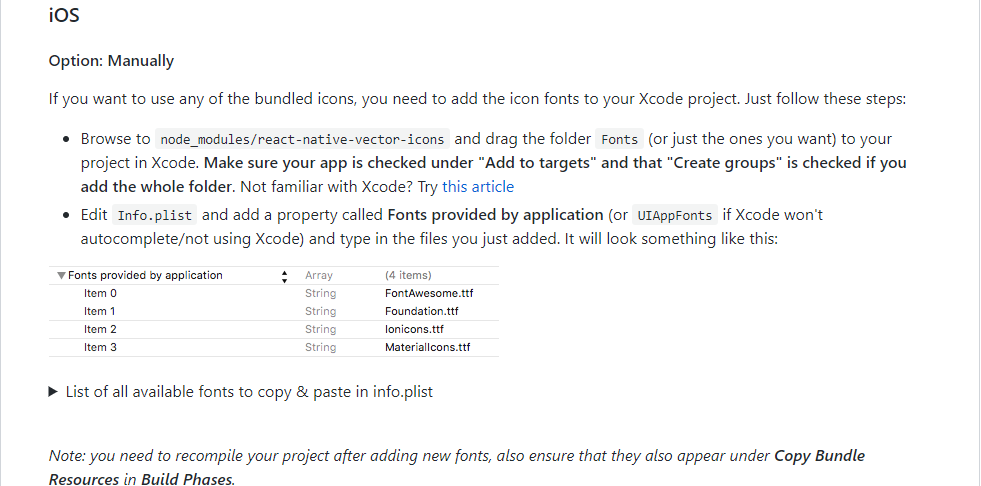


Avance para baixo até visualizer a página a seguir.

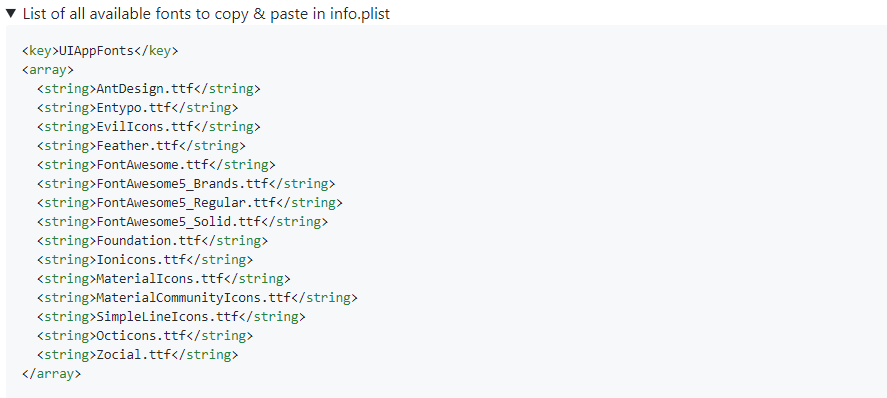


**Configuração do react-native-vector-icons para ios**

Na janela acima, clique no link do IOS e será exbida a janela abaixo.

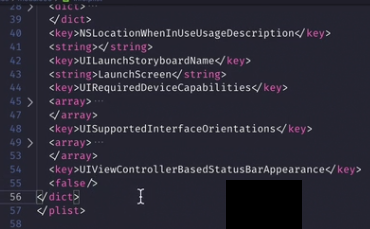
****

Na janela acima clique na seta ao lado esquerdo da **opção List off all available fonts to copy & paste in info.plist** e será exibida a janela abaixo.

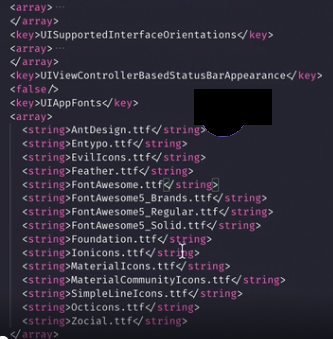


Copie todo o conteúdo e retorne ao VS code execute os passos abaixo:

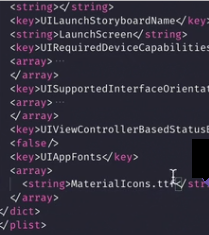
1. Abra a pasta ios
2. Abra a pasta do seu projeto, por exemplo, mobile
3. Abra o arquivo info.plist
4. Vai no final do arquivo antes da tag </dict> (conforme print abaixo)



1. Cole o conteúdo que foi copiado da página do GitHub, conforme print a seguir.

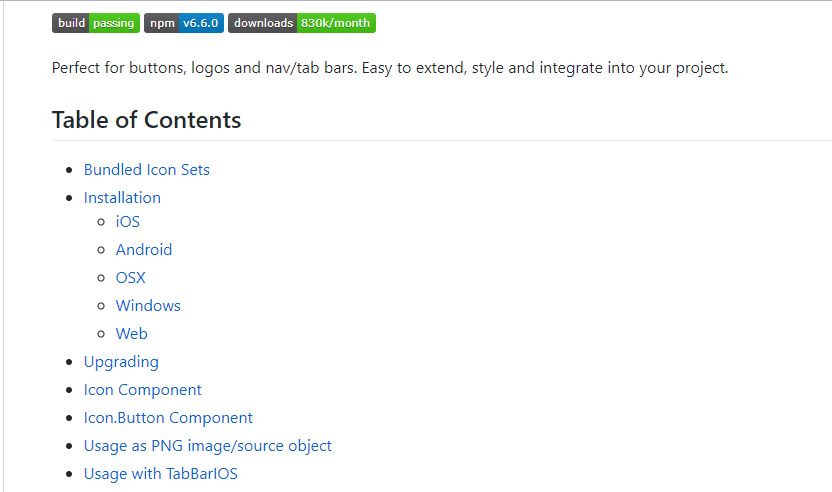


1. Deixe somente a fonte que será utilizada, no exemplo, será a MaterialIcons.ttf, conforme print a seguir.



1. Salve o arquivo
2. No terminal dentro da pasta do projeto, por exemplo, mobile, entre na pasta ios digitando **cd ios** e depois o commando **pod install**
3. Quando o pod install finalizar, retorne a pasta anterior digitando o comando **cd ..** e posteriormente o commando **react-native run-ios**

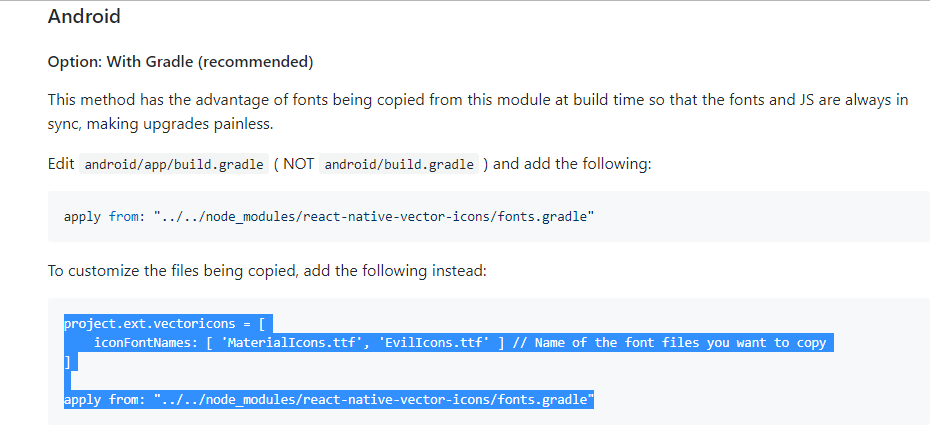
**Configuração do react-native-vector-icons para Android**



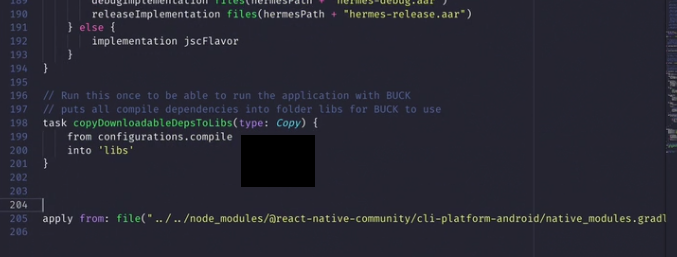
1. Na janela acima, clique no link do Android e será exibida a tela a seguir.

****

1. Selecione e copie o texto que começa com project.ext.vectoricons, conforme print a seguir.



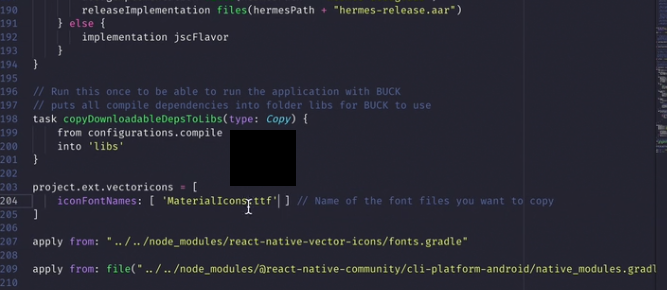
1. Copie todo o conteúdo e retorne ao VS code execute os passos abaixo:
2. Abra a pasta android
3. Abra a pasta app
4. Abra o arquivo build.gradle e vai até o final dele e antes do ultimo apply conforme print abaixo



1. Cole o contéudo que foi copiado da página do GitHub conforme print a seguir.

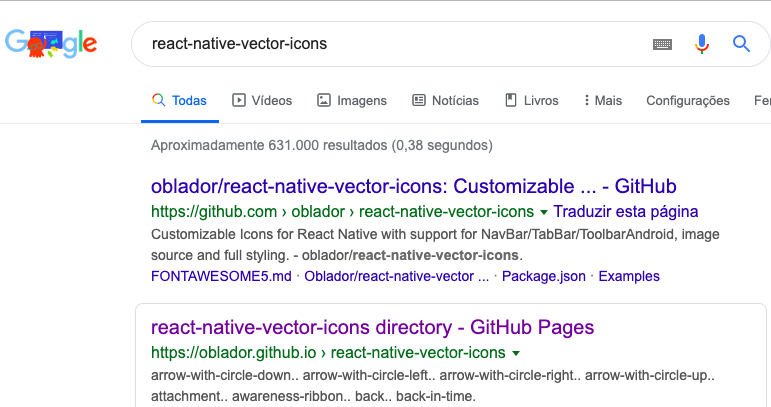


e) Deixe somente a fonte MateriaIcons que será a utilizada pela aplicação, conforme print a seguir.

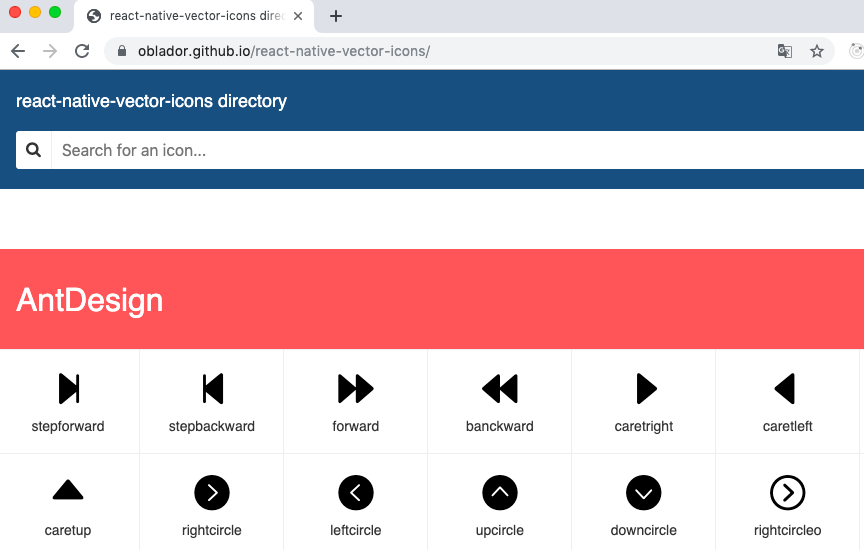


f) Abra o terminal dentro da pasta do projeto, por exemplo, mobile e digite o commando **react-native run-android.**

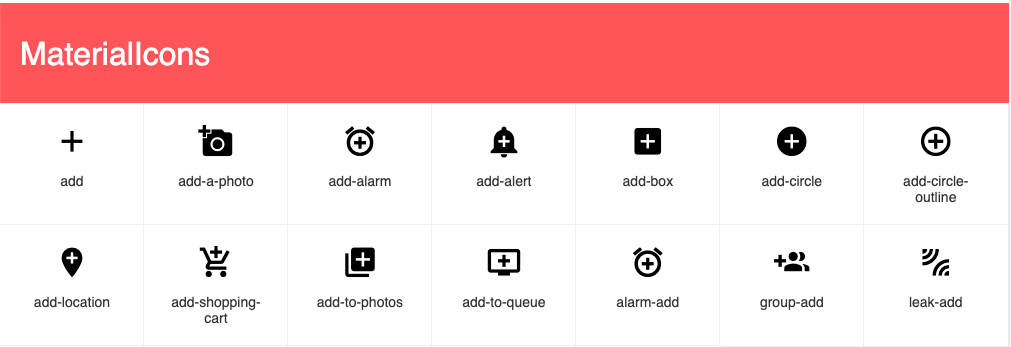
Para visualizar os ícones que podem ser utilizados, no google digite react-native-vector-icons e acesse a página do github.



Clique no link **react-native-vector-icons directory – GitHub pages** e será exibida a janela a seguir.



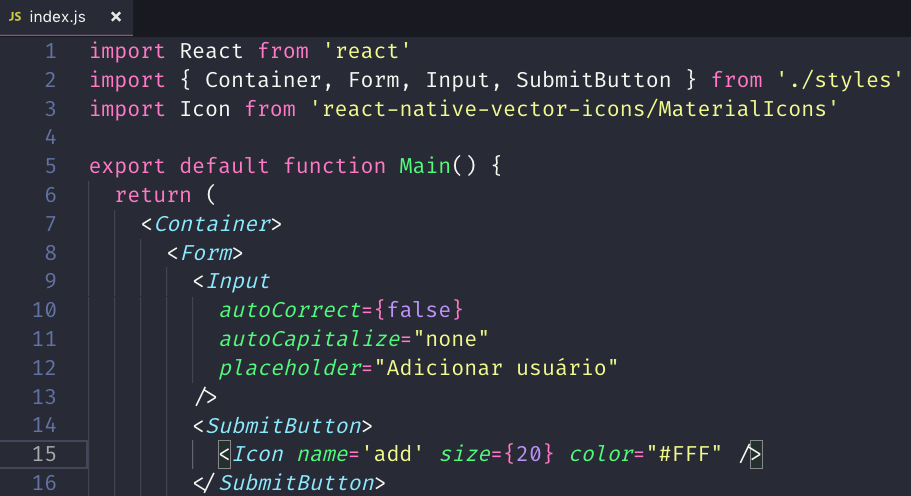
Na janela acima, digite **add** e serão exibidos os ícones de adicionar, localize o add do MaterialIcons, conforme print a seguir.



Copie o texto add e retorne para o Visual Studio Code

Altere a linha 2, incluindo os styles: Form, Input e SubmitButton

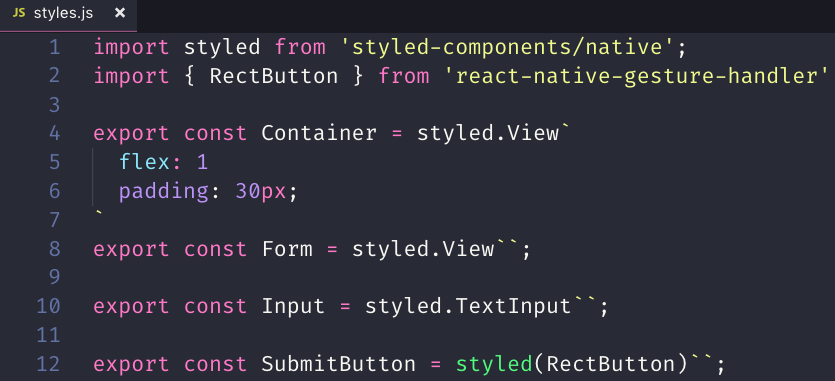
Insira as linhas 3 e 15



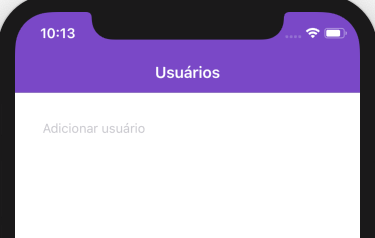
**Criando os estilos (styles) da página Main**

Edite o arquivo src/pages/Main/styles.js conforme print a seguir.

Insira a linha 2 e da linha 8 até 12



Visualize a aplicação no smartphone, a mesma deve estar conforme print a seguir.



**Implementando a estilização**

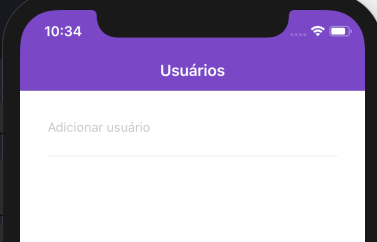
**Edite o arquivo src/pages/Main/styles.js conforme print a seguir.**

**Estilizando o Form**

Insira da linha 9 até a 12

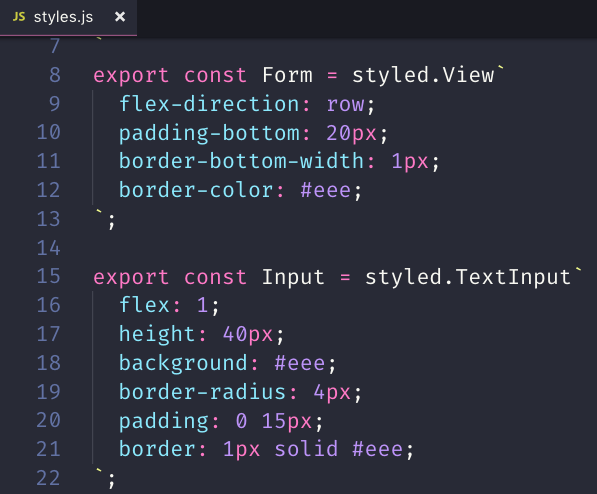


Visualize a aplicação no smartphone e deverá estar conforme print a seguir.

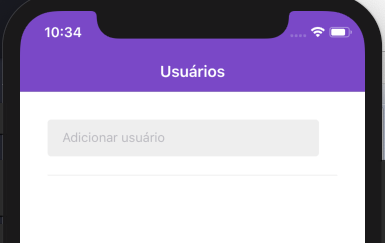


**Estilizando o input**

Insira da linha 16 até 21.



Visualize a aplicação no smartphone e deverá estar conforme print a seguir



**Estilizando o botão**

Insira da linha 25 até 30.



Visualize a aplicação no smartphone e deverá estar conforme print a seguir

