

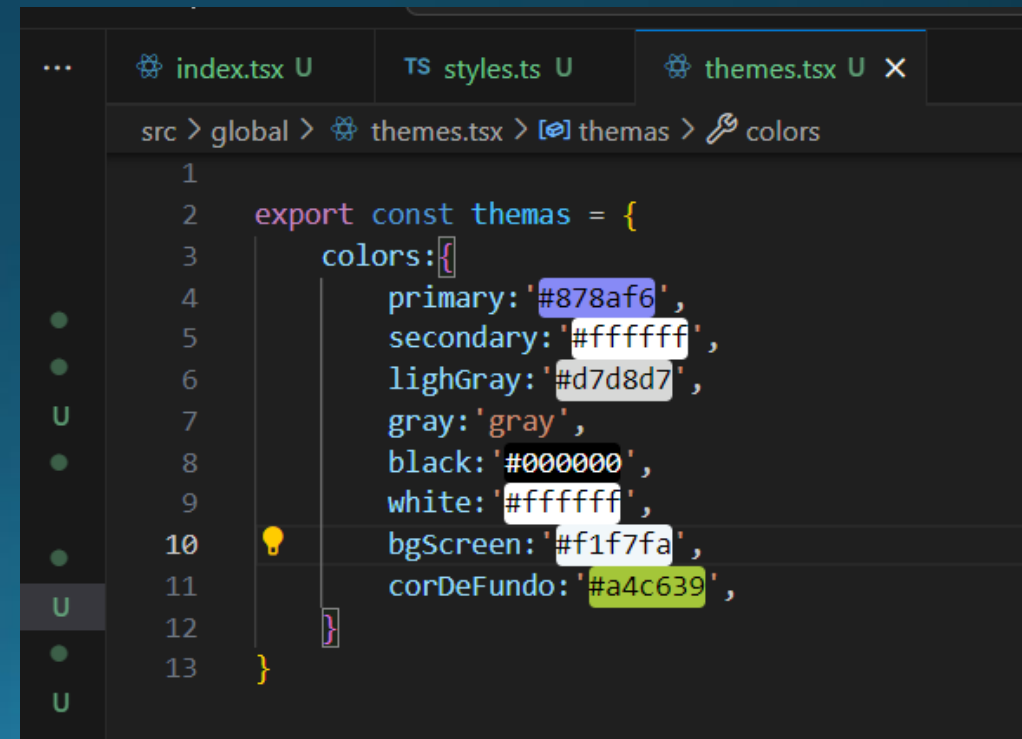
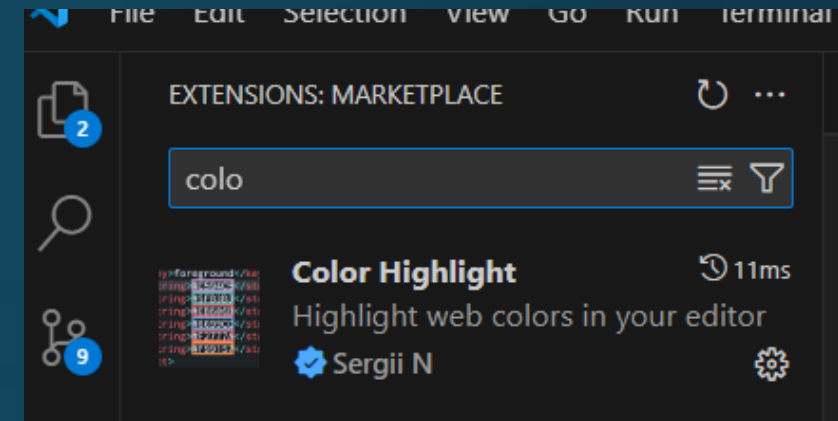
Professor Crispim Luiz Martins

React Native com Expo

parte 02

Criando nossos Temas

- Na pasta global crie a themes.tsx
- Use a extensão para ver as cores →
- Dentro do arquivo themes coloque seus temas









Biblioteca de vetores fontes

- Página = [react-native-vector-icons – npm](#)
- Na página temos a documentação de uso, leia depois
- Comando para execução.
- **npm i react-native-vector-icons**
- Temos várias funções veja ao lado →
- Mas para ver os ícones vá em
- [react-native-vector-icons directory](#)
- Os name dos ícones pegue na biblioteca acima







Table of Contents

- Bundled Icon Sets
- Installation
 - iOS Setup
 - Android Setup
 - macOS Setup
 - Windows Setup
 - React-native-web Setup
 - Web Setup
- Upgrading
- Icon Component
- Icon.Button Component
- Usage as PNG Image/Source Object
- Multi-Style Fonts
- Custom Fonts
- Animation
- Usage Examples
- TabBar
- Generating Your Own Icon Set from a CSS File
- Changelog
- Troubleshooting
- License

Terá de colocar o name do que for usar

| AntDesign | | | |
|---|---|---|--|
|  |  |  | |
| stepforward | stepbackward | forward | |
|  |  |  | |
| caretup | rightcircle | leftcircle | |

Terá de usar o name dos ícones

| AntDesign | | | |
|---|---|---|--|
|  |  |  | |
| stepforward | stepbackward | forward | |
|  |  |  | |
| caretup | rightcircle | leftcircle | |

Colocando ícones no código

- Importe a biblioteca → `import {MaterialIcons} from '@expo/vector-icons';`

```
src > pages > login > index.tsx > Login
1  import React from "react";
2  import {
3    Text,
4    View,
5    Image,
6    TextInput
7  } from 'react-native';
8  import {style} from './styles';
9  import {MaterialIcons} from '@expo/vector-icons';
10 import Logo from '../assets/login-96.png';
11 export default function Login(){
12   return(
```



Inserindo no corpo do código

Estava assim

```
return(  
  <View style={style.container}>  
    <View style={style.boxTop}>  
      <Image source={Logo} />  
      <Text>Bem Vindo!!!</Text>  
    </View>  
    <View style={style.boxMid}>  
      <Text>Endereço de E-mail</Text>  
      <TextInput style={style.boxInput}/>  
      <Text>Coloque sua Senha</Text>  
      <TextInput style={style.boxInput}/>  
    </View>  
    <View style={style.boxBottom}>  
      <Text>Bottom</Text>  
    </View>  
  </View>  
)
```

```
return(  
  <View style={style.container}>  
    <View style={style.boxTop}>  
      <Image source={Logo} />  
      <Text>Bem Vindo!!!</Text>  
    </View>  
    <View style={style.boxMid}>  
      <Text style={style.title}>E-mail</Text>  
      <View style={style.boxInputEmail}>  
        <TextInput style={style.input}/>  
        <MaterialIcons name="email" size={20} color={themas.colors.gray} />  
      </View>  
      <Text style={style.title}>Coloque sua Senha</Text>  
      <View style={style.boxInputSenha}>  
        <TextInput style={style.input}/>  
        <MaterialIcons name="password" size={20} color={themas.colors.gray} />  
      </View>  
    </View>  
    <View style={style.boxBottom}>  
      <Text>Bottom</Text>  
    </View>  
  </View>  
)
```

Coloque mais View para separar melhor os espaços e layout da página

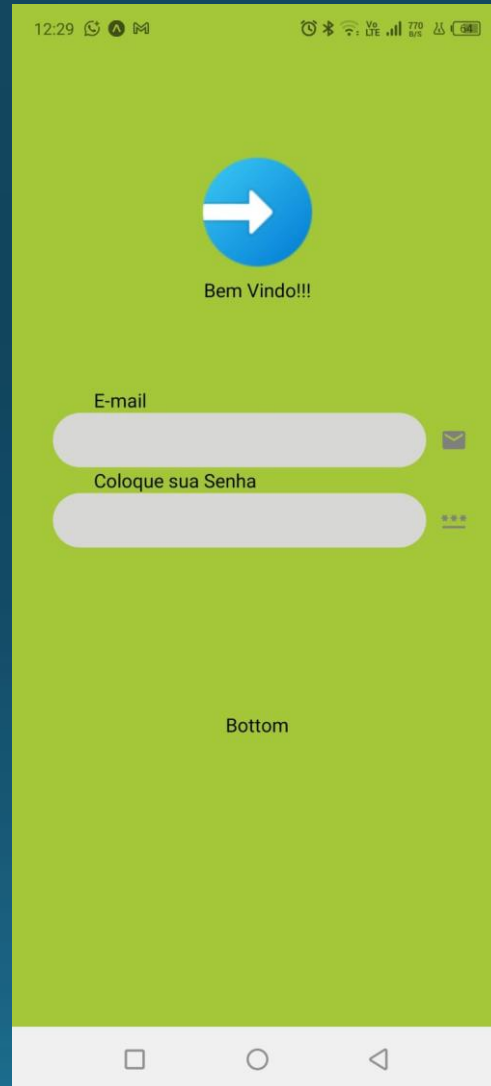
Como ficou meu style

```
index.tsx U X TS styles.ts U X
src > pages > login > TS styles.ts > style > boxBottom

1 import { Dimensions, StyleSheet } from "react-native";
2 import { temas } from "../../global/themes";
3
4 export const style = StyleSheet.create({
5   container:{
6     flex:1,
7     width:"100%",
8     paddingTop:50,
9     justifyContent:"center",
10    backgroundColor:temas.colors.corDeFundo,
11  },
12
13  title:{
14    marginLeft: 40,
15  },
16
17  boxTop:{
18    alignItems: 'center',
19    justifyContent: 'center',
20    height:Dimensions.get("window").height/3,
21    width:"100%"
22  },
23  boxMid:{
24    height:Dimensions.get("window").height/3,
25    width:"100%",
26    marginLeft: 20,
27  },
```

```
28   boxBottom:{
29     alignItems: 'center',
30     height:Dimensions.get("window").height/3,
31     width:"100%"
32   },
33   boxInputEmail:{
34     width:"90%",
35     height:40,
36     backgroundColor:temas.colors.corDeFundo,
37     alignItems: 'center',
38     flexDirection: 'row',
39     justifyContent: 'space-between',
40     paddingHorizontal: 10,
41   },
42   boxInputSenha:{
43     width:"90%",
44     height:40,
45     backgroundColor:temas.colors.corDeFundo,
46     alignItems: 'center',
47     flexDirection: 'row',
48     justifyContent: 'space-between',
49     paddingHorizontal: 10,
50   },
51   input:{
52     borderRadius:40,
53     height:"100%",
54     width:"90%",
55     backgroundColor: temas.colors.lighGray,
56   }
57 })
```

Resultado



Criando nosso button

```
<View style={style.boxBottom}>
  <TouchableOpacity style={style.button}>
    <Text>Entrar</Text>
  </TouchableOpacity>
</View>
```

```
<View style={style.container}>
  <View style={style.boxTop}>
    <Image source={Logo} />
    <Text>Bem Vindo!!!</Text>
  </View>
  <View style={style.boxMid}>
    <Text style={style.title}>E-mail</Text>
    <View style={style.boxInputEmail}>
      <TextInput style={style.input}/>
      <MaterialIcons name="email" size={20} color={themas.colors.gray} />
    </View>
    <Text style={style.title}>Coloque sua Senha</Text>
    <View style={style.boxInputSenha}>
      <TextInput style={style.input}/>
      <MaterialIcons name="password" size={20} color={themas.colors.gray} />
    </View>
  </View>
  <View style={style.boxBottom}>
    <TouchableOpacity style={style.button}>
      <Text>Entrar</Text>
    </TouchableOpacity>
  </View>
</View>
```

Style Button

```
boxBottom:{  
  marginTop:-100,  
  width:"100%",  
  alignItems: 'center',  
  justifyContent: 'center',  
  height:Dimensions.get("window").height/3,  
},  
button:{  
  width:200,  
  height:40,  
  alignItems: 'center',  
  justifyContent: 'center',  
  borderRadius:40,  
  backgroundColor:themas.colors.primary,  
}
```

Criando as variáveis

```
1 import React, { useState } from "react";
```

```
2 import { View, Text, TextInput, MaterialIcons }
```

```
13 export default function Login(){  
14   const [email, setEmail] = useState('');  
15   const [senha, setSenha] = useState('');  
16  
17   return(  
18     <View style={style.boxMid}>  
19       <Text style={style.title}>E-mail</Text>  
20       <View style={style.boxInputEmail}>  
21         <TextInput  
22           style={style.input}  
23           value={email}  
24           onChangeText={(e) => setEmail(e)} />  
25         <MaterialIcons name="email" size={20} color={themas.colors.gray} />  
26       </View>  
27       <Text style={style.title}>Coloque sua Senha</Text>  
28       <View style={style.boxInputSenha}>  
29         <TextInput style={style.input}  
30           value={senha}  
31           onChangeText={(e) => setSenha(e)}  
32           secureTextEntry={true} />  
33         <MaterialIcons name="password" size={20} color={themas.colors.gray} />  
34       </View>  
35     </View>  
36   );  
37 }
```

```
<View style={style.boxMid}>  
  <Text style={style.title}>E-mail</Text>  
  <View style={style.boxInputEmail}>  
    <TextInput  
      style={style.input}  
      value={email}  
      onChangeText={(e) => setEmail(e)} />  
    <MaterialIcons name="email" size={20} color={themas.colors.gray} />  
  </View>  
  <Text style={style.title}>Coloque sua Senha</Text>  
  <View style={style.boxInputSenha}>  
    <TextInput style={style.input}  
      value={senha}  
      onChangeText={(e) => setSenha(e)}  
      secureTextEntry={true} />  
    <MaterialIcons name="password" size={20} color={themas.colors.gray} />  
  </View>  
</View>
```

Tem que buscar na biblioteca.

O useState é um hook do React que permite que você adicione e gerencie o estado em componentes funcionais.

Vamos colocar as variáveis no código

Função completa no próximo slide.

`value={email}` ← Defino o valor atual de entrada.

`onChangeText={(e) => setEmail(e)}`

`onChangeText` é uma propriedade do `TextInput` que detecta quando o texto no campo de entrada muda.

```
13 export default function Login(){
14
15   return(
16     <View style={style.container}>
17       <View style={style.boxTop}>
18         <Image source={Logo} />
19         <Text>Bem Vindo!!!</Text>
20       </View>
21       <View style={style.boxMid}>
22         <Text style={style.title}>E-mail</Text>
23         <View style={style.boxInputEmail}>
24           <TextInput
25             style={style.input}
26             value={email}
27             onChangeText={(e) => setEmail(e)} />
28           <MaterialIcons name="email" size={20} color={themas.colors.gray} />
29         </View>
30         <Text style={style.title}>Coloque sua Senha</Text>
31         <View style={style.boxInputSenha}>
32           <TextInput style={style.input}
33             value={senha}
34             onChangeText={(e) => setSenha(e)}
35             secureTextEntry={true} />
36           <MaterialIcons name="password" size={20} color={themas.colors.gray} />
37         </View>
38       </View>
39       <View style={style.boxBottom}>
40         <TouchableOpacity style={style.button}>
41           <Text>Entrar</Text>
42         </TouchableOpacity>
43       </View>
44     </View>
45   )
46 }
47
48 }
```

Resultado

Não muda a tela

