

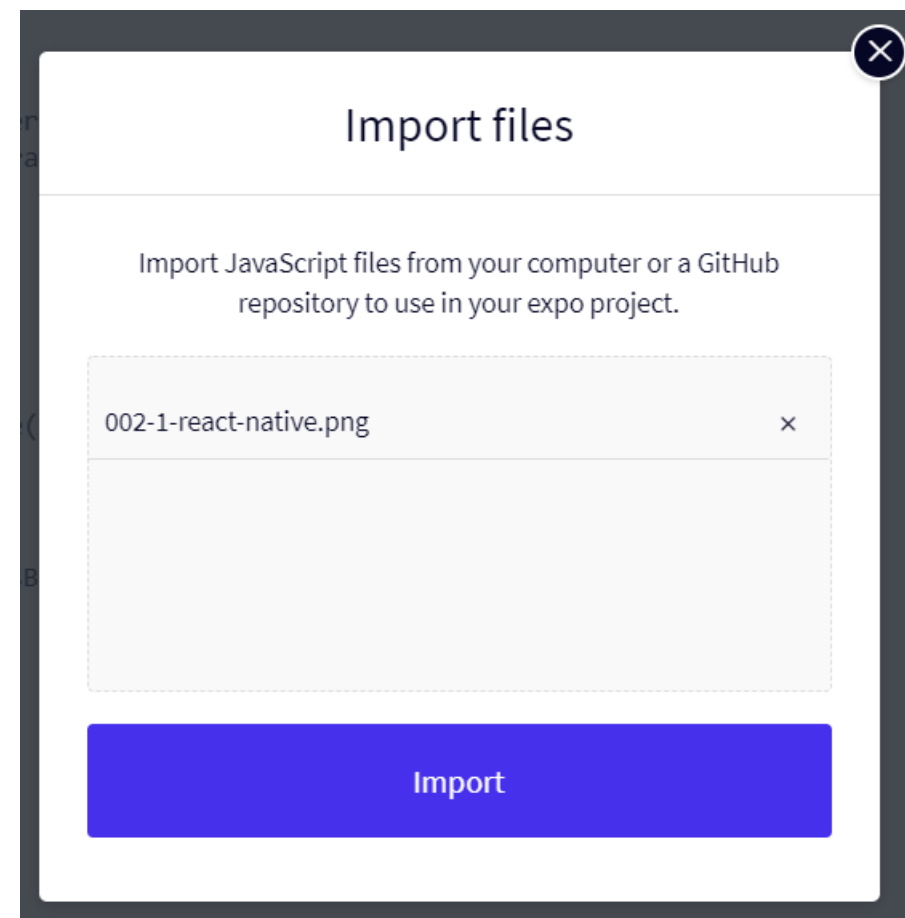
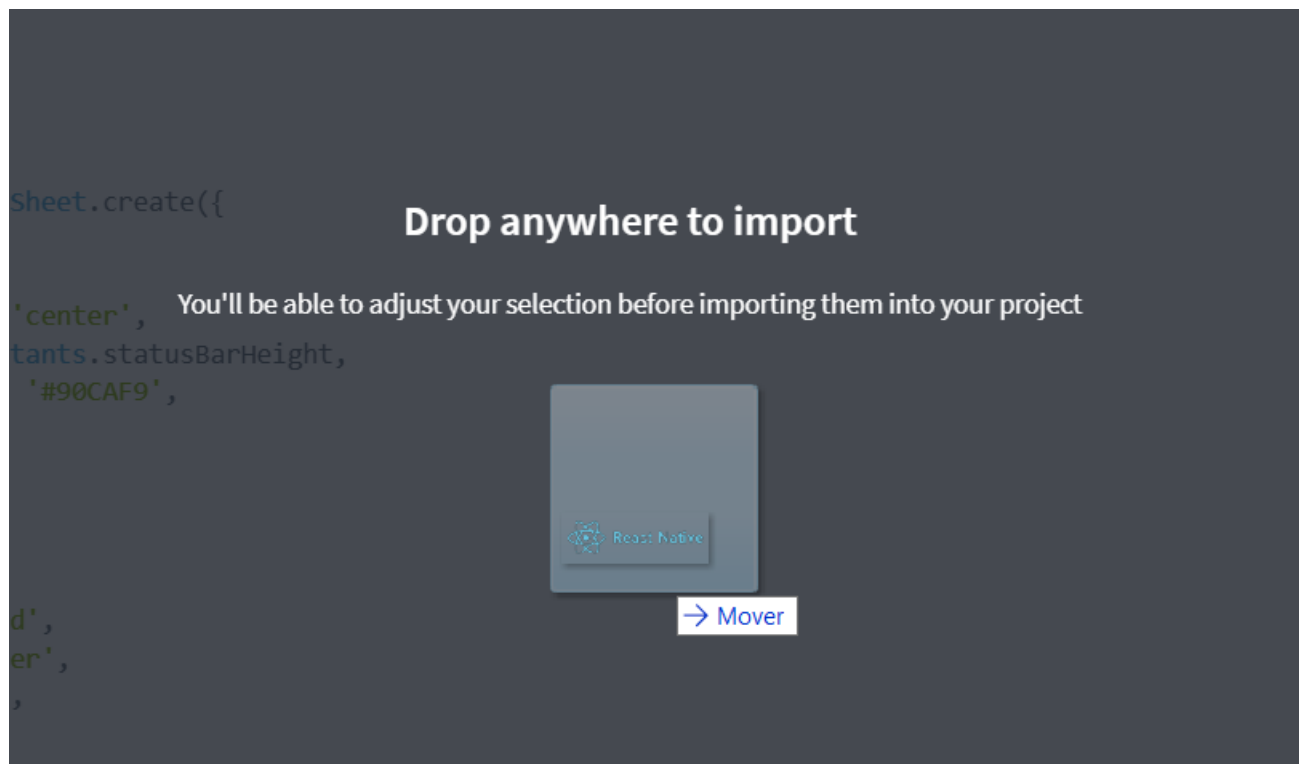


Image e Props

Ewerton J. Silva

ewerton.silva41@etec.sp.gov.br

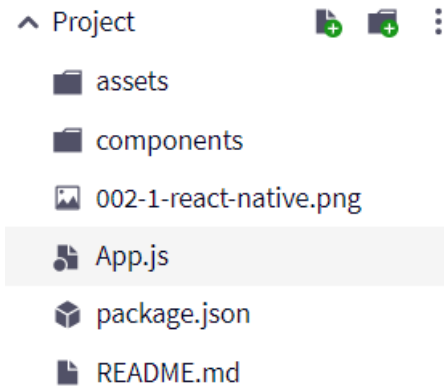
Crie um novo snack para este exemplo, nele iremos utilizar uma imagem no cabeçalho, para isso escolha uma imagem do seu computador, em seguida clique e arraste-a para o navegador com o snack e solte, clique em “import”...



...aguarde o término do processo, o arquivo importado deve aparecer na pasta do projeto, em seguida mova o arquivo para a pasta “assets”, para isso basta selecioná-lo e arrasta-lo para a pasta, deixando assim nosso aplicativo mais organizado, pois teremos “components” utilizada para os componentes “.js” e “assets” para outros tipos de arquivos.

1 file imported successfully

[Dismiss](#)



Open files

002-1-react-native.png

App.js

Project

assets

002-1-react-native.png

snack-icon.png

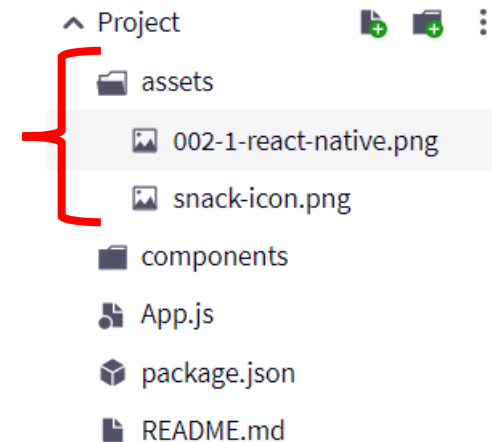
components

002-1-react-native.png

App.js

package.json

README.md



Criando componente principal

Dentro da pasta “components” crie um novo arquivo com o nome “Index.js” e insira o código apresentado ao lado.

Este componente vai retornar uma View com um Text dentro.

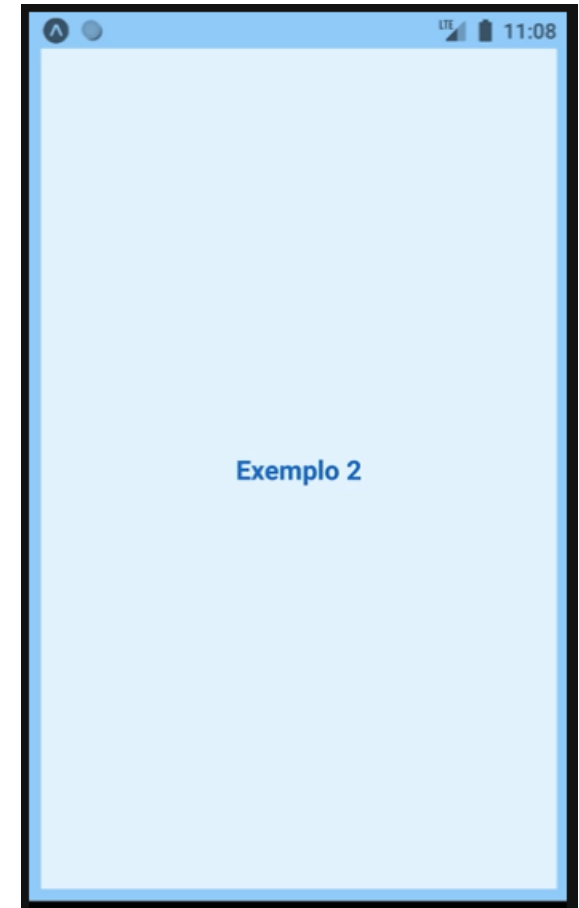


```
1  import * as React from 'react';
2  import { Text, View, StyleSheet } from 'react-native';
3
4  export default function Index() {
5    return (
6      <View style={styles.container}>
7        <Text style={styles.paragraph}>
8          Exemplo 2
9        </Text>
10     </View>
11   );
12 }
13
14 const styles = StyleSheet.create({
15   container: {
16     flex: 1,
17     justifyContent: 'center',
18     backgroundColor: '#E3F2FD',
19     padding: 8,
20   },
21   paragraph: {
22     margin: 24,
23     fontSize: 18,
24     fontWeight: 'bold',
25     textAlign: 'center',
26     color: '#1565C0',
27   },
28 });
```

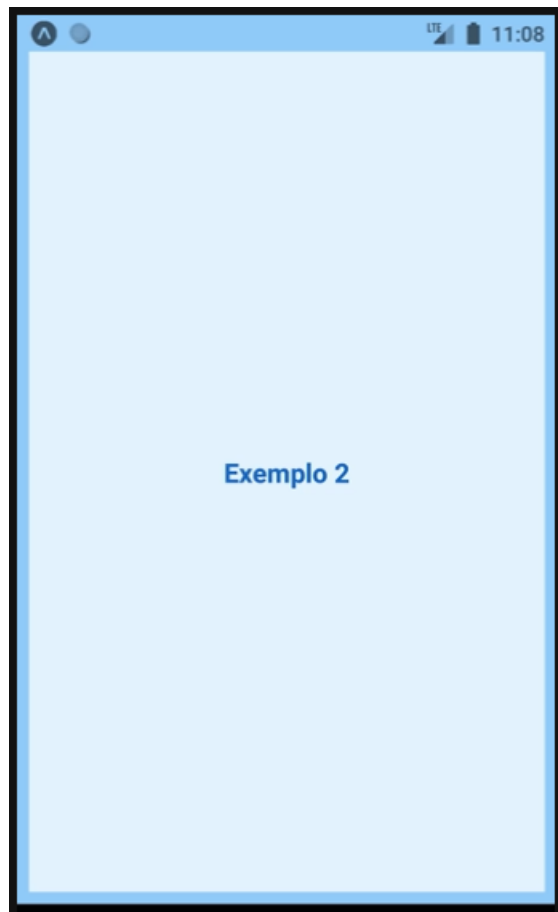
Em “App.js”, vamos importar o componente “Index.js” e utilizá-lo dentro do “App.js”. Em nossas aulas iremos manter sempre esse padrão sempre que iniciarmos um novo projeto, limpando o código e chamando um componente criado para realizarmos nossos exemplos.

Deixe o código de “App.js” conforme o exemplo ao lado, aqui iremos retornar uma View com o componente criado anteriormente. Observe a borda, e veja que temos dois componentes renderizados em tela.

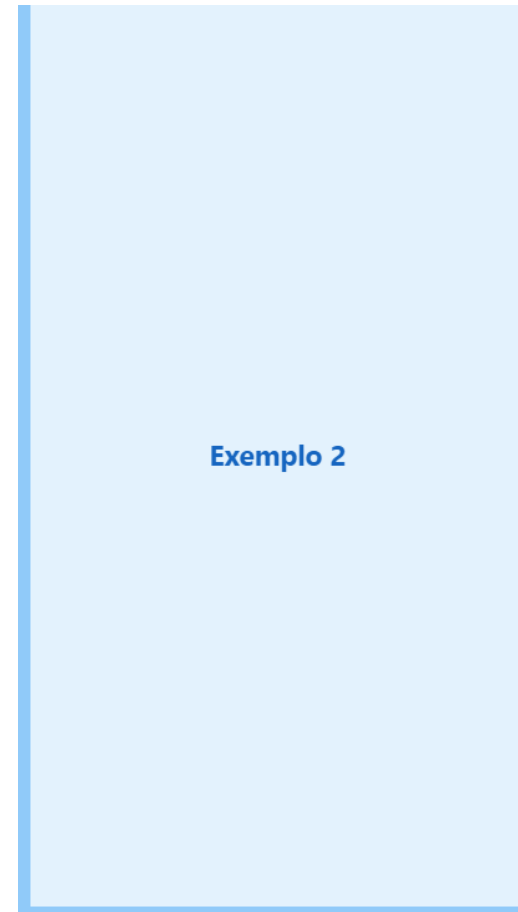
```
1  import * as React from 'react';
2  import { View, StyleSheet } from 'react-native';
3  import Constants from 'expo-constants';
4
5  import Index from './components/Index';
6
7  export default function App() {
8    return (
9      <View style={styles.container}>
10        <Index />
11      </View>
12    );
13  }
14
15  const styles = StyleSheet.create({
16    container: {
17      flex: 1,
18      justifyContent: 'center',
19      paddingTop: Constants.statusBarHeight,
20      backgroundColor: '#90CAF9',
21      padding: 8,
22    },
23  });
```



Visualização Android X Web



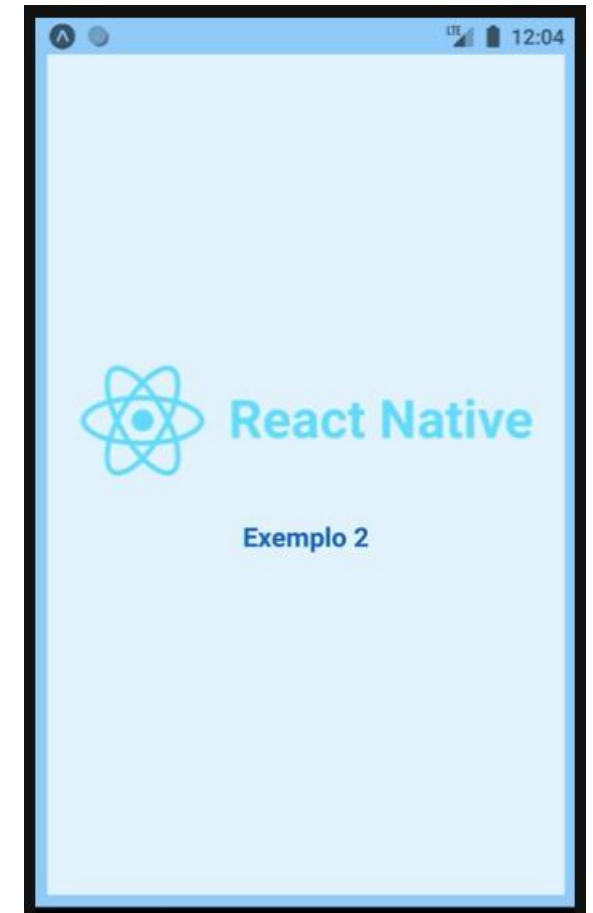
?



Inserindo imagem

Vamos inserir a imagem que adicionamos recentemente ao projeto, para isso será necessário importar o componente e a imagem, para depois inseri-la dentro do componente.

```
1  import * as React from 'react';
2  import { Text, View, StyleSheet, Image } from 'react-native';
3
4  import logo from '../assets/002-1-react-native.png';
5
6  export default function Index() {
7    return (
8      <View style={styles.container}>
9        <Image source={logo} style={{ width: '100%', height: '15%' }} />
10       <Text style={styles.paragraph}>
11         Exemplo 2
12       </Text>
13     </View>
14   );
15 }
```



Estilizando a imagem

No slide anterior fizemos a estilização diretamente da declaração da imagem, para simplificar a estrutura iremos definir um objeto de estilo para a imagem, para isso declare o nome do objeto e defina suas propriedades conforme apontado nas imagens.

No exemplo utilizei porcentagem na altura e largura, também é possível passar esses valores em pixels, além disso em imagens também é possível referenciar URLs, para ver outros exemplos clique no link abaixo:

<https://docs.expo.io/tutorial/image/>

```
17  const styles = StyleSheet.create({
18    container: {
19      flex: 1,
20      justifyContent: 'center',
21      backgroundColor: '#E3F2FD',
22      padding: 8,
23    },
24    paragraph: {
25      margin: 24,
26      fontSize: 18,
27      fontWeight: 'bold',
28      textAlign: 'center',
29      color: '#1565C0',
30    },
31    image: {
32      width: '90%',
33      height: '15%',
34      margin: 10,
35    },
36  });
```



```
<Image source={logo} style={styles.image} />
```


Uso de componentes repetitivos

Conforme apresentado no exemplo abaixo, podemos utilizar o mesmo componente diversas vezes no leiaute, porém existem repetições que podem ser evitadas, como a definição da formatação do objeto exibido.

Já sabemos que é possível definir nossos próprios componentes e apresenta-los da maneira que achamos mais viável, assim...



```
16 export default function Index() {
17   return (
18     <View style={styles.container}>
19       <Image source={logo} style={styles.image} />
20       <Text style={styles.paragraph}>
21         Exemplo 2
22       </Text>
23
24       /* lista bem vindo - ex. comentário*/
25       <Text style={{textAlign: 'center'}}>
26         Olá Mario!
27       </Text>
28       <Text style={{textAlign: 'center'}}>
29         Olá Maria!
30       </Text>
31       <Text style={{textAlign: 'center'}}>
32         Olá Bruna!
33       </Text>
34       <Text style={{textAlign: 'center'}}>
35         Olá Bruno!
36       </Text>
37
38     </View>
39   );
40 }
```

Componente, parâmetros e propriedades

...podemos criar um componente, dentro deste que estamos trabalhando, apenas para apresentar uma caixa de texto personalizada.

Fazemos isto armazenando a saída deste componente em uma variável através de uma arrow function (veremos este assunto mais adiante), com um parâmetro.

Este parâmetro é chamado de “props” e é utilizado para a passagem de valores na chamada de um componente.

Insira o código ao lado no local apontado!

```
1  import * as React from 'react';
2  import { Text, View, StyleSheet, Image } from 'react-native';
3
4  import logo from '../assets/002-1-react-native.png';
5
6  const Saudacoes = (props) => {
7    return(
8      <Text style={{textAlign: 'center'}}>
9        Olá {props.name}!
10      </Text>
11    );
12  }
13
14  export default function Index() {
```



Código JS sempre entre {}

Agora para exibir os nomes será necessário apenas chamar o componente criado e passar o nome como parâmetro.

```
16 export default function Index() {
17   return (
18     <View style={styles.container}>
19       <Image source={logo} style={styles.image} />
20       <Text style={styles.paragraph}>
21         Exemplo 2
22       </Text>
23
24       {
25         <Saudacoes name='Mario' />
26         <Saudacoes name='Maria' />
27         <Saudacoes name='Bruna' />
28         <Saudacoes name='Bruno' />
29       }
30     </View>
31   );
}
```



React Native

Exemplo 2

Olá Mario!
Olá Maria!
Olá Bruna!
Olá Bruno!

Atividade

Deixe o exemplo 2 conforme o modelo ao lado, onde o alinhamento está de cima para baixo e não centralizado, além disso há espaçamento entre os nomes e uma cor aplicada. Compartilhe o link e o print da tela em um arquivo de texto.



Referências

- <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Fun%C3%A7%C3%B5es>
- <https://docs.expo.io/tutorial/image/>
- <https://pt-br.reactjs.org/docs/components-and-props.html>
- <https://reactnative.dev/docs/tutorial>
- <https://material.io/archive/guidelines/style/color.html#>
- <https://reactnative.dev/docs/image>
- <https://www.npmjs.com/package/react-native-responsive-fontsize>