

Estácio

Campus: Rua Manoel João Gonçalves, 410/412 – Alcântara CEP: 24711-080

Curso: Desenvolvimento Full-Stack

Disciplina: RPG0023 - Vamos criar um App

Turma: 9001

Semestre letivo: 2023.5 FLEX

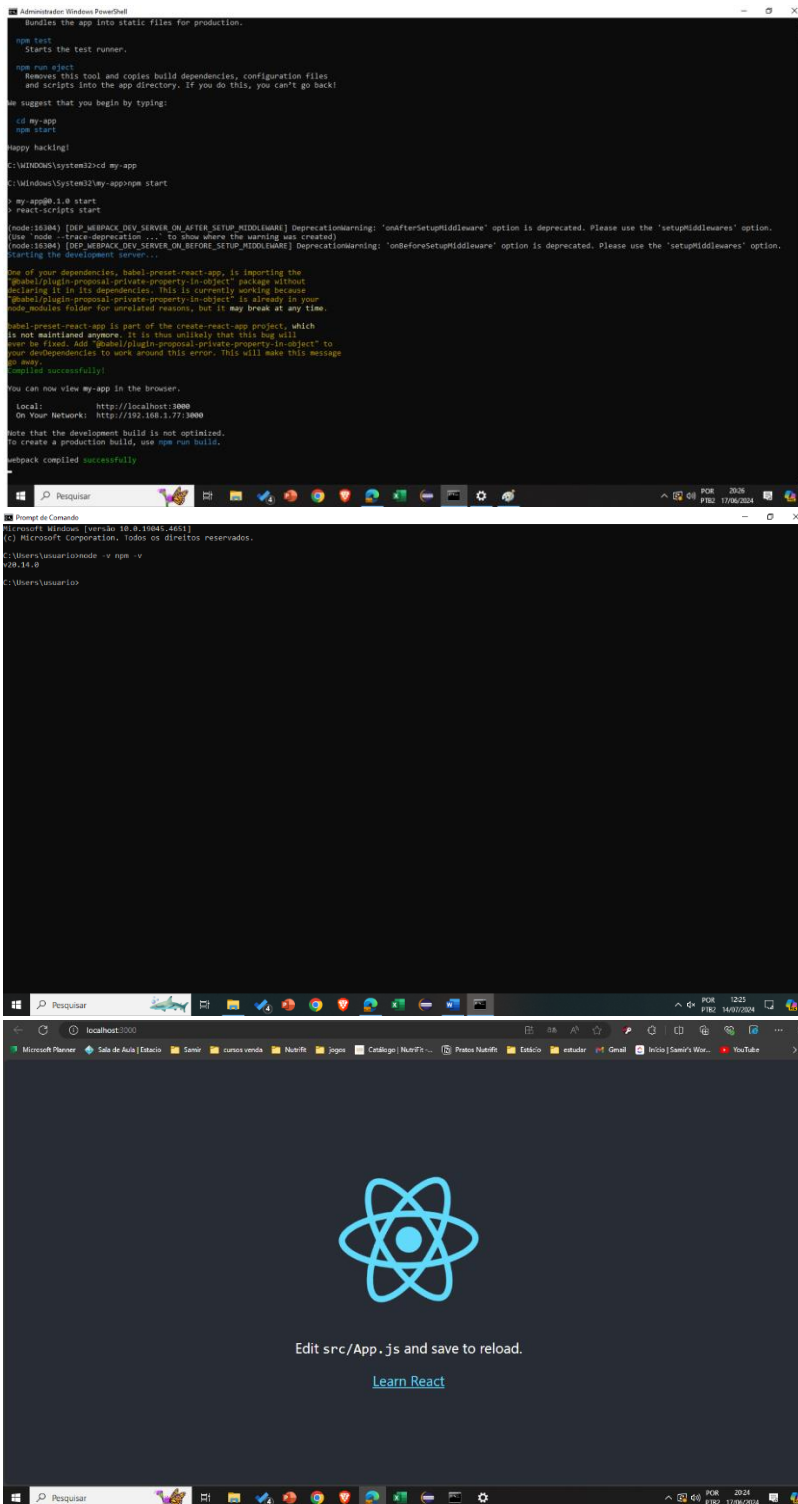
Integrante:

Nome: Samir Campos Lima

Matrícula: 2022.11.47141-1

Link do repositório no GIT [samircamposlima/Miss-o-Pr-tica-N-vel-1-Mundo-4: Missão Prática | Nível 1 | Mundo 4 \(github.com\)](https://github.com/samircamposlima/Miss-o-Pr-tica-N-vel-1-Mundo-4)

Microatividade 1: Configurar o ambiente de desenvolvimento React Native



```
Administrator Windows PowerShell

Bundles the app into static files for production.

npm test
  Starts the test runner.

npm run eject
  Removes this tool and copies build dependencies, configuration files
  and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd my-app
  npm start

Happy hacking!
C:\Windows\System32>cd my-app
C:\Windows\System32>npm start

> my-app@0.1.0 start
> react-scripts start

(node:16384) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
(node:16384) --trace-deprecation ...
(node:16384) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
Starting the development server...

One of your dependencies, babel-preset-react-app, is importing the
@babel/plugin-proposal-private-property-in-object package without
including it in its dependencies. This is currently working because
@babel/plugin-proposal-private-property-in-object is already in your
node_modules folder for unrelated reasons, but it may break at any time.

babel-preset-react-app is part of the create-react-app project, which
is not maintained anymore. It is thus unlikely that this bug will
ever be fixed. Add @babel/plugin-proposal-private-property-in-object to
your dependencies to work around this error. This will make this message
go away.

Compiled successfully!

You can now view my-app in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.1.77:3000

Note that the development build is not optimized.
To create a production build, use npm run build.


webpack compiled successfully

C:\Users\usuario>

Microsoft Corporation. Todos os direitos reservados.
C:\Users\usuario>node -v
v20.14.0
C:\Users\usuario>
```

localhost:3000

Microsoft Planner Sala de Aula [Edição] Semir cursos venda Nuffit jogos Catálogo Nuffit... Preços Nuffit Estácio estudar Gmail Início [Santia's Wor... YouTube



Edit src/App.js and save to reload.

[Learn React](#)

Microatividade 2: Implementar a funcionalidade de entrada de texto em um componente React Native

- Procedimentos

1. Abra o seu editor de código (por exemplo, VS Code).
2. Crie um novo projeto React Native ou utilize um projeto existente.
3. Crie um TextInput para inserir um texto;
4. A função a ser chamada toda vez que o texto é alterado, é o onChangeTextprop;
5. A função a ser chamada quando o texto é enviado, é onSubmitEditingprop.

maneira: 🗨️. Portanto, a frase "Olá, Bob" seria traduzida como "🗨️ 🗨️ 🗨️". O código a seguir realiza essa ação.

```
import { useState } from 'react';
import { TextInput, Text, View } from 'react-native';

export default function App() {

  const [text, setText] = useState('');

  return (

    <View style={{flex: 1,
                  backgroundColor: '#fff',
                  alignItems: 'center',
                  justifyContent: 'center',padding: 10}}>

      <TextInput

        style={{height: 40}}

        placeholder="Type here to translate!"

        onChangeText={newText => setText(newText)}

        defaultValue={text} />

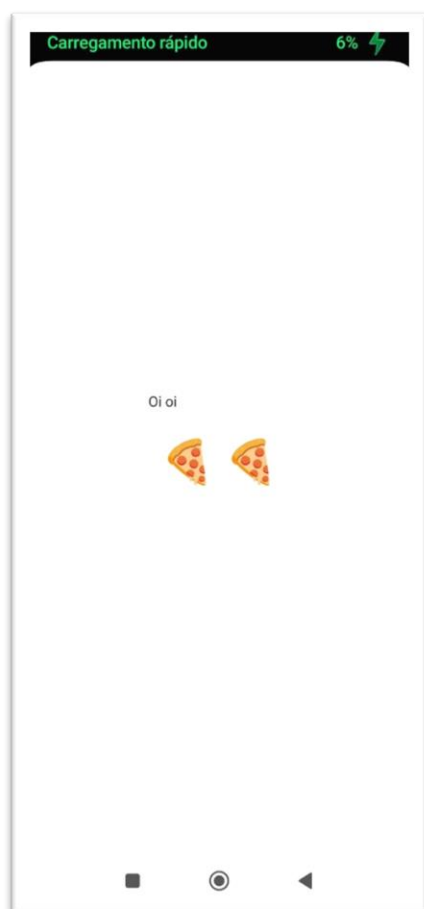
      <Text style={{padding: 10, fontSize: 42}}>

        {text
          .split(' ')

          .map(word => word && '🗨️')}

        .join(' ')}
      </Text>
    </View>
  );
};
```

- Resultados esperados



Microatividade 3: Implementar um Componente de Lista Dinâmica (ScrollView)

1. Criando um Componente de Lista dinâmica:

- Abra o seu editor de código (por exemplo, VS Code).
- Crie um novo arquivo JavaScript (ou JSX) para o seu componente.
- ScrollView para rolagem vertical:

•

```
import React from 'react';

import {Image, ScrollView, Text} from 'react-native';

export default function App() {

  const logo = {

    uri: 'https://reactnative.dev/img/tiny_logo.png',

    width: 64,

    height: 64,

  };

  return (

    <ScrollView>

      <Text style={{fontSize: 96}}>Scroll me plz</Text>

      <Image source={logo} />

      <Image source={logo} />

      <Image source={logo} />

      <Image source={logo} />

      <Image source={logo} />

      <Text style={{fontSize: 96}}>If you like</Text>

      <Image source={logo} />

      <Image source={logo} />

      <Image source={logo} />

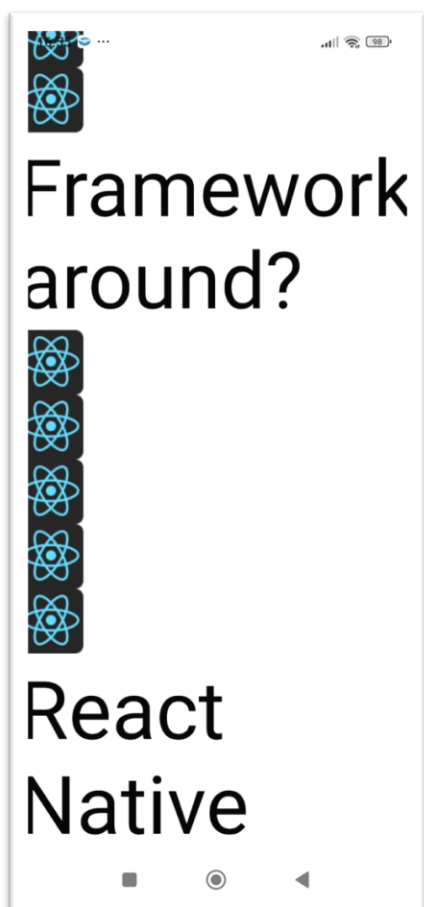
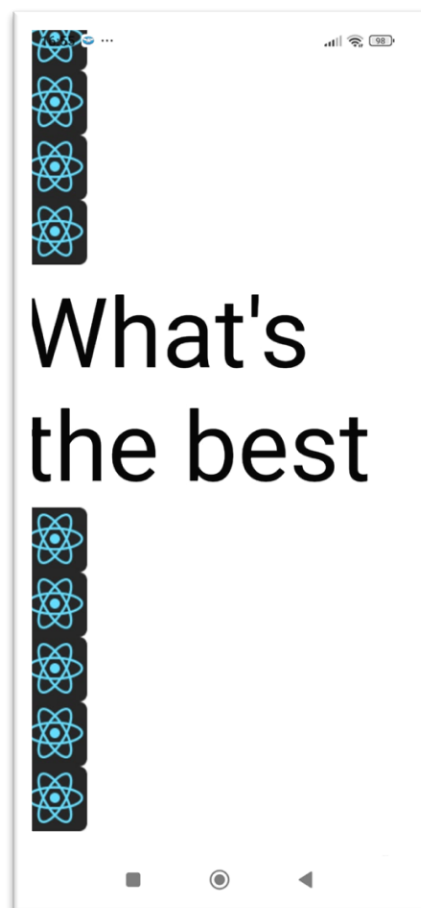
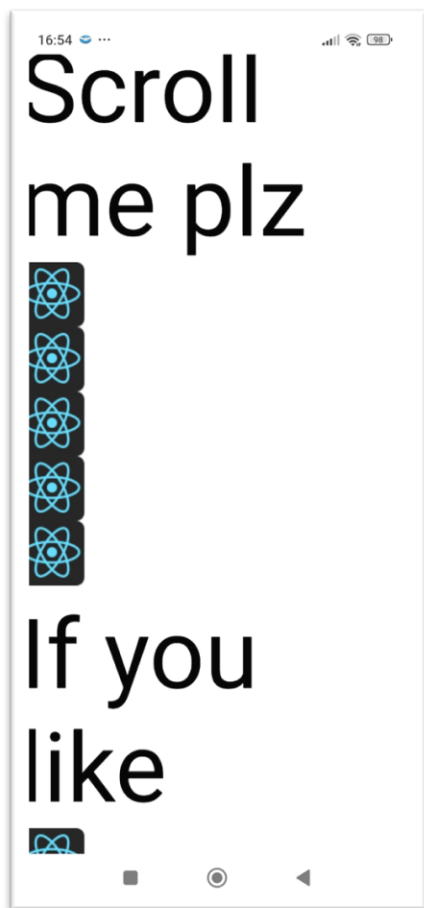
      <Image source={logo} />

      <Image source={logo} />

      <Text style={{fontSize: 96}}>Scrolling down</Text>
```

```
•
•   <Image source={logo} />
•
•   <Image source={logo} />
•
•   <Image source={logo} />
•
•   <Image source={logo} />
•
•   <Image source={logo} />
•
•   <Text style={{fontSize: 96}}>What's the best</Text>
•
•   <Image source={logo} />
•
•   <Image source={logo} />
•
•   <Image source={logo} />
•
•   <Image source={logo} />
•
•   <Image source={logo} />
•
•   <Text style={{fontSize: 78}}>Framework around?</Text>
•
•   <Image source={logo} />
•
•   <Image source={logo} />
•
•   <Image source={logo} />
•
•   <Image source={logo} />
•
•   <Image source={logo} />
•
•   <Text style={{fontSize: 80}}>React Native</Text>
•
• </ScrollView>
•
• );
•
• };
```

- Resultados esperados



Microatividade 4: Criando o visualizador de listas

1. Criando uma Lista Simples:

Abra o seu editor de código (por exemplo, VS Code).

Crie um novo arquivo JavaScript (ou JSX) para o seu componente.

substituir os nomes da lista de acordo com sua necessidade):

```
import React from 'react';

import {FlatList, StyleSheet, Text, View} from 'react-native';

export default function App() {

  const styles = StyleSheet.create({

    container: {

      flex: 1,

      paddingTop: 22,

    },

    item: {

      padding: 10,

      fontSize: 18,

      height: 44,

    },

  });

  return (

    <View style={styles.container}>

      <FlatList

        data={[

          {key: 'Devin'},

          {key: 'Dan'},

          {key: 'Dominic'},
```



```

        {key: 'Jackson'},

        {key: 'James'},

        {key: 'Joel'},

        {key: 'John'},

        {key: 'Jillian'},

        {key: 'Jimmy'},

        {key: 'Julie'},

    ]}

    renderItem={({item}) => <Text style={styles.item}>{item.key}</Text>{

    />

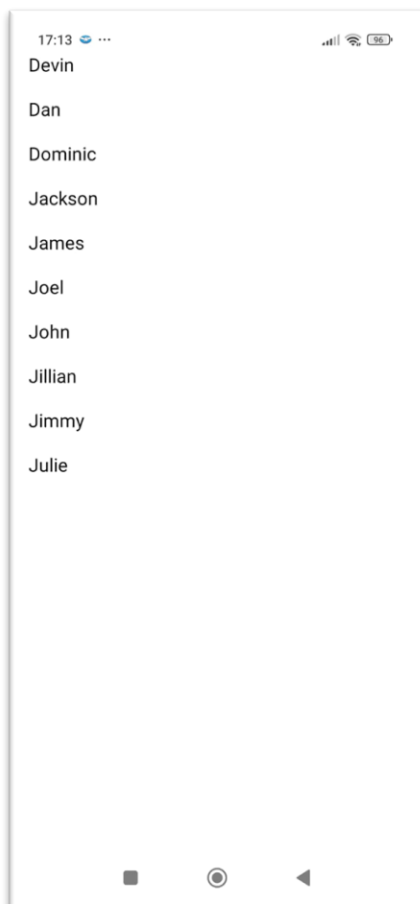
</View>

);

};

```

- Resultados esperados



2. Criando uma Lista com Seções:

Se você quiser renderizar uma lista com seções lógicas, talvez com cabeçalhos de seção, então um `SectionList` é o caminho a percorrer. Adicione um código semelhante ao exemplo a seguir para usar `SectionList`:

```
import React from 'react';

import {SectionList, StyleSheet, Text, View} from 'react-native';

export default function App() {

  const styles = StyleSheet.create({

    container: {

      flex: 1,

      paddingTop: 22,

    },

    sectionHeader: {

      paddingTop: 2,

      paddingLeft: 10,

      paddingRight: 10,

      paddingBottom: 2,

      fontSize: 14,

      fontWeight: 'bold',

      backgroundColor: 'rgba(247,247,247,1.0)',

    },

    item: {

      padding: 10,

      fontSize: 18,

      height: 44,

    },

  });

  return (
```

```
<View style={styles.container}>

  <SectionList

    sections={[

      {title: 'D', data: ['Devin', 'Dan', 'Dominic']},

      {

        title: 'J',

        data: [

          'Jackson',

          'James',

          'Jillian',

          'Jimmy',

          'Joel',

          'John',

          'Julie',

        ],

      },

    ]}

    renderItem={({item}) => <Text style={styles.item}>{item}</Text>}

    renderSectionHeader={({section}) => (

      <Text style={styles.sectionHeader}>{section.title}</Text>

    )}

    keyExtractor={item => `basicListEntry-${item}`}

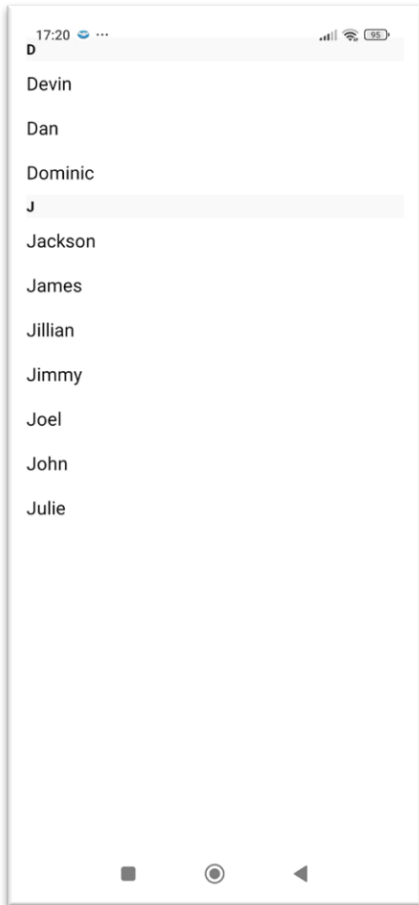
  />

</View>

);

};
```

- Resultados esperados



Microatividade 5: Empregar imagens, seja para exibir gráficos, ícones, fotos ou outros elementos visuais em um aplicativo React Native

1. Exibindo Imagens de Diferentes Fontes:

- Abra o seu editor de código (por exemplo, VS Code).
- Crie um novo arquivo JavaScript (ou JSX) para o seu componente.
- armazenamento local, rede e um esquema 'data':
-

```
• import React from 'react';
•
• import {View, Image, StyleSheet} from 'react-native';
•
• export default function App() {
•
•   const styles = StyleSheet.create({
•
•     container: {
•
•       paddingTop: 50,
•
•     },
•
•     tinyLogo: {
•
•       width: 50,
•
•       height: 50,
•
•     },
•
•     logo: {
•
•       width: 66,
•
•       height: 58,
•
•     },
•
•   });
•
•   return (
•
•     <View style={styles.container}>
•
•       <Image
•
•         style={styles.tinyLogo}
•
•       />
•
•     </View>
•
•   );
• }
```

```
        source={require('./node_modules/react-native/Libraries/NewAppScreen/components/logo.png')}

      />

    <Image

      style={styles.tinyLogo}

      source={{

        uri: 'https://reactnative.dev/img/tiny_logo.png',

      }}

    />

    <Image

      style={styles.logo}

      source={{

        uri:

'data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAADMAAAAzCAYAAAA6oTAqAAAAEXRFWHRTb2Z0d
2FyZQBwbmdjcnVzaEB1SfMAAABQSURBVGje7dSxQCBACARB+2/ab8BEeQNhfI6WSYzYL YudDQYGBgYGBgYGB
gYGBgYGBgYGBgYGBgmhivGQYGBgYGBgYGBgYGBgYGBgYGBbmQw+P/eMrC5UTVAAAAABJRU5ErkJggg==',

      }}

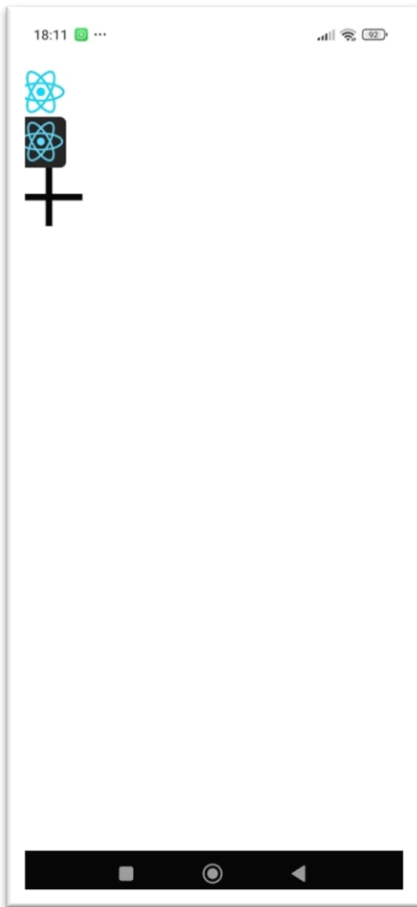
    />

  </View>

);

};
```

- Resultados esperados



2. Adicionando Estilos a uma Imagem:

- Copie e cole o código a seguir para adicionar estilos a uma imagem:

```
import React from 'react';

import {View, Image, StyleSheet} from 'react-native';

export default function App() {
  const styles = StyleSheet.create({

    container: {

      paddingTop: 50,

    },

    stretch: {

      width: 50,

      height: 200,

      resizeMode: 'stretch',

    },

  });

  return (

    <View style={styles.container}>

      <Image

        style={styles.stretch}

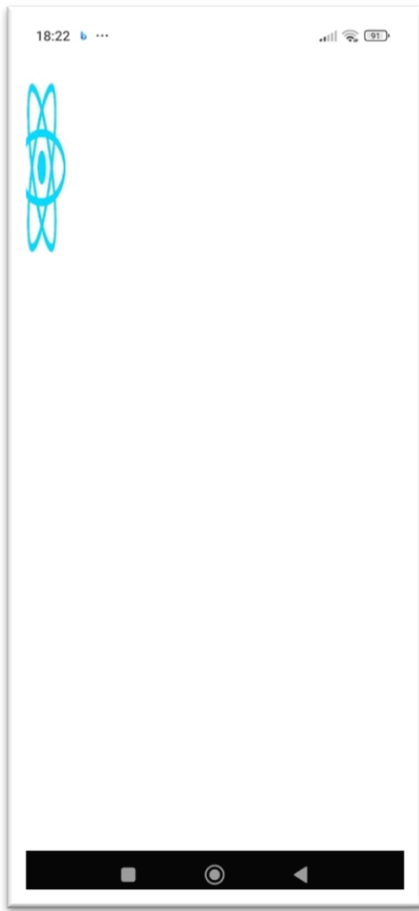
        source={require('./node_modules/react-native/Libraries/NewAppScreen/components/logo.png')}

      />

    </View>

  );
}
```


- Resultados esperados



Missão Prática | Vamos criar um App!

1. Criando um Componente Simples:

- Abra o seu editor de código.
- Crie um novo arquivo JavaScript (ou JSX) para o seu componente.
- dentro de um elemento **<Text>**:

```
import React from 'react';  
  
import { Text, View } from 'react-native';  
  
export default function App() {  
  const name = 'Maru';  
  return(  
    <View style={ {flex: 1,  
      backgroundColor: '#fff',  
      alignItems: 'center',  
      justifyContent: 'center',}}>  
      <Text>Hello, I am {name}!</Text></View>  
    );  
  };
```

- Resultados esperados



2. Adicionando Elementos ao Componente:

- com o **<Text>** dentro de um **<View>**:

```
import React from 'react';

import { Text, TextInput, View } from 'react-native';

export default function App() {

  return (

    <View style={ {flex: 1,
      backgroundColor: '#fff',
      alignItems: 'center',
      justifyContent: 'center',}}>

      <Text>Hello, I am...</Text>

      <TextInput

        style={{

          height: 40,

          borderColor: 'gray',

          borderWidth: 1,

        }}

        defaultValue="Name me!"

      />

    </View>

  );

};
```

Resultados esperados



3. Utilizando Props:

- (propriedades) e renderiza o nome de acordo com as props:

```
import React from 'react';

import { Text, View } from 'react-native';

export default function App() {

const Cat = (props) => {

  return (

    <View>

      <Text>Hello, I am {props.name}!</Text>

    </View>

  );

};

return (

  <View style={ {flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',}}>

    <Cat name="Maru" />

    <Cat name="Jellylorum" />

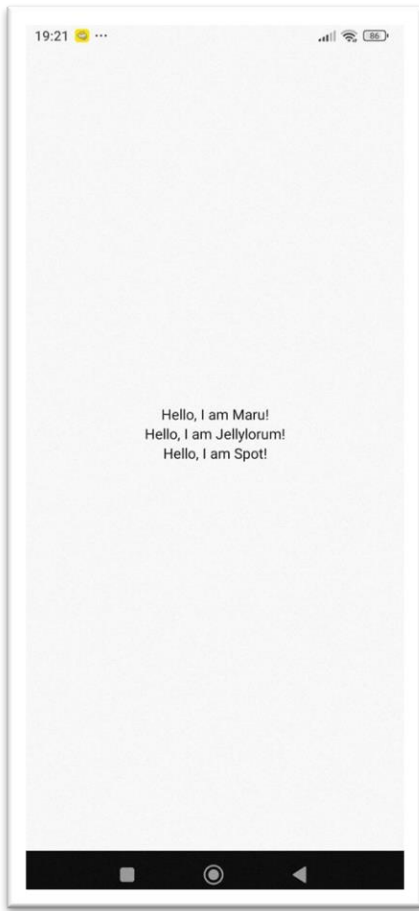
    <Cat name="Spot" />

  </View>

);

};
```

- Resultados esperados



4. Utilizando Imagens:

- Copie e cole o código a seguir para criar um componente que utiliza a tag `<Image>`
- para exibir uma imagem:

```
import React from 'react';

import { Text, View, Image } from 'react-native';

export default function App() {

  return (

    <View style={{flex: 1, alignItems: 'center', justifyContent: 'center'}}>

      <Image

        source={{

          uri: 'https://reactnative.dev/docs/assets/p_cat1.png',

        }}

        style={{ width: 200, height: 200 }}

      />

      <Text>Hello, I am your cat!</Text>

    </View>

  );

};
```

- Resultados esperados

