



Estácio

Programação Móvel e Android

ARA0089

Aula 02

PROFESSOR: EDIBERTO MARIANO

programacaoedi@gmail.com

Tema

1. SINTAXE E COMPONENTES DO REACTIVE NATIVE

Objetivos

Analisar a sintaxe JSX e os componentes básicos do React Native, baseando-se no ambiente do **Visual Studio Code**, em conjunto com o **Nodes.js**, para a construção de aplicativos móveis

Tópicos

1.3 COMPONENTES NATIVOS DO REACT NATIVE

1.4 DEPURAÇÃO DE APLICATIVOS

Componentes Básicos

A maioria dos aplicativos acabará usando um desses components básicos

Visualizar

O componente mais fundamental para a construção de uma interface do usuário.

Texto

Um componente para exibir texto.

Imagem

Um componente para exibir imagens.

TextInput

Um componente para inserir texto no aplicativo por meio de um teclado.

ScrollView

Fornece um contêiner de rolagem que pode hospedar vários componentes e exibições.

Folha de estilo

Fornece uma camada de abstração semelhante às folhas de estilo CSS.

React Native

Principais Componentes Nativos

COMPONENTE DE IU NATIVA DO REACT	VISUALIZAÇÃO DO ANDROID	VISUALIZAÇÃO DO IOS	ANALÓGICO DA WEB	DESCRIÇÃO
<View>	<ViewGroup>	<UIView>	Um sem rolagem<div>	Um contêiner que oferece suporte a layout com flexbox, estilo, algum manuseio de toque e controles de acessibilidade
<Text>	<TextView>	<UITextView>	<p>	Exibe, estiliza e aninha strings de texto e até manipula eventos de toque
<Image>	<ImageView>	<UIImageView>		Exibe diferentes tipos de imagens
<ScrollView>	<ScrollView>	<UIScrollView>	<div>	Um contêiner de rolagem genérico que pode conter vários componentes e visualizações
<TextInput>	<EditText>	<UITextField>	<input type="text">	Permite que o usuário insira texto

Programação Móvel e Android

React Native



Estácio

Componente Básico - View

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

const ViewBoxesWithColorAndText = () => {
  return (
    <View
      style={{
        flexDirection: 'row',
        height: 100,
        padding: 20,
      }}>
      <View style={{backgroundColor: 'blue', flex: 0.3}} />
      <View style={{backgroundColor: 'red', flex: 0.5}} />
      <Text>Hello World!</Text>
    </View>
  );
};

export default ViewBoxesWithColorAndText;
```





Componente Básico - **Text**

Texto

Um componente React para exibir texto.

Text suporta aninhamento, estilo e manipulação de toque.

No exemplo a seguir, o título aninhado e o corpo do texto herdarão de **fontFamily**, **styles.baseText** mas o título fornece seus próprios estilos adicionais. O título e o corpo serão empilhados um sobre o outro devido às novas linhas literais:

```
<Text>Meu segundo App em Rect-Native!</Text>
```




```
import React, {useState} from 'react';
import {Text, StyleSheet} from 'react-native';
const TextInANest = () => {
  const [titleText, setTitleText] = useState("Bird's Nest");
  const bodyText = 'This is not really a bird nest.';

  const onPressTitle = () => {
    setTitleText("Bird's Nest [pressed]");
  };
  return (
    <Text style={styles.baseText}>
      <Text style={styles.titleText} onPress={onPressTitle}>
        {titleText}
        {\n}
        {\n}
      </Text>
      <Text numberOfLines={5}>{bodyText}</Text>
    </Text>
  );
};
const styles = StyleSheet.create({
  baseText: {
    fontFamily: 'Cochin',
  },
  titleText: {
    fontSize: 20,
    fontWeight: 'bold',
  },
});
export default TextInANest;
```

React Native

Componentes Básicos - **Imagem**

```
<View style={styles.container}>  
  <Image  
    source={require('./assets/sample_image.jpg')}  
    style={styles.image}  
  />  
</View>
```

React Native

Componentes Básicos - **TextInput**

```
<View style={styles.container}>
  <TextInput
    style={styles.input}
    placeholder="Digite algo..."
    value={inputText}
    onChangeText={setInputText}
  />
  <Text>Você digitou: {inputText}</Text>
</View>
```

React Native

Componentes Básicos - **ScrollView**

```
const App = () => {  
  return (  
    <View style={styles.container}>  
      <ScrollView contentContainerStyle={styles.scrollContent}>  
        <Text style={styles.title}>Bem-vindo ao App ScrollView!</Text>  
        <Text style={styles.text}>  
          TEXTO LONGO.  
        </Text>  
      </ScrollView>  
    </View>  
  );  
}
```

React Native

Componentes Básicos – Folha de estilo - **styles**

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#f5f5f5',
  },
  title: {
    fontSize: 24,
    fontWeight: 'bold',
    marginBottom: 10,
  },
  subtitle: {
    fontSize: 18,
    color: '#666',
  },
});
```

React Native

Componentes do Usuário – Botão

```
<View style={styles.container}>  
  <Button  
    title="Pressione-me"  
    onPress={handlePress}  
    color="#3498db"  
  />  
</View>
```

React Native

Componentes do Usuário – Trocar

`const booleanValue = true; // Substitua pelo valor booleano que deseja renderizar`

```
return (  
  <View style={styles.container}>  
    <Text style={styles.text}>  
      O valor booleano é: {booleanValue ? 'Verdadeiro' : 'Falso'}  
    </Text>  
  </View>  
)
```

Programação Móvel e Android

React Native

Componentes de Lista - **ScrollView**

```
<ScrollView style = {styles.ScrollView} >  
  <Text style = {styles.text}>  
    escreve o texto aqui.....  
    escreve o texto aqui.....  
    ...  
    ...  
    escreve o texto aqui.....  
  </Text>  
</ScrollView>
```




Componentes de Lista - FlatList

```
const data = [  
  { key: 'item1', text: 'Item 1' },  
  { key: 'item2', text: 'Item 2' },  
  { key: 'item3', text: 'Item 3' },  
  // ... more data items  
];  
const renderItem = ({ item }) => (  
  <View style={{ padding: 10 }}>  
    <Text>{item.text}</Text>  
  </View>  
);  
const App = () => {  
  return (  
    <FlatList  
      data={data}  
      renderItem={renderItem}  
    />  
  );  
};
```



Componentes de Lista - **SectionList**

```
const data = [
  {
    title: 'Seção 1',
    data: [
      { key: 'item1', text: 'Item 1' },
      { key: 'item2', text: 'Item 2' },
    ],
  },
  {
    title: 'Seção 2',
    data: [
      { key: 'item3', text: 'Item 3' },
      { key: 'item4', text: 'Item 4' },
    ],
  },
  // ... more sections
];
```

```
const renderItem = ({ item }) => (
  <View style={{ padding: 10 }}>
    <Text>{item.text}</Text>
  </View>
);

const renderSectionHeader = ({ section }) => (
  <View style={{ backgroundColor: 'lightgray', padding: 10 }}>
    <Text>{section.title}</Text>
  </View>
);

const App = () => {
  return (
    <SectionList
      sections={data}
      renderItem={renderItem}
      renderSectionHeader={renderSectionHeader}
    />
  );
};
```