

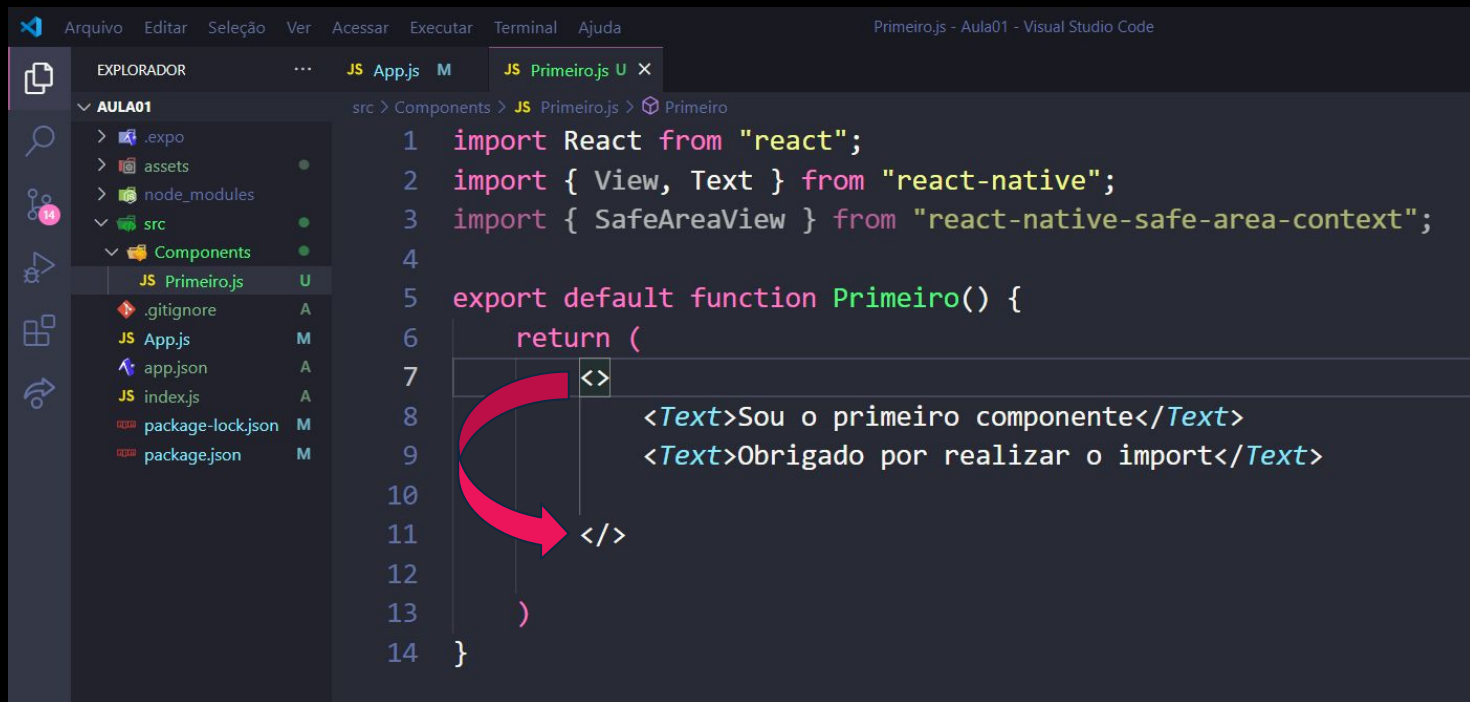
Mobile Application Development

Prof. Fernando Pinéo

Fragments, Botão, useState

Fragments

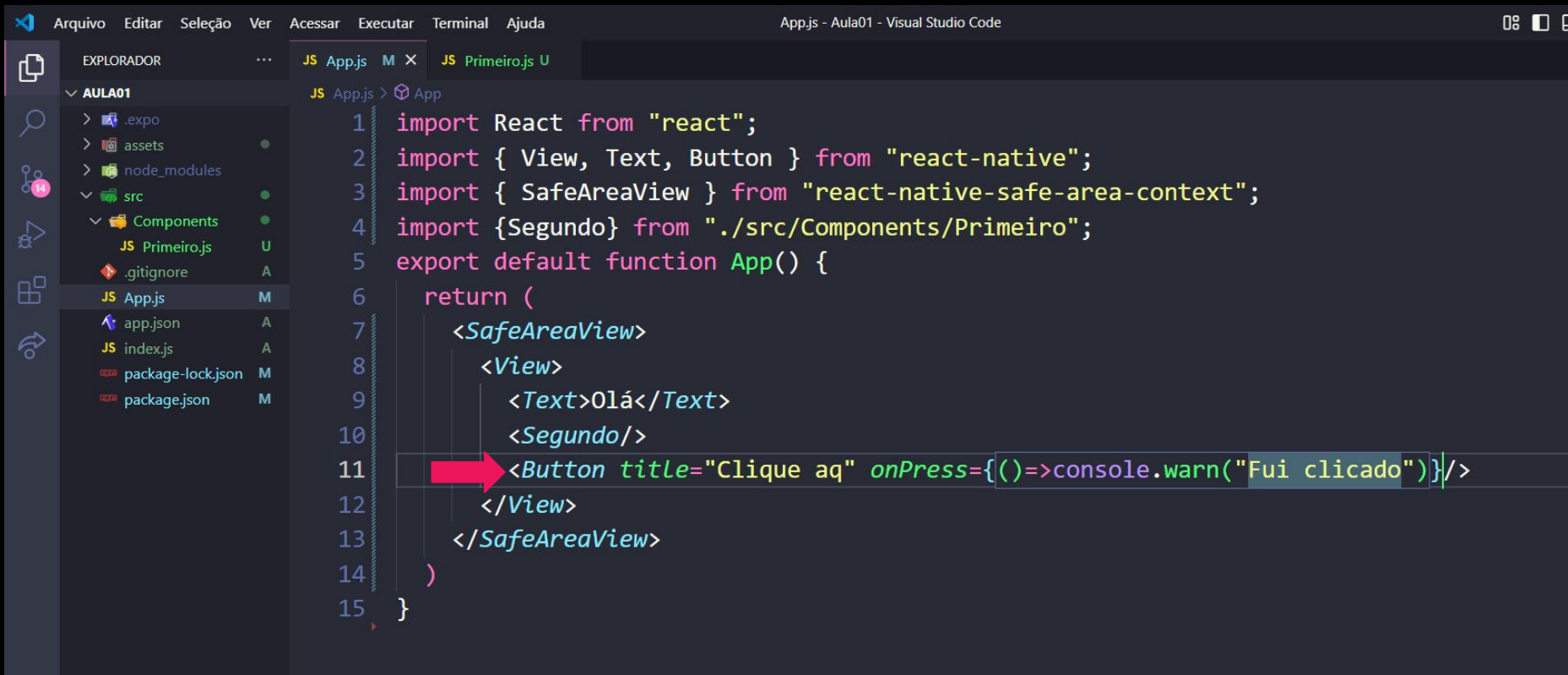
No React Native, os Fragments servem para agrupar múltiplos componentes sem adicionar um nó extra no DOM ou na árvore de visualização.



```
1 import React from "react";
2 import { View, Text } from "react-native";
3 import { SafeAreaView } from "react-native-safe-area-context";
4
5 export default function Primeiro() {
6   return (
7     <>
8       <Text>Sou o primeiro componente</Text>
9       <Text>Obrigado por realizar o import</Text>
10
11     </>
12
13   )
14 }
```

Elemento Botão

FIAP



The image shows a Visual Studio Code editor window with a dark theme. The top bar displays the menu (Arquivo, Editar, Seleção, Ver, Acessar, Executar, Terminal, Ajuda) and the title 'App.js - Aula01 - Visual Studio Code'. The Explorer sidebar on the left shows a project named 'AULA01' with a file tree including '.expo', 'assets', 'node_modules', 'src', 'Components', and 'App.js'. The main editor area shows the content of 'App.js', which imports React, react-native, and react-native-safe-area-context, and defines an App component. A red arrow points to the Button component in the JSX, which has the text 'Clique aq' and an onPress function that logs a warning to the console.

```
1 import React from "react";
2 import { View, Text, Button } from "react-native";
3 import { SafeAreaView } from "react-native-safe-area-context";
4 import {Segundo} from "../src/Components/Primeiro";
5 export default function App() {
6   return (
7     <SafeAreaView>
8       <View>
9         <Text>Olá</Text>
10        <Segundo/>
11        <Button title="Clique aq" onPress={()=>console.warn("Fui clicado")}/>
12      </View>
13    </SafeAreaView>
14  )
15 }
```

Elemento Botão

```
JS App.js M X
JS App.js > styles > container

1 import { StatusBar } from 'expo-status-bar';
2 import { StyleSheet, Text, View, Button } from 'react-native';
3
4 export default function App() {
5   const teste = ()=>{
6     console.log("teste")
7   }
8   return (
9     <View style={styles.container}>
10       <Text>Open up App.js to start working on your app!</Text>
11       <Button title='Clique aqui' onPress={teste} />
12       <Button title='Clique aqui2' onPress={function(){console.log("teste2")}} />
13       <Button title='Clique aqui3' onPress={()=>console.log("teste3")} />
14       <StatusBar style="auto" />
15     </View>
16   );
17 }
```

Hook useState

Utilizado para armazenar e gerenciar valores que podem mudar ao longo do tempo, como entradas de usuários, contadores, ou o estado de um componente, e atualiza a interface de acordo com essas mudanças.

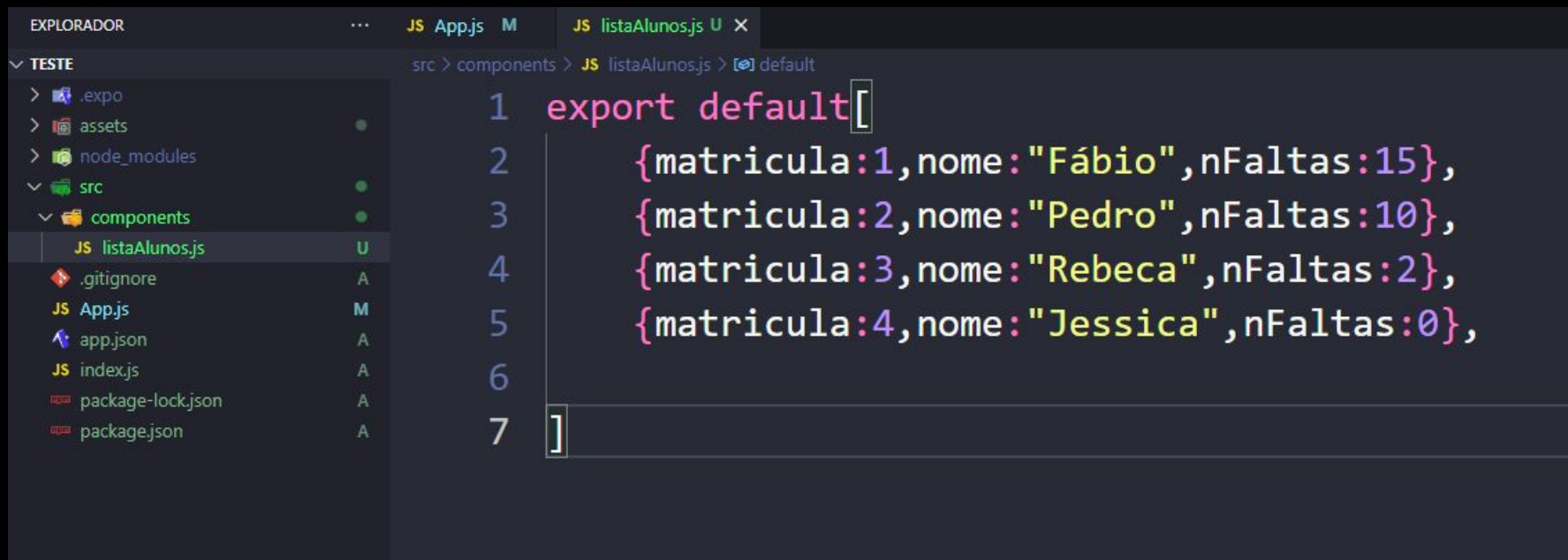
```
JS App.js M X
JS App.js > App > gap
1 import { StatusBar } from 'expo-status-bar';
2 import { useState } from 'react';
3 import { StyleSheet, Text, View, Button } from 'react-native';
4
5 export default function App() {
6   const [numero, setNumero] = useState(1)
7   return (
8     <View style={styles.container}>
9       <Text>numero {numero}</Text>
10      <View style={{flexDirection:"row",gap:"5"}}>
11        <Button title='Incrementar' onPress={() => setNumero(numero + 1)} />
12        <Button title='Decrementar' onPress={() => setNumero(numero - 1)} />
13      </View>
14      <StatusBar style="auto" />
15    </View>
16  );
17 }
```



The screenshot shows an Android emulator window titled 'Android Emulator - Medium_Phone_API_35:5554'. The status bar at the top shows the time 11:29 and LTE signal. The app interface consists of a white background with the text 'numero 1' in the center. Below the text are two blue buttons with white text: 'INCREMENTAR' on the left and 'DECREMENTAR' on the right.

Trabalhando com Lista - Comp FlatList FIA/P

Criando com um array de objetos..

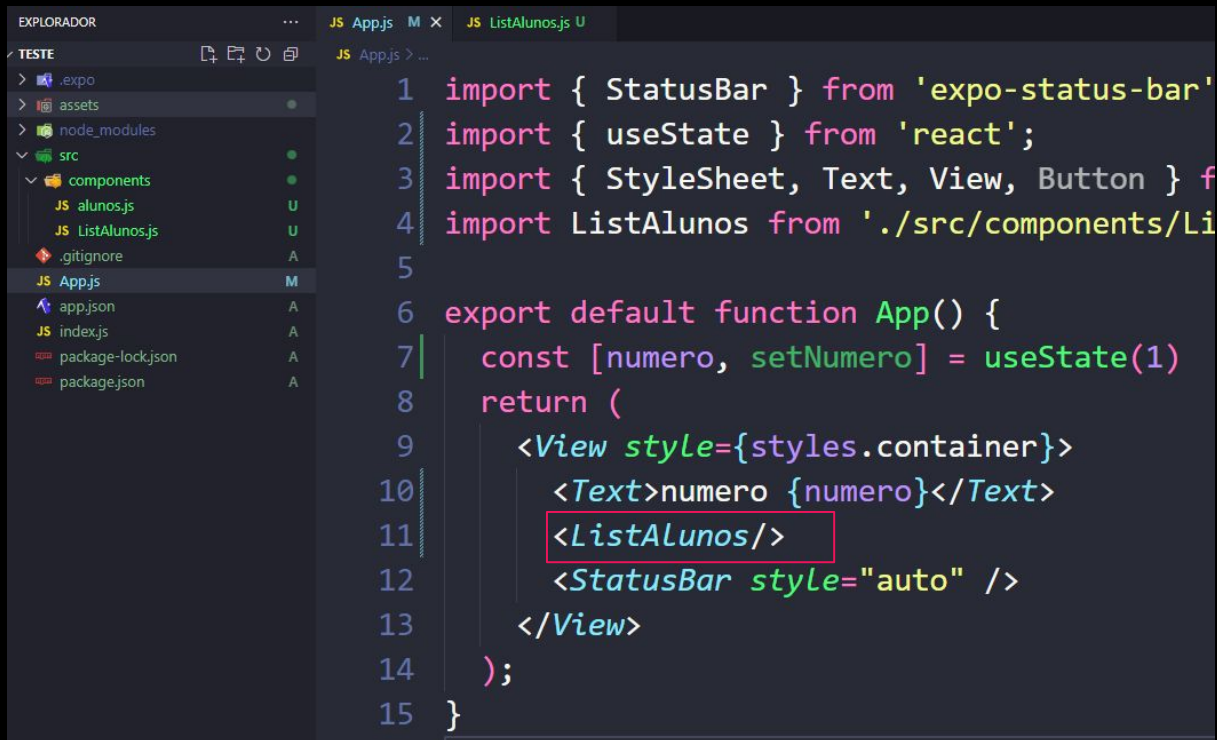


The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer, titled 'EXPLORADOR', shows a project structure with a 'components' folder containing 'listaAlunos.js'. The code editor, titled 'JS listaAlunos.js', shows the following code:

```
1 export default [  
2   {matricula:1,nome:"Fábio",nFaltas:15},  
3   {matricula:2,nome:"Pedro",nFaltas:10},  
4   {matricula:3,nome:"Rebeca",nFaltas:2},  
5   {matricula:4,nome:"Jessica",nFaltas:0},  
6 ]
```

Trabalhando com Lista - Comp FlatList FIA/P

Importando em App.js

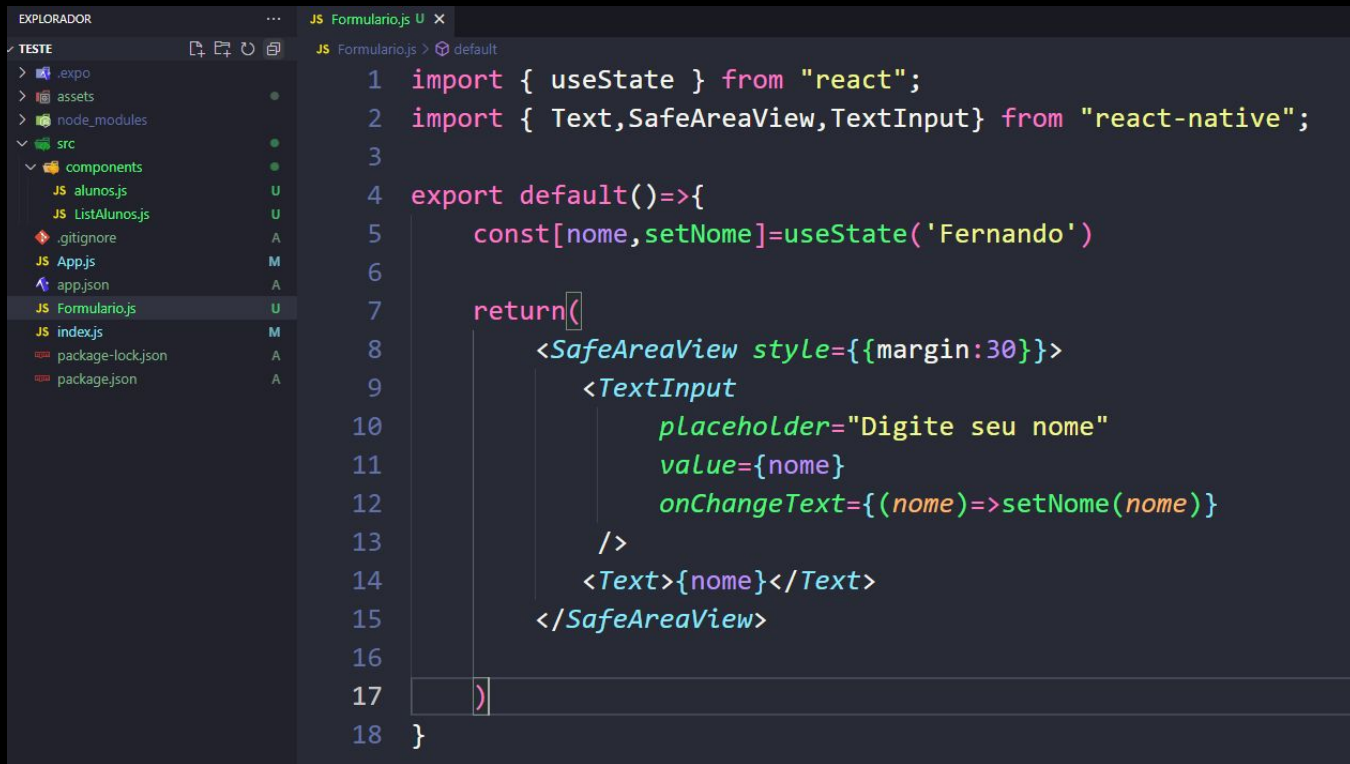


The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a 'src' directory containing 'components' and 'ListAlunos.js'. The code editor shows the 'App.js' file with the following code:

```
1 import { StatusBar } from 'expo-status-bar'
2 import { useState } from 'react';
3 import { StyleSheet, Text, View, Button } from 'react-native'
4 import ListAlunos from './src/components/ListAlunos'
5
6 export default function App() {
7   const [numero, setNumero] = useState(1)
8   return (
9     <View style={styles.container}>
10       <Text>numero {numero}</Text>
11       <ListAlunos/>
12       <StatusBar style="auto" />
13     </View>
14   );
15 }
```


TextInput

Utilizado para entrar com alguma informação no app, exemplo: Formulário



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer, titled 'EXPLORADOR', shows a project structure with folders like 'assets', 'node_modules', and 'src'. Under 'src', there's a 'components' folder containing 'alunos.js' and 'ListAlunos.js'. The 'Formulario.js' file is selected. The code editor, titled 'JS Formulario.js', shows the following code:

```
1 import { useState } from "react";
2 import { Text, SafeAreaView, TextInput } from "react-native";
3
4 export default()=>{
5   const[nome,setNome]=useState('Fernando')
6
7   return(
8     <SafeAreaView style={{margin:30}}>
9       <TextInput
10         placeholder="Digite seu nome"
11         value={nome}
12         onChangeText={(nome)=>setNome(nome)}
13       />
14       <Text>{nome}</Text>
15     </SafeAreaView>
16   )
17 }
18 }
```

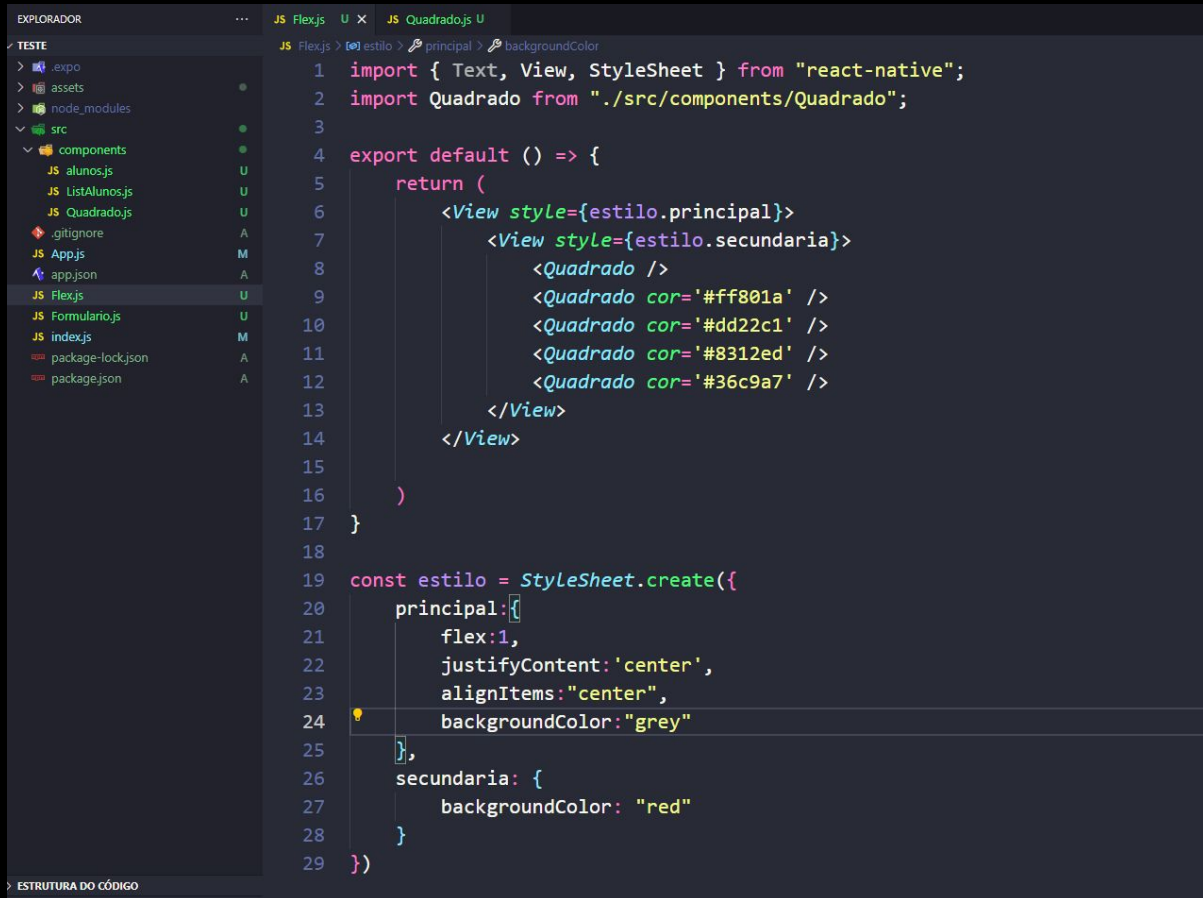
Flex-Box

FIAP

```
1  import { Text, View } from "react-native";
2
3  export default props =>{
4    return(
5      <View style={{
6        height:50,
7        width:50,
8        backgroundColor:props.cor || '#000'
9      }}
10     />
11   )
12 }
```

Flex-Box

FIAP

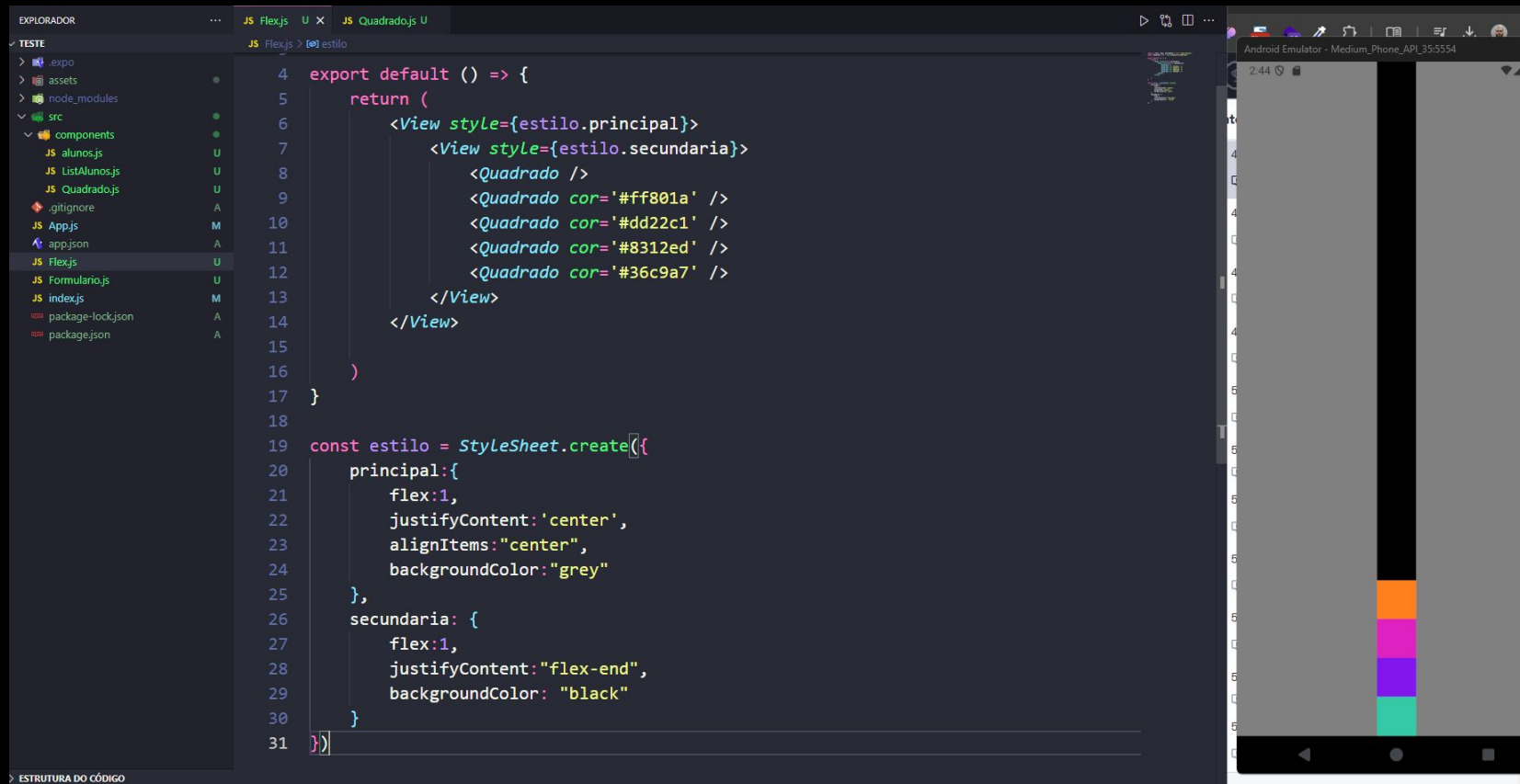


The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a 'src' directory containing 'components' and 'alunos.js'. The code editor shows the 'estilo.js' file with the following code:

```
1 import { Text, View, StyleSheet } from "react-native";
2 import Quadrado from "../src/components/Quadrado";
3
4 export default () => {
5   return (
6     <View style={estilo.principal}>
7       <View style={estilo.secundaria}>
8         <Quadrado />
9         <Quadrado cor='#ff801a' />
10        <Quadrado cor='#dd22c1' />
11        <Quadrado cor='#8312ed' />
12        <Quadrado cor='#36c9a7' />
13      </View>
14    </View>
15  )
16 }
17
18
19 const estilo = StyleSheet.create({
20   principal: {
21     flex: 1,
22     justifyContent: 'center',
23     alignItems: "center",
24     backgroundColor: "grey"
25   },
26   secundaria: {
27     backgroundColor: "red"
28   }
29 })
```

Flex-Box

FIAP



The image shows a code editor with a file explorer on the left and an Android emulator on the right. The code editor displays a JavaScript file named `estilo.js` with the following content:

```
4 export default () => {
5   return (
6     <View style={estilo.principal}>
7       <View style={estilo.secundaria}>
8         <Quadrado />
9         <Quadrado cor='#ff801a' />
10        <Quadrado cor='#dd22c1' />
11        <Quadrado cor='#8312ed' />
12        <Quadrado cor='#36c9a7' />
13      </View>
14    </View>
15  )
16 }
17
18
19 const estilo = StyleSheet.create({
20   principal:{
21     flex:1,
22     justifyContent:'center',
23     alignItems:"center",
24     backgroundColor:"grey"
25   },
26   secundaria: {
27     flex:1,
28     justifyContent:"flex-end",
29     backgroundColor: "black"
30   }
31 })
```

The Android emulator on the right shows a mobile screen with a grey background. In the center, there is a vertical stack of five colored squares: a small black square at the top, followed by orange, magenta, purple, and cyan squares at the bottom. This visualizes the layout defined in the code, where the main container has a grey background and the inner container (which is at the flex-end) has a black background and contains the colored squares.

Flex-Box

FLANP

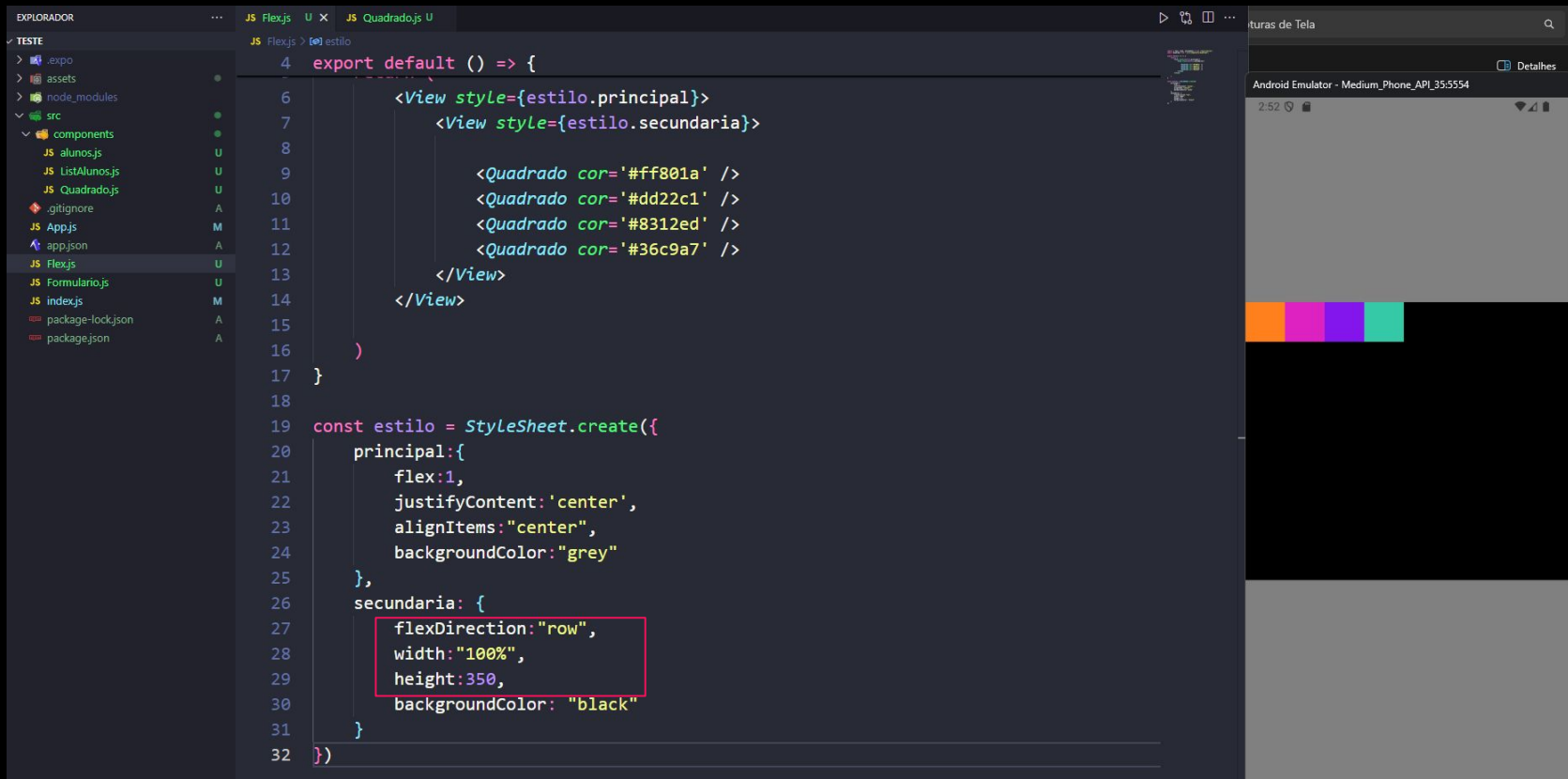
The image shows a code editor with a file explorer on the left and an Android emulator on the right. The code is in a file named `JS Flex.js` and defines a component that uses Flexbox to layout four colored squares.

```
4 export default () => {
5   return (
6     <View style={estilo.principal}>
7       <View style={estilo.secundaria}>
8
9         <Quadrado cor='#ff801a' />
10        <Quadrado cor='#dd22c1' />
11        <Quadrado cor='#8312ed' />
12        <Quadrado cor='#36c9a7' />
13      </View>
14    </View>
15  )
16 }
17
18
19 const estilo = StyleSheet.create({
20   principal:{
21     flex:1,
22     justifyContent:'center',
23     alignItems:"center",
24     backgroundColor:"grey"
25   },
26   secundaria: {
27     width:"100%",
28     alignItems:"center",
29     justifyContent:"flex-end",
30     backgroundColor: "black"
31   }
32 })
```

The Android emulator on the right shows the rendered output: a grey container with four colored squares (orange, magenta, purple, and teal) arranged horizontally. The status bar at the top of the emulator shows the time as 2:48.

Flex-Box

FLANP

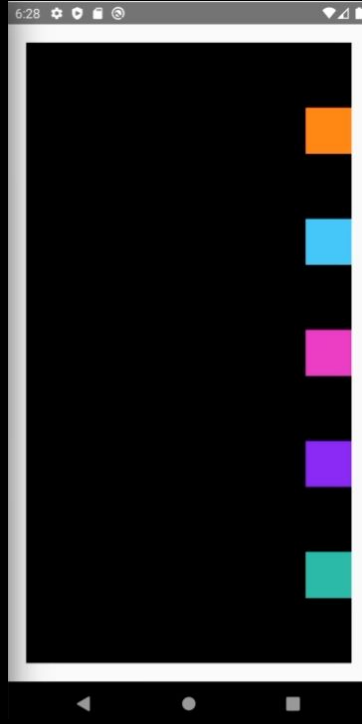


```
EXPLORADOR
... JS Flexjs U X JS Quadrado.js U
TESTE
> .expo
> assets
> node_modules
> src
  > components
    JS alunos.js U
    JS ListAlunos.js U
    JS Quadrado.js U
  .gitignore A
  JS App.js M
  app.json A
  JS Flexjs U
  JS Formulario.js U
  JS index.js M
  package-lock.json A
  package.json A

JS Flexjs > [estilo]
4 export default () => {
5
6   <View style={estilo.principal}>
7     <View style={estilo.secundaria}>
8
9       <Quadrado cor='#ff801a' />
10      <Quadrado cor='#dd22c1' />
11      <Quadrado cor='#8312ed' />
12      <Quadrado cor='#36c9a7' />
13    </View>
14  </View>
15
16  )
17 }
18
19 const estilo = StyleSheet.create({
20   principal:{
21     flex:1,
22     justifyContent:'center',
23     alignItems:"center",
24     backgroundColor:"grey"
25   },
26   secundaria: {
27     flexDirection:"row",
28     width:"100%",
29     height:350,
30     backgroundColor: "black"
31   }
32 })
```

Desafio - Flex

FIAP



Dúvidas?