

Mobile Application Development

Prof. Fernando Pinéo

FIREBASE(FIRESTORE)

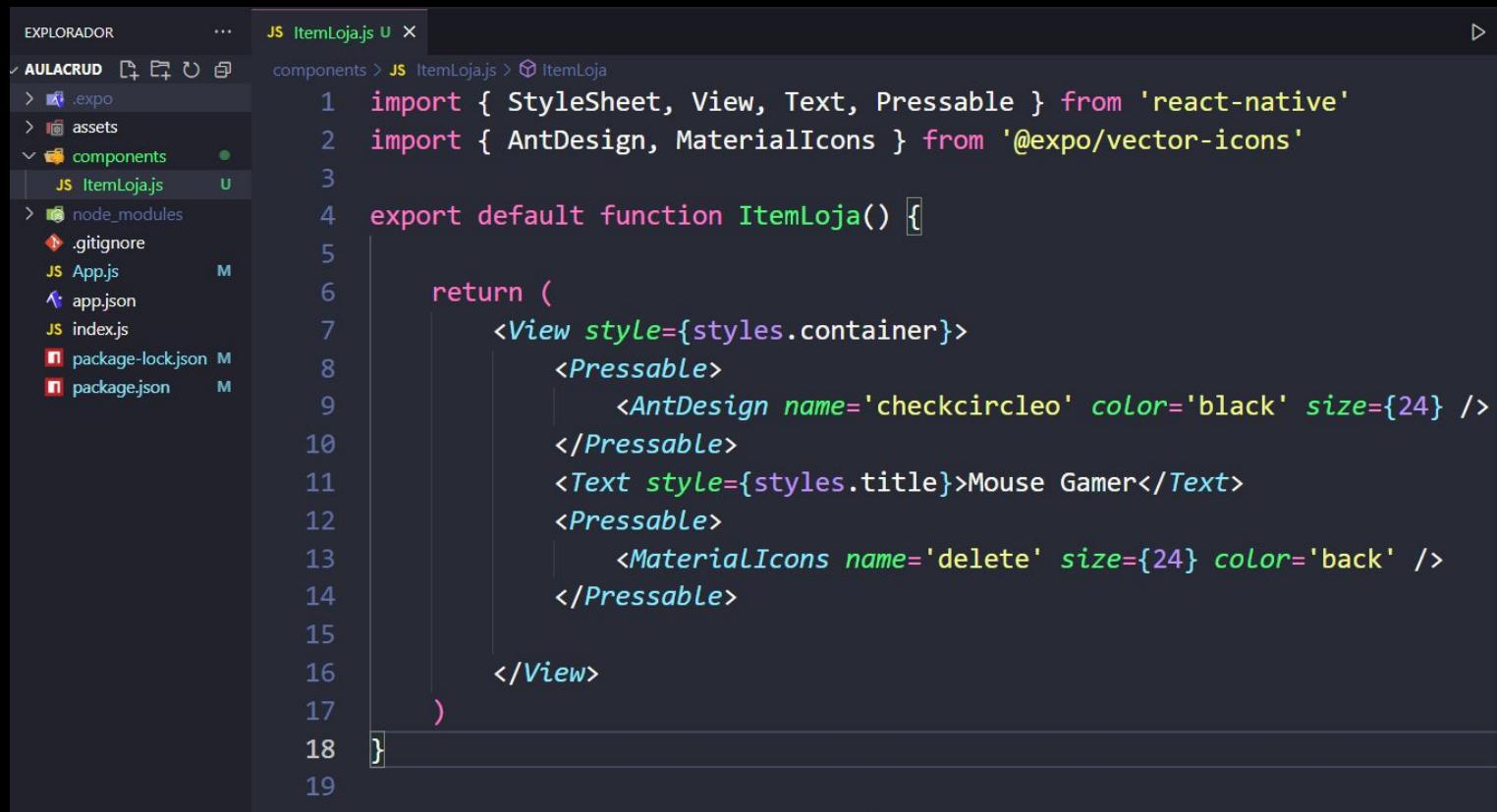
Criando projeto em branco

FIAP

```
npx create-expo-app AulaCrud --template blank
```

Criar o componente ItemLoja.js conforme imagem

FIA P



The screenshot shows a code editor interface with a dark theme. On the left is a sidebar labeled 'EXPLORADOR' containing project files: 'AULACRUD', '.expo', 'assets', 'components' (which contains 'ItemLoja.js'), 'node_modules', '.gitignore', 'App.js', 'app.json', 'index.js', 'package-lock.json', and 'package.json'. The main area is titled 'JS ItemLoja.js' and shows the following code:

```
1 import { StyleSheet, View, Text, Pressable } from 'react-native'
2 import { AntDesign, MaterialIcons } from '@expo/vector-icons'
3
4 export default function ItemLoja() {
5
6     return (
7         <View style={styles.container}>
8             <Pressable>
9                 <AntDesign name='checkcircleo' color='black' size={24} />
10            </Pressable>
11            <Text style={styles.title}>Mouse Gamer</Text>
12            <Pressable>
13                <MaterialIcons name='delete' size={24} color='black' />
14            </Pressable>
15
16        </View>
17    )
18 }
19
```

Estilos do ItemLoja.js

FIAP

The screenshot shows a code editor interface with a dark theme. On the left is a sidebar labeled 'EXPLORADOR' containing project files: .expo, assets, components (with ItemLoja.js selected), node_modules, .gitignore, App.js, app.json, index.js, package-lock.json, and package.json. The main area is titled 'JS ItemLoja.js' and displays the following code:

```
4  export default function ItemLoja() {
15    return (
16      <View style={styles.container}>
17        <Text style={styles.title}>
18      </View>
19    );
20  }
21  const styles = StyleSheet.create({
22    container: {
23      flexDirection:'row',
24      backgroundColor:'lightgray',
25      justifyContent:'space-between',
26      alignItems:'center',
27      padding:10,
28      width:'90%',
29      alignSelf:'center',
30      borderRadius:10
31    },
32    title:{ 
33      flex:1,
34      marginLeft:10,
35      fontSize:17,
36      fontWeight:500
37    }
38  })
```

Colocando um TextInput

FIAP

```
export default function App(){
  return(
    <View style={styles.container}>
      <ItemLoja/>
      <ItemLoja/>
      <ItemLoja/>
      <ItemLoja/>
      <TextInput placeholder='Digite o nome do produto' style={styles.input}/>
    </View>
  )
}
const styles = StyleSheet.create({
  container:{
    flex:1
  },
  input:{
    backgroundColor:'lightgrey',
    padding:10,
    fontSize:15,
    width:'90%',
    alignSelf:'center',
    borderRadius:10,
    marginTop:'auto'
  }
})
```

Resultado

FIAP

<input checked="" type="checkbox"/> Mouse gamer	

Digite o nome do produto

No Firebase, criar um projeto

FIAP

- × Criar um projeto

Vamos começar nomeando o
projeto [?]

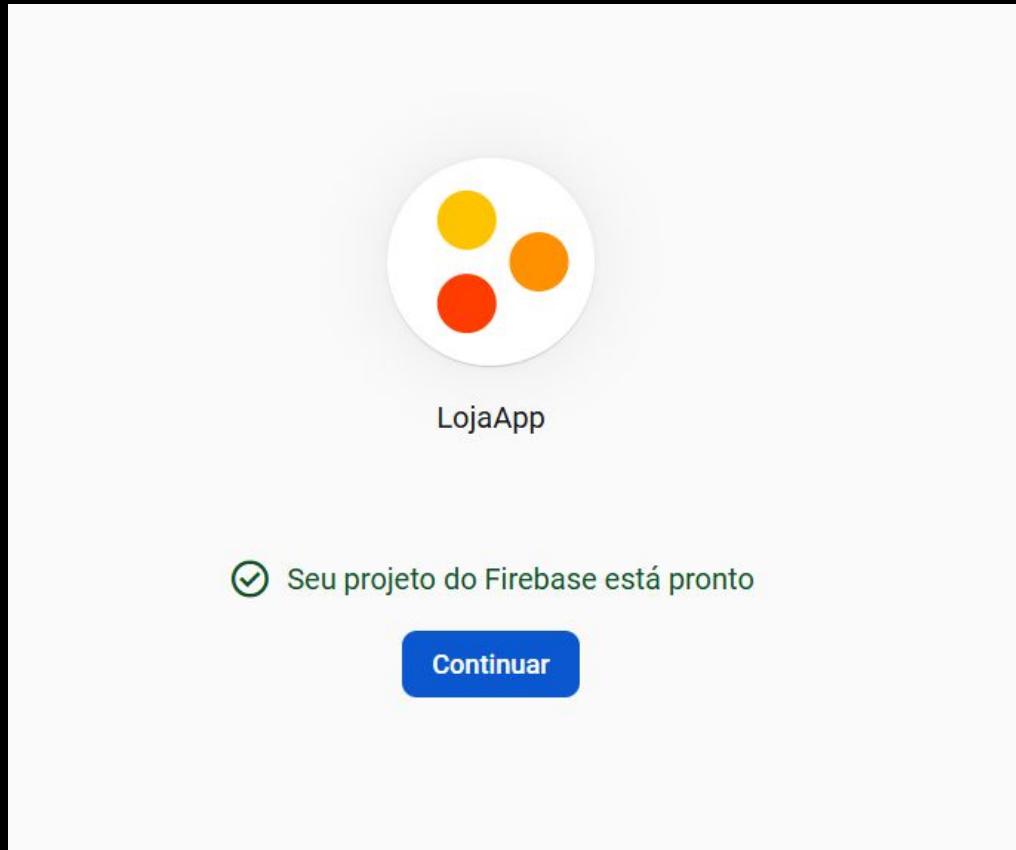
Nome do projeto

LojaApp

 lojaapp-cd25e

No Firebase, criar um projeto

FIAP



No Firebase, registrar app

FIAP

The screenshot shows the Firebase console interface. On the left, there's a sidebar with navigation links: 'Visão geral do projeto' (selected), 'Categorias dos produtos', 'Criação', 'Executar', 'Analytics', 'AI', 'Todos os produtos', and 'Ferramentas de desenvolvimento relacionadas' (with 'Firebase Studio' and 'Checks' listed). At the bottom of the sidebar are buttons for 'Spark' and 'Fazer upgrade'. The main area is titled 'LojaApp' and shows a dark-themed dashboard. It features a call-to-action button 'Começando agora? Conte ao Gemini sobre seu projeto' with a blue outline. Below it, the text 'Comece adicionando o Firebase ao seu aplicativo' is displayed, with a red arrow pointing to the 'iOS+' icon. There are also icons for TV, code (HTML/CSS), and other platforms like Android and Web. A 3D rendering of a person thinking is on the right. At the bottom, there's a link 'Adicione um app para começar'.

Instalar o firebase

FIAP

`npm install firebase`

No Firebase, registrar app copiar as configs e colocar no arq firebaseConfig

FIAP

The screenshot shows the Firebase console interface for a project named 'lojaapp-cd25e'. The top navigation bar includes links for 'Início | Microsoft...', 'Alunos - Home', and the current URL 'console.firebaseio.google.com/project/lojaapp-cd25e/overview?hl=pt-br'. A sidebar on the left lists 'Registrar app' (marked with a checkmark) and 'Adicionar o SDK do Firebase' (step 2). Below these, there are two radio button options: 'Usar o npm' (selected) and 'Usar a tag <script>'. A note below explains that if using npm and a bundler like webpack or Rollup, the command '\$ npm install firebase' should be run. A code editor window displays the generated JavaScript code for initializing the Firebase SDK:

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyAXBFYmFxp_QRnAAGS-PhfLYRroNNjkri4",
  authDomain: "lojaapp-cd25e.firebaseio.com",
  projectId: "lojaapp-cd25e",
  storageBucket: "lojaapp-cd25e.firebaseiostorage.app",
  messagingSenderId: "215240764016",
  appId: "1:215240764016:web:e755b9b012dd2618c0a2b5"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
```

At the bottom, a note states: 'Observação: essa opção usa o [SDK modular para JavaScript](#), que reduz o tamanho do SDK.'

fascinated green almond ⓘ
Log in to save your changes as you work

Open files Project assets components services firebaseConfig.js App.js package.json README.md

```
1 // Import the functions you need from the SDKs you need
2 import { initializeApp } from "firebase/app";
3 // TODO: Add SDKs for Firebase products that you want to use
4 // https://firebase.google.com/docs/web/setup#available-libraries
5
6 // Your web app's Firebase configuration
7 const firebaseConfig = {
8   apiKey: "AIzaSyAXBfYmFXp_QRnAA0S-PhfLYRroNNjkri4",
9   authDomain: "lojaapp-cd25e.firebaseio.com",
10  projectId: "lojaapp-cd25e",
11  storageBucket: "lojaapp-cd25e.firebaseiostorage.app",
12  messagingSenderId: "215240764016",
13  appId: "1:215240764016:web:e755b9b012dd2618c0a2b5"
14 };
15
16 // Initialize Firebase
17 const app = initializeApp(firebaseConfig);
```

Criando o serviço de database

FIAP

The screenshot shows the Firebase console interface. On the left, a sidebar lists various services under 'Criação': App Check, App Hosting, Authentication, Data Connect, Extensions, Firestore Database (which has a red arrow pointing to it), Functions, Hosting, Machine Learning, Realtime Database, and Storage. The main area displays the 'LojaApp' project details, including the 'Piano Spark' plan, a note about starting with Gemini 2.0, and a button to 'Faça o teste agora mesmo'. At the bottom, there's a banner encouraging app development acceleration.

Criando banco de dados

FIAP

The screenshot shows the Firebase Cloud Firestore dashboard for a project named "LojaApp". The left sidebar includes links for "Visão geral do projeto", "Atalhos do projeto", "Firestore Database" (which is selected and highlighted in blue), "Categorias dos produtos", "Criação", "Executar", "Analytics", "AI", and "Todos os produtos". The main content area features the title "Cloud Firestore" and the text "Atualizações em tempo real, consultas eficientes e escalonamento automático". It contains two buttons: "Criar banco de dados" (highlighted with a red arrow) and "Pedir ao Gemini". To the right is a stylized illustration of a stack of blue server racks with a magnifying glass examining them, set against a dark background with orange outlines.

Firebase

LojaApp ▾

Visão geral do projeto

Atalhos do projeto

Firestore Database

Categorias dos produtos

Criação

Executar

Analytics

AI

Todos os produtos

Ferramentas de desenvolvimento relacionadas

[Firebase Studio](#) ?

[Checks](#) ?

Cloud Firestore

Atualizações em tempo real, consultas eficientes e escalonamento automático

[Criar banco de dados](#)

[Pedir ao Gemini](#)

Saiba mais

Criando banco de dados

FIAP

The screenshot shows the Firebase console interface for creating a new database. On the left, there's a sidebar with project settings like 'Visão geral do projeto', 'Atalhos do projeto', and 'Firestore Database' (which is selected). The main area has a dark theme with a central modal window titled 'Criar banco de dados' (Create database). The modal is on step 1: 'Definir nome e local' (Define name and location). It contains a note about using Firestore with MongoDB and a link to learn more. Below that, there's a field for the database ID set to '(default)' and a dropdown for the location set to 'southamerica-east1 (São Paulo)'. A red arrow points to this location dropdown. At the bottom of the modal, there's a warning message: 'Não será possível alterar o local depois de terminar a configuração dele.' (It will not be possible to change the location after finishing the configuration) and a 'Saiba mais' link. The 'Cancelar' (Cancel) and 'Próxima' (Next) buttons are also visible. The overall theme is dark with blue and white UI elements.

Criando banco de dados

FIAP

The screenshot shows the Firebase console interface. On the left, there's a sidebar with navigation links like 'Visão geral do projeto', 'Atalhos do projeto', 'Firestore Database' (which is selected), 'Categorias dos produtos', 'Criação', 'Executar', 'Analytics', 'AI', and 'Todos os produtos'. Below these are sections for 'Ferramentas de desenvolvimento relacionadas', 'Firebase Studio', and 'Checks'. At the bottom, it says 'Spark Sem custos (US\$ 0/mês)' and 'Fazer upgrade NOVO'.

The main area shows a modal titled 'Criar banco de dados' (Create database). It has two tabs: 'Definir nome e local' (Step 1) and 'Regras de segurança' (Step 2, currently selected). A message at the top of the modal reads: 'Após definir a estrutura de dados, será necessário criar regras para proteger seus dados.' with a link 'Saiba mais'.

Two options are available for starting the database:

- Iniciar no **modo de produção**: Descrição: 'Seus dados são particulares por padrão. O acesso de leitura/gravação do cliente será concedido apenas se especificado por suas regras de segurança.'
- Iniciar no **modo de teste**: Descrição: 'Por padrão, seus dados estão definidos para permitir uma configuração rápida. Porém, você precisa atualizar suas regras de segurança em até 30 dias para permitir o acesso de leitura/gravação do cliente em longo prazo.'

A red arrow points from the text 'As regras de segurança padrão para o modo de teste permitem que qualquer pessoa com a referência do seu banco de dados acesse, edite e exclua todos os dados nos próximos 30 dias.' to the 'modo de teste' section.

In the modal, there's a code snippet for Firestore security rules:

```
rules_version = '2';

service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if
        request.time < timestamp.date(2025, 6, 14);
    }
  }
}
```

A red arrow also points to the 'Criar' button at the bottom right of the modal.

Banco de dados criado

FIAP

Firebase

Visão geral do projeto |

Atalhos do projeto

Firestore Database

Categorias dos produtos

Criação

Executar

Analytics

AI

Todos os produtos

Ferramentas de desenvolvimento relacionadas

[Firebase Studio](#) [Checks](#)

LojaApp ▾

Cloud Firestore

Adicionar banco de dados

Pergunte ao Gemini como começar a usar o Firestore

Dados Regras Índices Recuperação de desastres NOVO Uso Extensions

Proteja os recursos do Cloud Firestore de abusos, como fraude de faturamento ou phishing [Configurar o App Check](#)

Conheça a edição Enterprise do Firestore, que tem compatibilidade com o MongoDB. Crie um banco de dados do Firestore Enterprise no console do Google Cloud.

[Visualização do painel](#)

(default)

+ Iniciar coleção



O objeto do nosso produto possui 03 informações

FIAP

- ID
- Title
- isChecked

Acessando a doc do firestore

FIAP

The screenshot shows the Firebase documentation interface. The top navigation bar includes 'Criar', 'Execução', 'Soluções', 'Documentos' (highlighted in orange), and 'Mais'. A search bar and a language switcher are also present. The left sidebar has sections for 'Visão geral', 'Princípios básicos', 'Criar' (selected), 'Execução', 'Referência', and 'Exemplos'. Under 'Criar', there are dropdowns for 'Authentication', 'App Check', 'Data Connect', and 'Cloud Firestore'. 'Cloud Firestore' is expanded, showing 'Introdução' (selected) and 'Começar' (highlighted with an orange rounded rectangle). Sub-items under 'Começar' include: 'Noções básicas sobre o Cloud Firestore', 'Gerenciar bancos de dados;', 'Adicionar e gerenciar dados', 'Ler dados', 'Proteger e validar dados', 'Soluções', 'Uso, limites e preços', and 'Backups e recuperação pontual'. The main content area displays steps for setting up Cloud Firestore:

1. Siga as instruções para [adicionar o Firebase ao seu app da Web](#).
2. O SDK do Cloud Firestore SDK está disponível como um pacote npm.

Below these steps is a code snippet:

```
npm install firebase@11.7.1 --save
```

It also states: "Você precisará importar o Firebase e o Cloud Firestore."

```
import { initializeApp } from "firebase/app";
import { getFirestore } from "firebase/firestore";
```

A note at the bottom says: "Está procurando uma biblioteca compacta do Firestore e precisa de apenas recursos REST/CRUD simples? Teste o [SDK do Firestore Lite](#), disponível apenas pelo npm."

Criar protótipos e fazer testes com Firebase Local Emulator Suite (opcional)

Adicionado o firestore no firebaseConfig F | A | P

The screenshot shows a code editor interface with a dark theme. On the left is a sidebar titled "Open files" containing a project structure:

- Project
 - assets
 - components
 - AssetExample.js
 - ItemLoja.js
 - services
 - firebaseConfig.js
- App.js
- package.json
- README.md

The main editor area contains a JavaScript file with the following code:

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
import {getFirestore,collection,addDoc} from 'firebase/firestore'
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries
// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyAXBfYmFXp_QRnAA0S-PhfLYRroNNjkri4",
  authDomain: "lojaapp-cd25e.firebaseio.com",
  projectId: "lojaapp-cd25e",
  storageBucket: "lojaapp-cd25e.firebaseiostorage.app",
  messagingSenderId: "215240764016",
  appId: "1:215240764016:web:e755b9b012dd2618c0a2b5"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const db = getFirestore(app)

export {[app,db,getFirestore,collection,addDoc]}
```

Two specific sections of the code are highlighted with red boxes:

- A red box surrounds the import statement for Firestore: `import {getFirestore,collection,addDoc} from 'firebase/firestore'`
- A larger red box surrounds the entire block of code starting with `// Initialize Firebase` up to the final export statement.

Criando estado para armazenar title

FIA P

```
1  import {View,TextInput,StyleSheet} from 'react-native'
2  import ItemLoja from './components/ItemLoja'
3  import React,{useState} from 'react'
4
5
6  export default function App(){
7    const[title,setTitle]=useState('')
8    return(
9      <View style={styles.container}>
10        <ItemLoja/>
11        <ItemLoja/>
12        <ItemLoja/>
13        <ItemLoja/>
14        <TextInput
15          placeholder='Digite o nome do produto'
16          style={styles.input}
17          value={title}
18          onChangeText={(value)=>setTitle(value)}
19        />
20      </View>
21    )
22  }
23  const styles = StyleSheet.create({
24    container:{
25      flex:1
26    },
27    input:{
28      backgroundColor:'lightgrey',
```

Função Salvar

FIA P

The screenshot shows a code editor interface with a dark theme. On the left is a sidebar titled 'EXPLORADOR' showing the project structure:

- TESTE43
 - .expo
 - assets
 - components
 - node_modules
 - services
 - .gitignore
 - App.js
 - app.json
 - index.js
 - package-lock.json
 - package.json

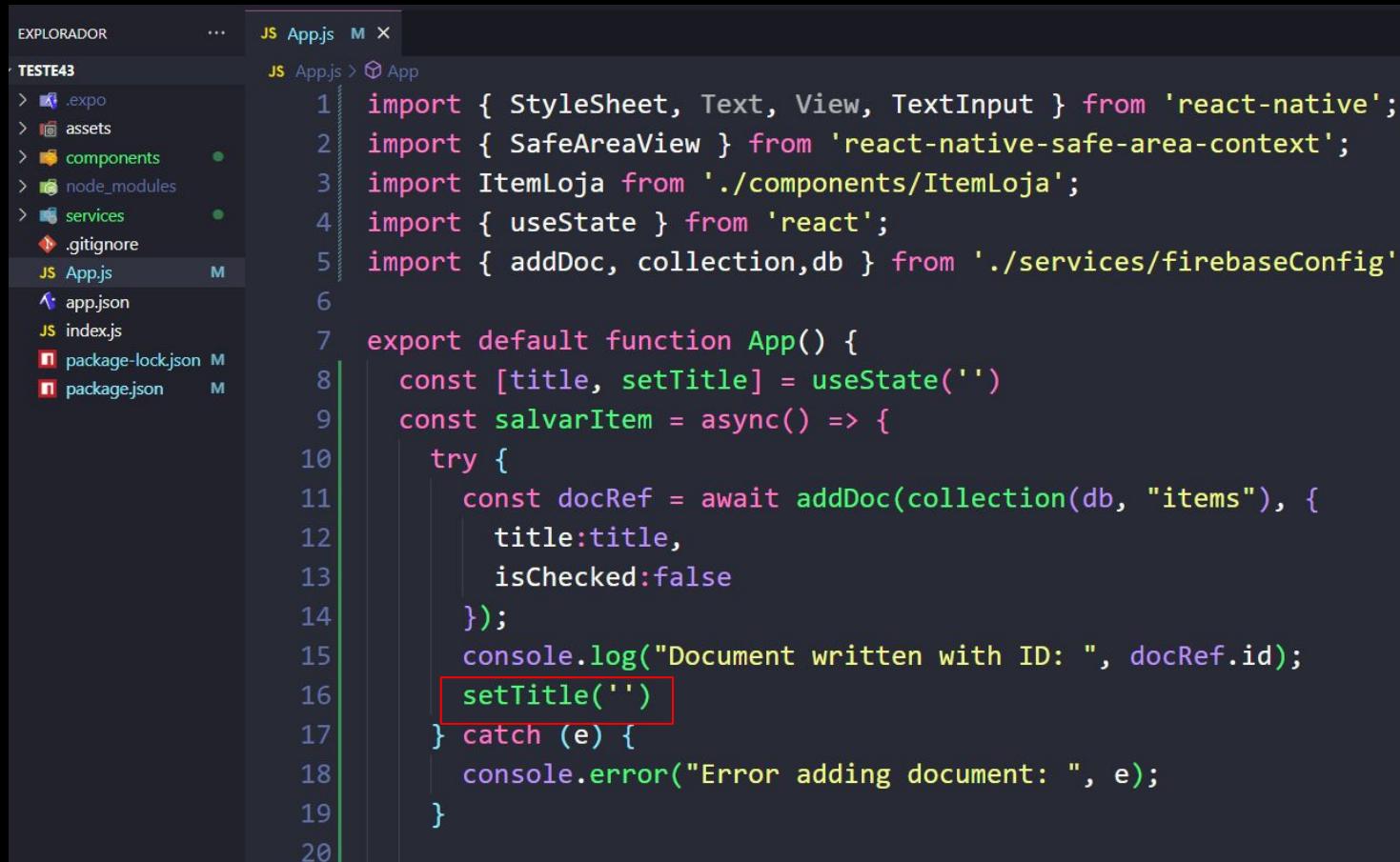
The main editor area has tabs for 'JS App.js' and 'styles'. The code is written in JavaScript and uses 'react-native' and 'react-native-safe-area-context' libraries. It imports 'useState' from 'react' and 'addDoc', 'collection', and 'db' from a local 'services/firebaseConfig' file.

```
import { StyleSheet, Text, View, TextInput } from 'react-native';
import { SafeAreaView } from 'react-native-safe-area-context';
import ItemLoja from './components/ItemLoja';
import { useState } from 'react';
import { addDoc, collection, db } from './services.firebaseioConfig'

export default function App() {
  const [title, setTitle] = useState('')
  const salvarItem = async() => {
    try {
      const docRef = await addDoc(collection(db, "items"), {
        title: title,
        isChecked: false
      });
      console.log("Document written with ID: ", docRef.id);
    } catch (e) {
      console.error("Error adding document: ", e);
    }
  }
}
```

A red rectangular box highlights the import statement for 'addDoc' and the entire function definition for 'salvarItem'. A green vertical bar is positioned to the left of the code area.

Limpar o formulário depois de salvo FIA P



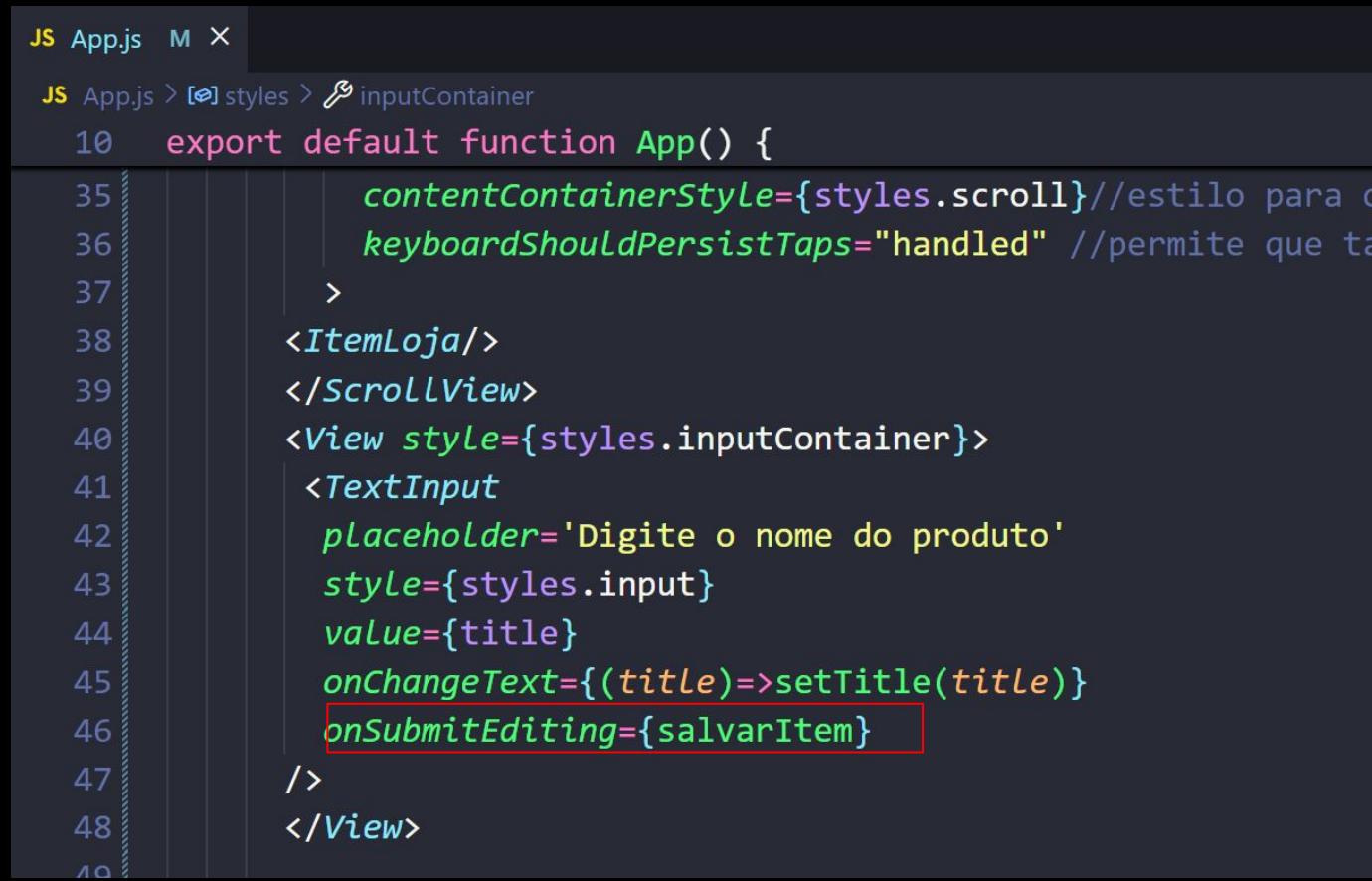
The screenshot shows a VS Code interface with the following details:

- EXPLORADOR**: Shows a project structure named "TESTE43" containing files like .expo, assets, components (with a green dot), node_modules, services (with a green dot), .gitignore, App.js (marked with a green dot), app.json, index.js, package-lock.json, and package.json.
- JS App.js M X**: The active tab, showing the code for App.js.
- JS App.js > App**: A secondary tab showing the code for App.js.
- Code Content (App.js):**

```
1 import { StyleSheet, Text, View, TextInput } from 'react-native';
2 import { SafeAreaView } from 'react-native-safe-area-context';
3 import ItemLoja from './components/ItemLoja';
4 import { useState } from 'react';
5 import { addDoc, collection, db } from './services/firebaseConfig'
6
7 export default function App() {
8     const [title, setTitle] = useState('')
9     const salvarItem = async() => {
10         try {
11             const docRef = await addDoc(collection(db, "items"), {
12                 title: title,
13                 isChecked: false
14             });
15             console.log("Document written with ID: ", docRef.id);
16             setTitle("")
17         } catch (e) {
18             console.error("Error adding document: ", e);
19         }
20     }
}
```

Chamando a função salvarItem no botão OK do teclado

FIAP



The screenshot shows a code editor window with the following details:

- File: App.js
- Line 10: `export default function App() {`
- Line 35: `contentContainerStyle={styles.scroll}//estilo para o`
- Line 36: `keyboardShouldPersistTaps="handled" //permite que ta`
- Line 37: `>`
- Line 38: `<ItemLoja/>`
- Line 39: `</ScrollView>`
- Line 40: `<View style={styles.inputContainer}>`
- Line 41: `<TextInput`
- Line 42: `placeholder='Digite o nome do produto'`
- Line 43: `style={styles.input}`
- Line 44: `value={title}`
- Line 45: `onChangeText={(title)=>setTitle(title)}`
- Line 46: `onSubmitEditing={salvarItem}` (This line is highlighted with a red border)
- Line 47: `/>`
- Line 48: `</View>`
- Line 49: (partially visible)

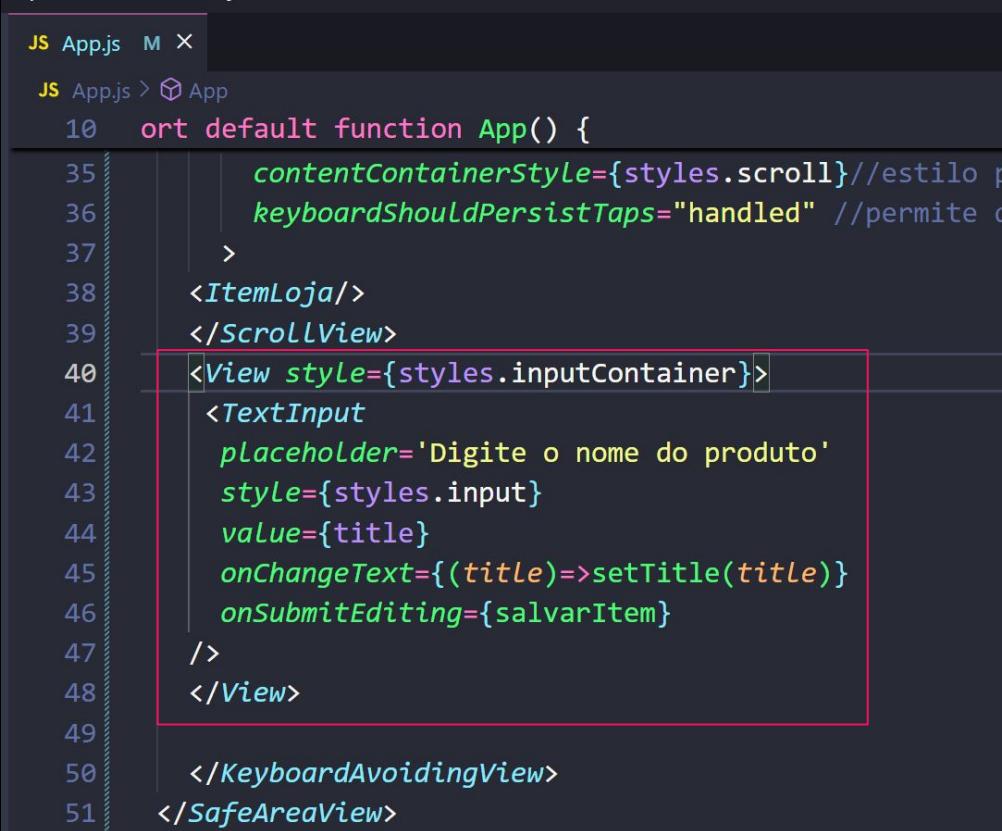
Configurando o teclado para ficar abaixo do TextInput

FIAP



```
JS App.js M X
JS App.js > ⚙ App
10  export default function App() {
25 |
26    return (
27      <SafeAreaView style={styles.container}>
28        <KeyboardAvoidingView //é um componente que ajusta o layout automaticamente
29          //para evitar que o teclado cubra os campos de entrada (TextInput).
30          style={styles.container} //usa o estilo container
31          behavior={Platform.OS === 'ios' ? 'padding' : 'height'} //no iOS usa 'padding' para
32          keyboardVerticalOffset={60} // desloca o conteúdo verticalmente (60 pixels) para
33        >
34          <ScrollView
35            contentContainerStyle={styles.scroll}//estilo para o conteúdo interno
36            keyboardShouldPersistTaps="handled" //permite que taps no teclado sejam tratados
37          >
38            <ItemLoja/>
39          </ScrollView>
```

Configurando TextInput para ficar no centro



```
JS App.js M X
JS App.js > ⚙ App
10  import React from 'react'
11  import { SafeAreaView, KeyboardAvoidingView, StyleSheet, View, Text, TextInput, Button, Alert } from 'react-native'
12  import { ItemLoja } from './ItemLoja'
13
14  const styles = StyleSheet.create({
15    scroll: {
16      flex: 1,
17      padding: 10,
18      backgroundColor: '#fff',
19      //centerContent: true
20    },
21    inputContainer: {
22      width: '100%',
23      height: 40,
24      margin: 10,
25      padding: 5,
26      border: 1px solid #ccc,
27      borderRadius: 5
28    },
29    title: {
30      color: '#007bff',
31      margin: 10
32    }
33  })
34
35  function App() {
36    const [title, setTitle] = useState('')
37
38    return (
39      <SafeAreaView style={styles.scroll}>
40        <KeyboardAvoidingView style={styles.scroll}>
41          <ScrollView style={styles.scroll}>
42            <View style={styles.inputContainer}>
43              <TextInput
44                placeholder='Digite o nome do produto'
45                style={styles.input}
46                value={title}
47                onChangeText={(title)=>setTitle(title)}
48                onSubmitEditing={salvarItem}
49              />
50            </View>
51          </ScrollView>
52        </KeyboardAvoidingView>
53      </SafeAreaView>
54    )
55  }
56
57  export default App
```

Estilo do inputContainer

The screenshot shows a code editor window with the following details:

- File:** JS App.js
- Style Scope:** styles > inputContainer
- Content:** A portion of the `StyleSheet.create` function, specifically defining styles for `inputContainer` and `input`.
- Styling Block:** The `inputContainer` style is highlighted with a red rectangular selection.
- Annotations:** A yellow circular icon with a question mark is located at the bottom left of the selected area.
- Text:**

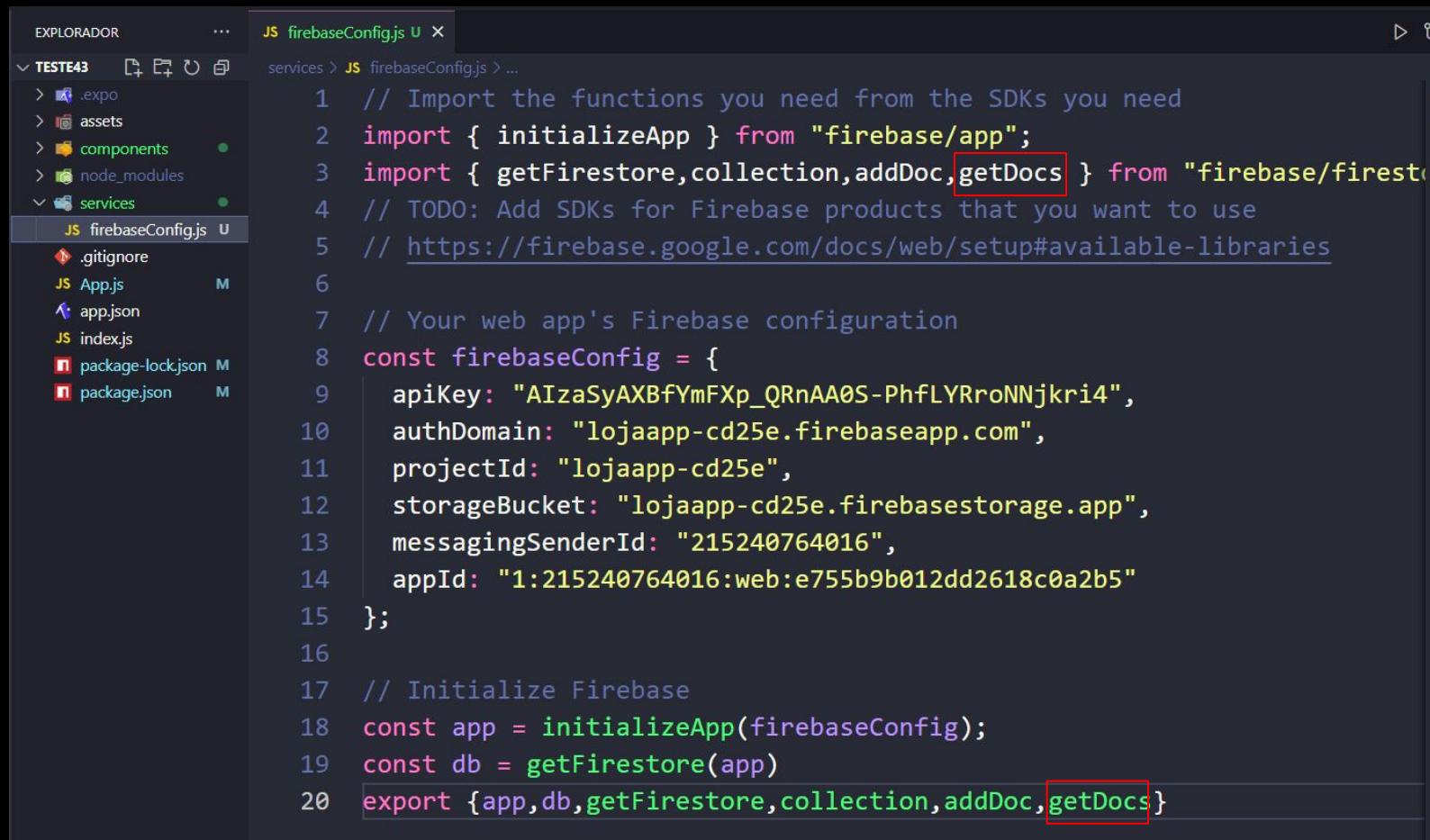
```
JS App.js M X
JS App.js > [o] styles > inputContainer
55 const styles = StyleSheet.create({
56   container: [
57     {
58       ...
59     },
60     {
61       input: {
62         backgroundColor: "lightgrey",
63         padding: 10,
64         fontSize: 15,
65         width: '90%',
66         alignSelf: "center",
67         borderRadius: 10,
68         marginTop: 'auto'
69       },
70       inputContainer: [
71         {
72           width: '100%', // Garante que o TextInput pode ocupar 90% da tela
73           alignItems: 'center', // Centraliza o TextInput
74           marginBottom: 10
75         }
76       );
77     }
78   )
79 });

79 );
```

Realizando a leitura dos dados

Realizando a leitura dos dados

FIAP

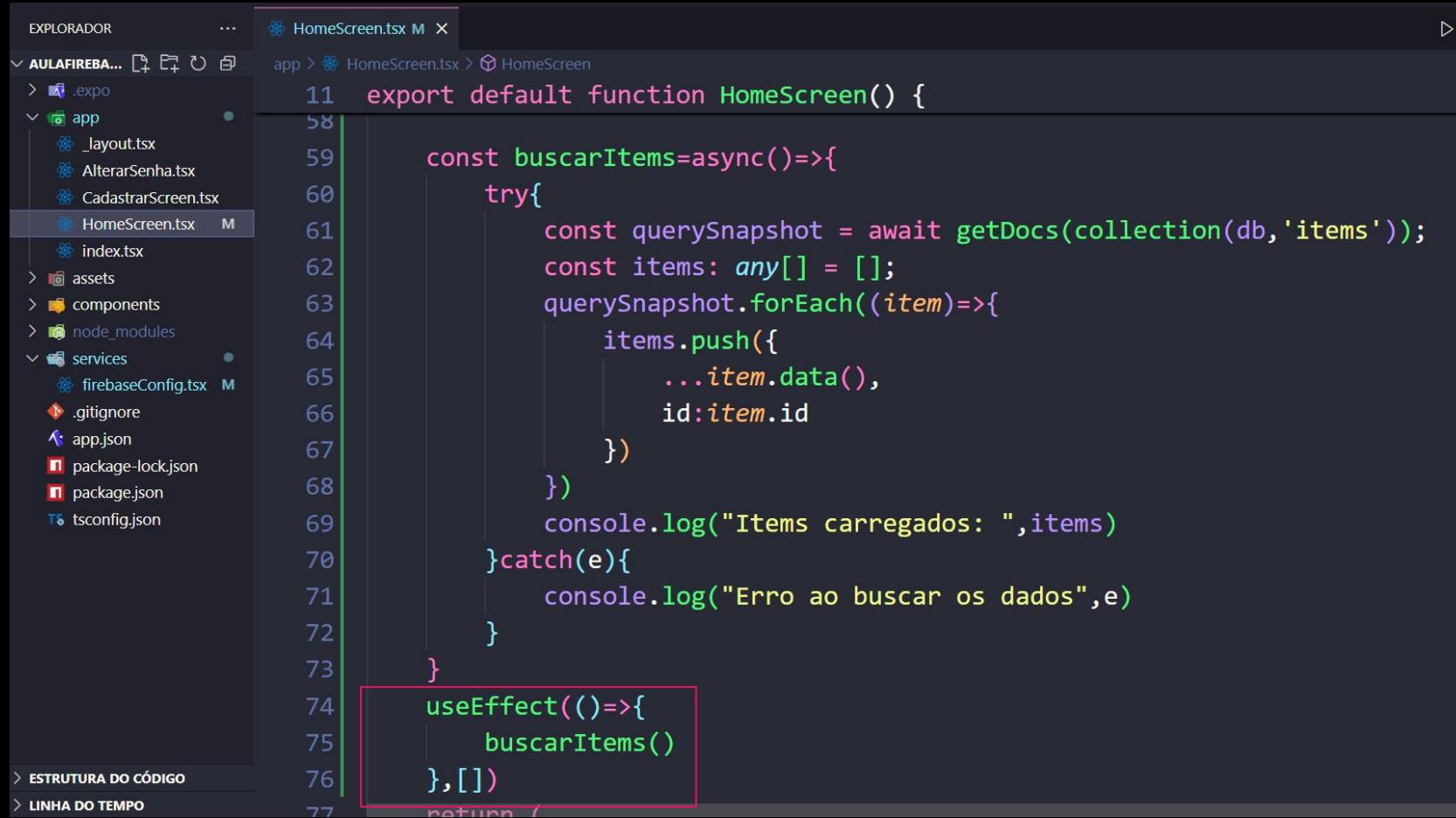


```
EXPLORADOR ... JS firebaseConfig.js ✘
TESTE43 .expo assets components node_modules services
  JS firebaseConfig.js
  .gitignore
  JS App.js M
  app.json
  JS index.js
  package-lock.json M
  package.json M

services > JS firebaseConfig.js > ...
1 // Import the functions you need from the SDKs you need
2 import { initializeApp } from "firebase/app";
3 import { getFirestore,collection,addDoc,getDocs } from "firebase/firestore"
4 // TODO: Add SDKs for Firebase products that you want to use
5 // https://firebase.google.com/docs/web/setup#available-libraries
6
7 // Your web app's Firebase configuration
8 const firebaseConfig = {
9   apiKey: "AIzaSyAXBfYmFXp_QRnAA0S-PhfLYRroNNjkri4",
10  authDomain: "lojaapp-cd25e.firebaseio.com",
11  projectId: "lojaapp-cd25e",
12  storageBucket: "lojaapp-cd25e.firebaseiostorage.app",
13  messagingSenderId: "215240764016",
14  appId: "1:215240764016:web:e755b9b012dd2618c0a2b5"
15 };
16
17 // Initialize Firebase
18 const app = initializeApp(firebaseConfig);
19 const db = getFirestore(app)
20 export {app,db,getFirestore,collection,addDoc,getDocs}
```

Função buscarDados

FIA P



The screenshot shows a code editor interface with the following details:

- EXPLORADOR**: Shows the project structure: AULAFIREBASE... (root), app (containing .expo, _layout.tsx, AlterarSenha.tsx, CadastrarScreen.tsx, HomeScreen.tsx, index.tsx), assets, components, node_modules, services (containing firebaseConfig.tsx), .gitignore, app.json, package-lock.json, package.json, and tsconfig.json.
- HomeScreen.tsx M X**: The active file tab.
- Path**: app > HomeScreen.tsx > HomeScreen
- Code Content** (Lines 11 to 77):

```
11  export default function HomeScreen() {  
58  
59      const buscarItems=async()=>{  
60          try{  
61              const querySnapshot = await getDocs(collection(db,'items'));  
62              const items: any[] = [];  
63              querySnapshot.forEach((item)=>{  
64                  items.push({  
65                      ...item.data(),  
66                      id:item.id  
67                  })  
68              })  
69              console.log("Items carregados: ",items)  
70          }catch(e){  
71              console.log("Erro ao buscar os dados",e)  
72          }  
73      }  
74      useEffect(()=>{  
75          buscarItems()  
76      },[  
77  ])
```

A red rectangular box highlights the line `useEffect(()=>{` starting at line 74.

- Bottom Navigation**: ESTRUTURA DO CÓDIGO, LINHA DO TEMPO.

Criando interface e estado para armazenamento

FIAP

The screenshot shows a code editor with a dark theme. On the left is the Explorer sidebar showing project files like .expo, app, services, and firebaseConfig.tsx. The main editor area has the following code:

```
import { deleteUser } from "firebase/auth";
import { collection, addDoc, db, getDocs } from "../services/firebaseConfig"

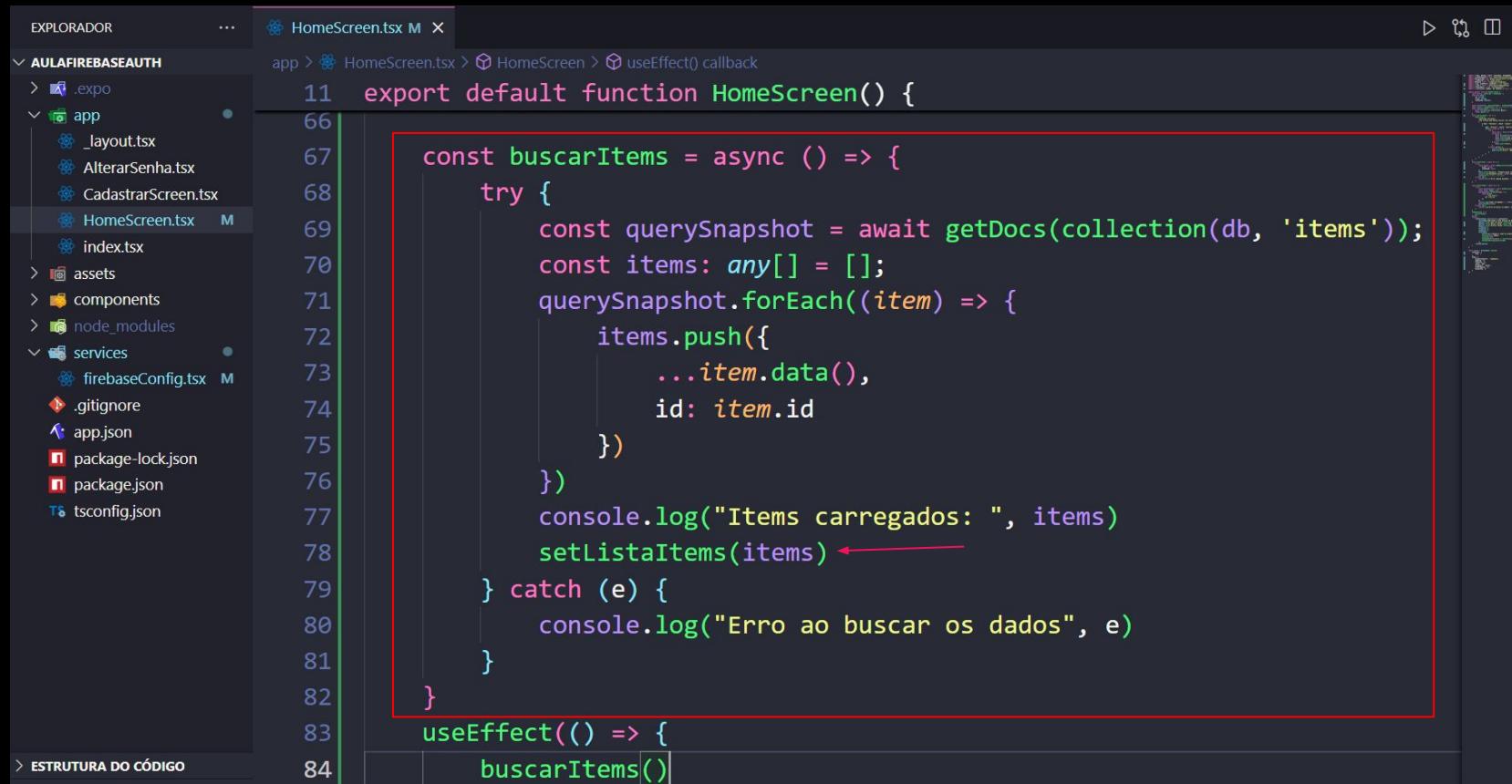
export default function HomeScreen() {
  const [title, setTitle] = useState('')
  interface Item {
    id: string;
    title: string;
    isChecked: boolean;
  }
  const [listaItems, setListaItems] = useState<Item[]>([])
  const router = useRouter() // Hook de navegação entre telas
  const realizarLogoff = async () => {
    await AsyncStorage.removeItem('@user')
    router.push('/')
  }
  const excluirConta = () => {
    Alert.alert(
      "Confirmar Exclusão",
      "Tem certeza que deseja excluir sua conta?"
    )
  }
}

const realizarLogoff = () => {
  const userRef = collection(db, "users").doc(user.uid)
  userRef.delete()
  AsyncStorage.removeItem('@user')
  router.push('/')
}
```

A red box highlights the interface definition and the useState call for listaItems. A yellow lightbulb icon is on line 24, indicating a potential issue.

Criando interface e estado para armazenamento

FIAP



The screenshot shows a code editor with a dark theme. On the left is the Explorer sidebar showing project files like .expo, app, assets, components, node_modules, services, and various configuration files. The main editor area has a breadcrumb navigation bar: app > HomeScreen.tsx > HomeScreen > useEffect() callback. The code itself is a TypeScript function named HomeScreen. A red box highlights the useEffect() callback section, specifically the part where it calls the buscarItems() function.

```
11  export default function HomeScreen() {
66
67    const buscarItems = async () => {
68      try {
69        const querySnapshot = await getDocs(collection(db, 'items'));
70        const items: any[] = [];
71        querySnapshot.forEach((item) => {
72          items.push({
73            ...item.data(),
74            id: item.id
75          })
76        })
77        console.log("Items carregados: ", items)
78        setListaItems(items) ←
79      } catch (e) {
80        console.log("Erro ao buscar os dados", e)
81      }
82    }
83
84    useEffect(() => {
      buscarItems()
    })
  }
```

Criando FlatList para renderizar e passando props 'title'

FIAP

The screenshot shows a code editor interface with a dark theme. On the left is a file explorer sidebar containing project files like .expo, assets, components (with itemLoja.js), node_modules, services (with firebaseCo...), .gitignore, App.js (selected), app.json, index.js, package-lock.json, and package.json. The main editor area has tabs for JS App.js and M. The code is as follows:

```
12  export default function App() {  
13    return (  
14      <KeyboardAvoidingView //é um componente que ajusta o layout automaticamente para evitar que o teclado cubra os campos de entrada (TextInput)  
15        style={styles.container} //usa o estilo container  
16        behavior={Platform.OS === 'ios' ? 'padding' : 'height'} //no iOS  
17        keyboardVerticalOffset={60} // desloca o conteúdo verticalmente (60 pixels)  
18      >  
19        {listItems.length <= 0 ? (  
20          <ActivityIndicator />  
21        ) : (  
22          <FlatList  
23            data={listItems}  
24            renderItem={({ item }) => <ItemLoja title={item.title} />}  
25          >  
26        )  
27      }  
28    <ScrollView  
29      contentContainerStyle={styles.scrollViewContent}  
30    >  
31  }
```

A red rectangular box highlights the FlatList component and its properties.

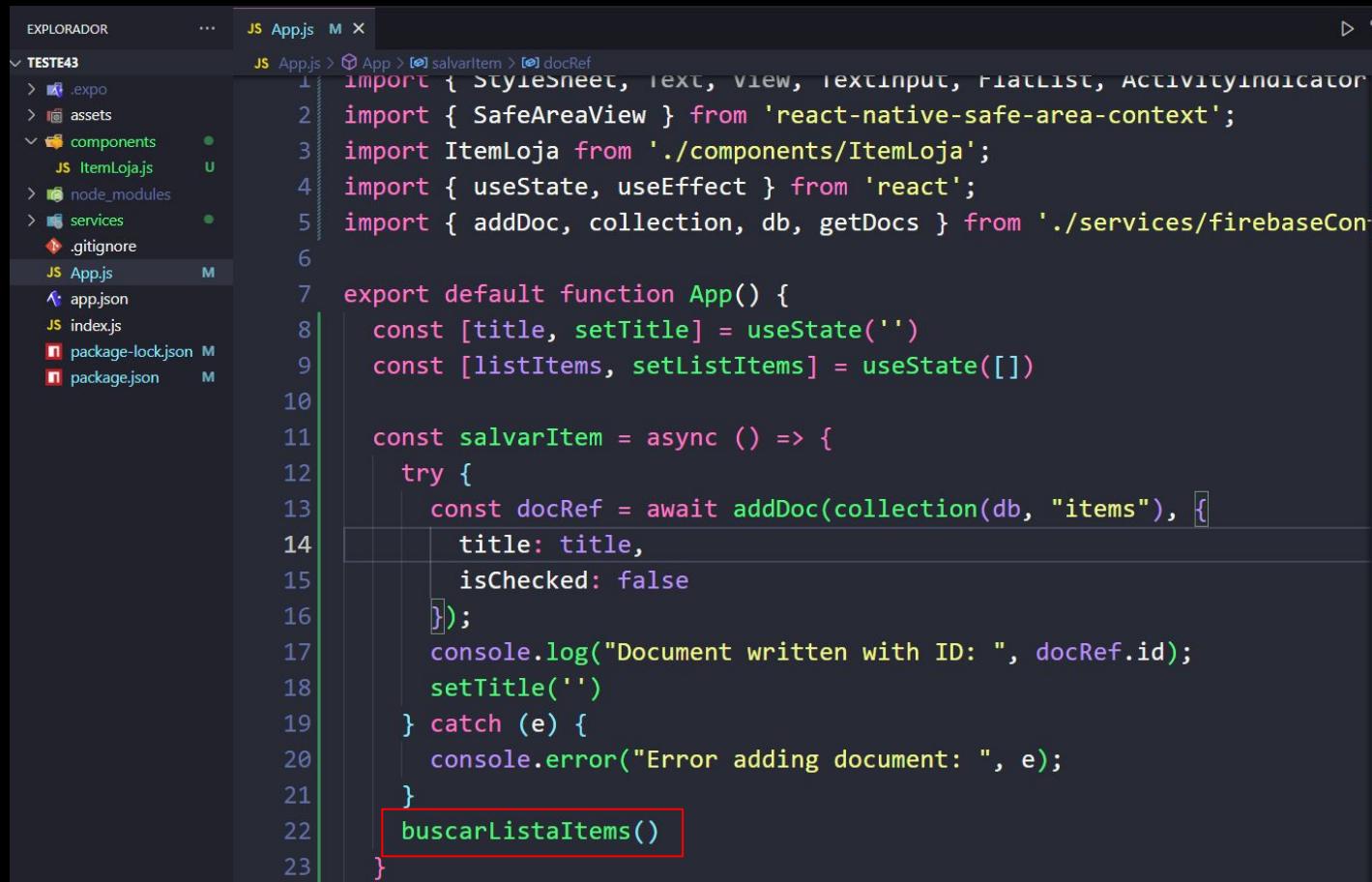
Pegando o props no comp ItemLoja

FIAP

```
EXPLORADOR ... ItemLoja.tsx M X
AULAFIREBA... D E U A
> expo
  app
    _layout.tsx
    AlterarSenha.tsx
    CadastrarScreen.tsx
    HomeScreen.tsx M
    index.tsx
> assets
  components
    ItemLoja.tsx M
> node_modules
  services
    firebaseConfig.tsx M
    .gitignore
    app.json
    package-lock.json
    package.json
    tsconfig.json
ESTRUTURA DO CÓDIGO
LINHA DO TEMPO
components > ItemLoja.tsx > ItemLoja
1 import { StyleSheet, View, Text, Pressable } from "react-native";
2 import { AntDesign, MaterialIcons } from '@expo/vector-icons'
3
4 export default function ItemLoja(props: any){
5   return(
6     <View style={styles.container}>
7       <Pressable>
8         <AntDesign name="checkcircleo" color='black' size={24}/>
9       </Pressable>
10      <Text style={styles.title}>{props.title}</Text>
11      <Pressable>
12        <MaterialIcons name='delete' size={24} color='black'/>
13      </Pressable>
14    </View>
15  )
16}
17
18 const styles = StyleSheet.create({
19   container: {
20     flexDirection: 'row',
```

Chamando função após salvar produto

FIAP



```
EXPLORADOR JS App.js M X
JS App.js > ⚡ App > [o] salvarItem > [o] docRef
1 import { StyleSheet, Text, View, TextInput, FlatList, ActivityIndicator } from 'react-native';
2 import { SafeAreaView } from 'react-native-safe-area-context';
3 import ItemLoja from './components/ItemLoja';
4 import { useState, useEffect } from 'react';
5 import { addDoc, collection, db, getDocs } from './services.firebaseioConfig';
6
7 export default function App() {
8   const [title, setTitle] = useState('')
9   const [listItems, setListItems] = useState([])
10
11   const salvarItem = async () => {
12     try {
13       const docRef = await addDoc(collection(db, "items"), [
14         title,
15         isChecked: false
16       ]);
17       console.log("Document written with ID: ", docRef.id);
18       setTitle('')
19     } catch (e) {
20       console.error("Error adding document: ", e);
21     }
22     buscarListaItems()
23   }
}
```

Configurando o isChecked

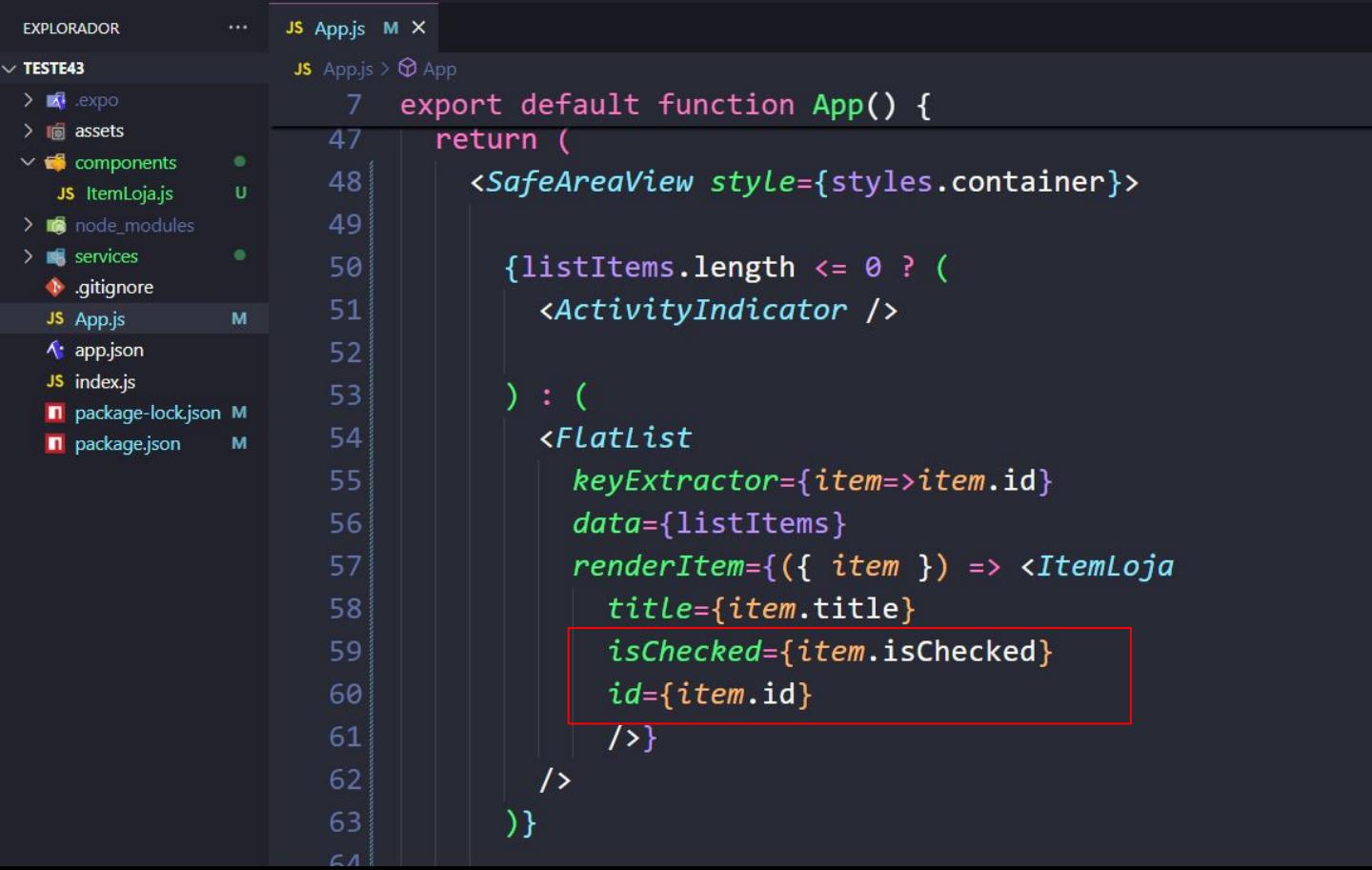
FIAP

The screenshot shows a dark-themed interface of the Visual Studio Code code editor. On the left is the Explorer sidebar with project files: TESTE43, components (with ItemLoja.js selected), node_modules, services, .gitignore, App.js, app.json, index.js, package-lock.json, and package.json. The main editor area displays the ItemLoja.js file. The code uses React Native components like StyleSheet, View, Text, Pressable, AntDesign, and MaterialIcons. A specific line of code, `const[isChecked, setIsChecket]=useState(props.isChecked)`, is highlighted with a red rectangle. The code is as follows:

```
1 import { StyleSheet, View, Text, Pressable } from 'react-native'
2 import { AntDesign, MaterialIcons } from '@expo/vector-icons'
3 import { useState } from 'react'
4
5 export default function ItemLoja(props) {
6   const[isChecked, setIsChecket]=useState(props.isChecked)
7   return (
8     <View style={styles.container}>
9       <Pressable>
10         <AntDesign name='checkcircleo' color='black' size={24} />
11       </Pressable>
12       <Text style={styles.title}>{props.title}</Text>
13       <Pressable>
14         <MaterialIcons name='delete' size={24} color='black' />
15       </Pressable>
16
17     </View>
18
19   )
20 }
```

Passando os dados para o comp ItemLoja

FIA P



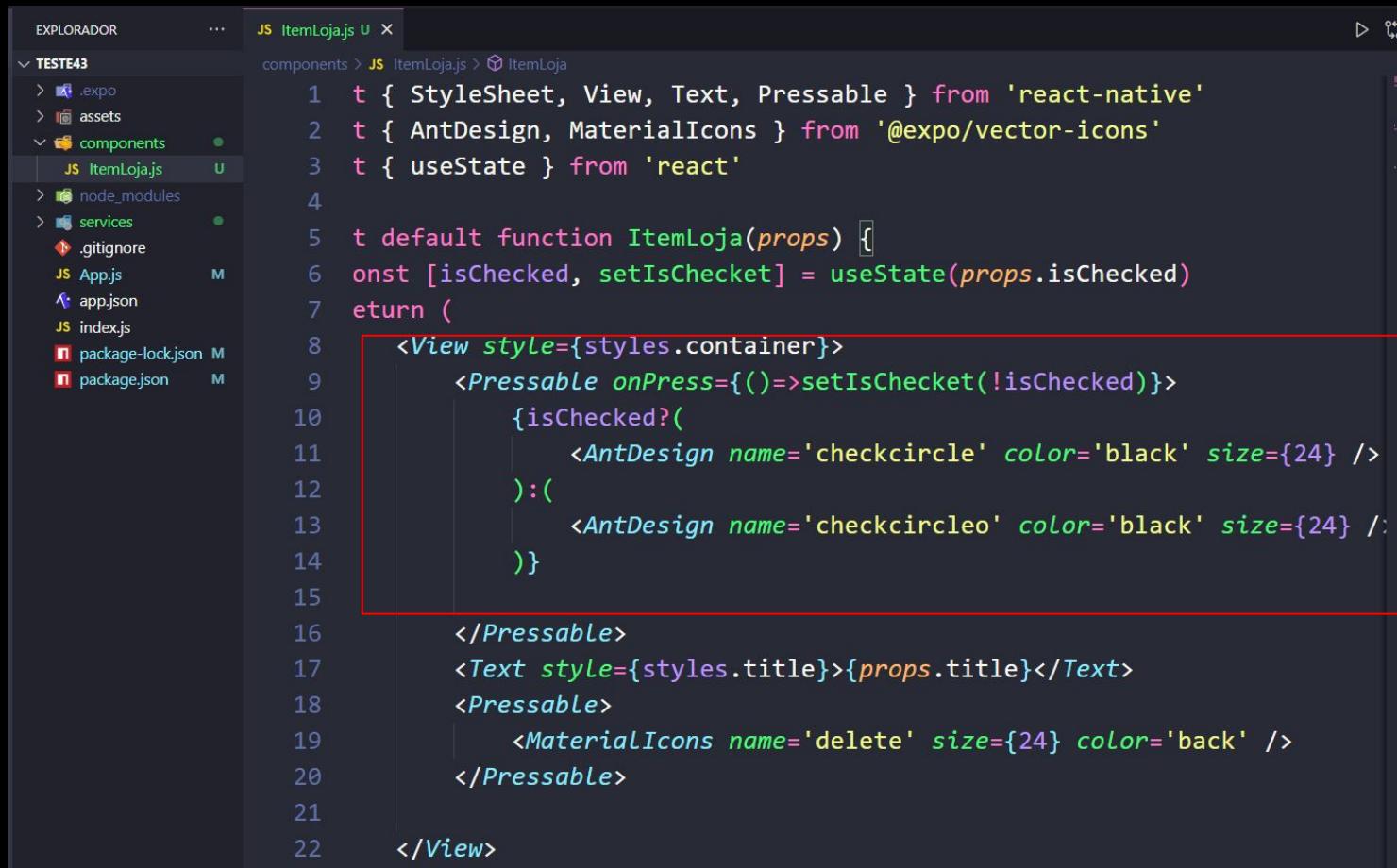
The screenshot shows the VS Code interface with the following details:

- Explorer View:** Shows a project structure named "TESTE43" containing files like .expo, assets, components (with ItemLoja.js), node_modules, services, .gitignore, App.js, app.json, index.js, package-lock.json, and package.json.
- Code Editor:** The file "App.js" is open, displaying the following code:

```
7  export default function App() {
47    return (
48      <SafeAreaView style={styles.container}>
49
50        {listItems.length <= 0 ? (
51          <ActivityIndicator />
52        ) : (
53          <FlatList
54            keyExtractor={item=>item.id}
55            data={listItems}
56            renderItem={({ item }) => <ItemLoja
57              title={item.title}
58              isChecked={item.isChecked}
59              id={item.id}
60            />}
61          />
62        )}
63    )
64  }
```
- Selection:** A red rectangular selection highlights the props passed to the `<ItemLoja>` component: `isChecked={item.isChecked}` and `id={item.id}`.

Lógica para alterar o ícone

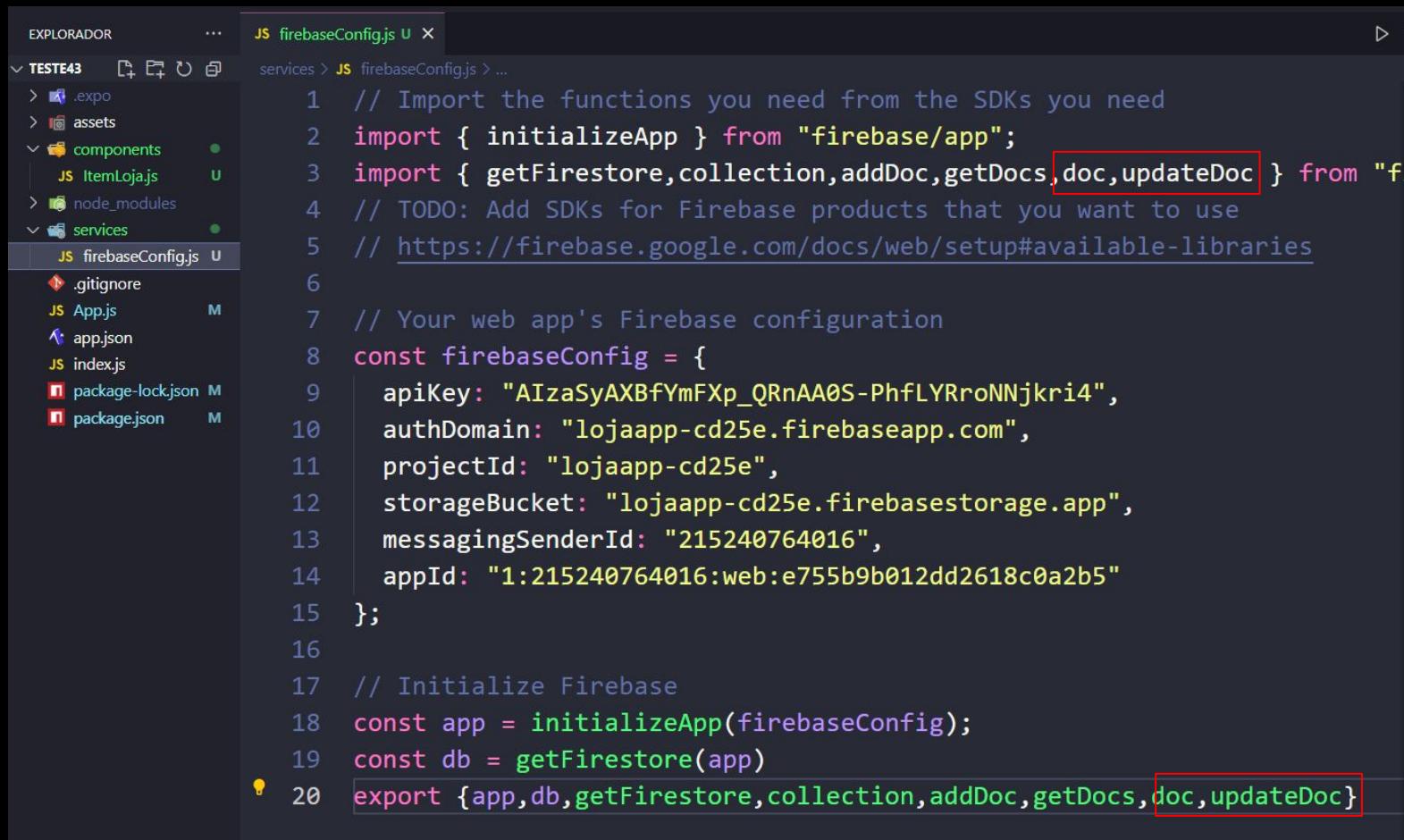
FIAP



```
EXPLORADOR JS ItemLoja.js U X
TESTE43 components > JS ItemLoja.js > ItemLoja
JS ItemLoja.js
components > JS ItemLoja.js > ItemLoja
1 t { StyleSheet, View, Text, Pressable } from 'react-native'
2 t { AntDesign, MaterialIcons } from '@expo/vector-icons'
3 t { useState } from 'react'
4
5 t default function ItemLoja(props) {
6 onst [isChecked, setIsChecket] = useState(props.isChecked)
7 eturn (
8     <View style={styles.container}>
9         <Pressable onPress={()=>setIsChecket(!isChecked)}>
10            {isChecked?(
11                <AntDesign name='checkcircle' color='black' size={24} />
12            ):(
13                <AntDesign name='checkcircleo' color='black' size={24} />
14            )}
15        </Pressable>
16        <Text style={styles.title}>{props.title}</Text>
17        <Pressable>
18            <MaterialIcons name='delete' size={24} color='back' />
19        </Pressable>
20
21
22    </View>
```

Desenvolvendo o update

FIAP



The image shows a screenshot of a code editor with a dark theme. On the left, the file structure of a project named "TESTE43" is visible in the Explorer panel. The "services" folder contains a file named "firebaseConfig.js". The main editor area displays the contents of this file:

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
import { getFirestore, collection, addDoc, getDocs, doc, updateDoc } from "f
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyAXBfYmFXp_QRnAA0S-PhfLYRroNNjkri4",
  authDomain: "lojaapp-cd25e.firebaseio.com",
  projectId: "lojaapp-cd25e",
  storageBucket: "lojaapp-cd25e.firebaseiostorage.app",
  messagingSenderId: "215240764016",
  appId: "1:215240764016:web:e755b9b012dd2618c0a2b5"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const db = getFirestore(app)
export {app, db, getFirestore, collection, addDoc, getDocs, doc, updateDoc}
```

The code imports the necessary functions from the Firebase SDKs and defines the Firebase configuration object. It then initializes the app and gets a reference to the Firestore database. Finally, it exports the app, database, and various Firestore-related functions.

EXPLORADOR ...

TESTE43 .expo assets components ItemLoja.js node_modules services .gitignore App.js app.json index.js package-lock.json package.json

JS ItemLoja.js U x

components > JS ItemLoja.js > ItemLoja

```
1 import { StyleSheet, View, Text, Pressable } from 'react-native'
2 import { AntDesign, MaterialIcons } from '@expo/vector-icons'
3 import { useState, useEffect } from 'react'
4 import { db, doc, updateDoc } from '../services/firebaseConfig'

5
6 export default function ItemLoja(props) {
7     const [isChecked, setIsChecket] = useState(props.isChecked)

9     const updateIsChecked = async () => {
10         const itemRef = doc(db, "items", props.id);

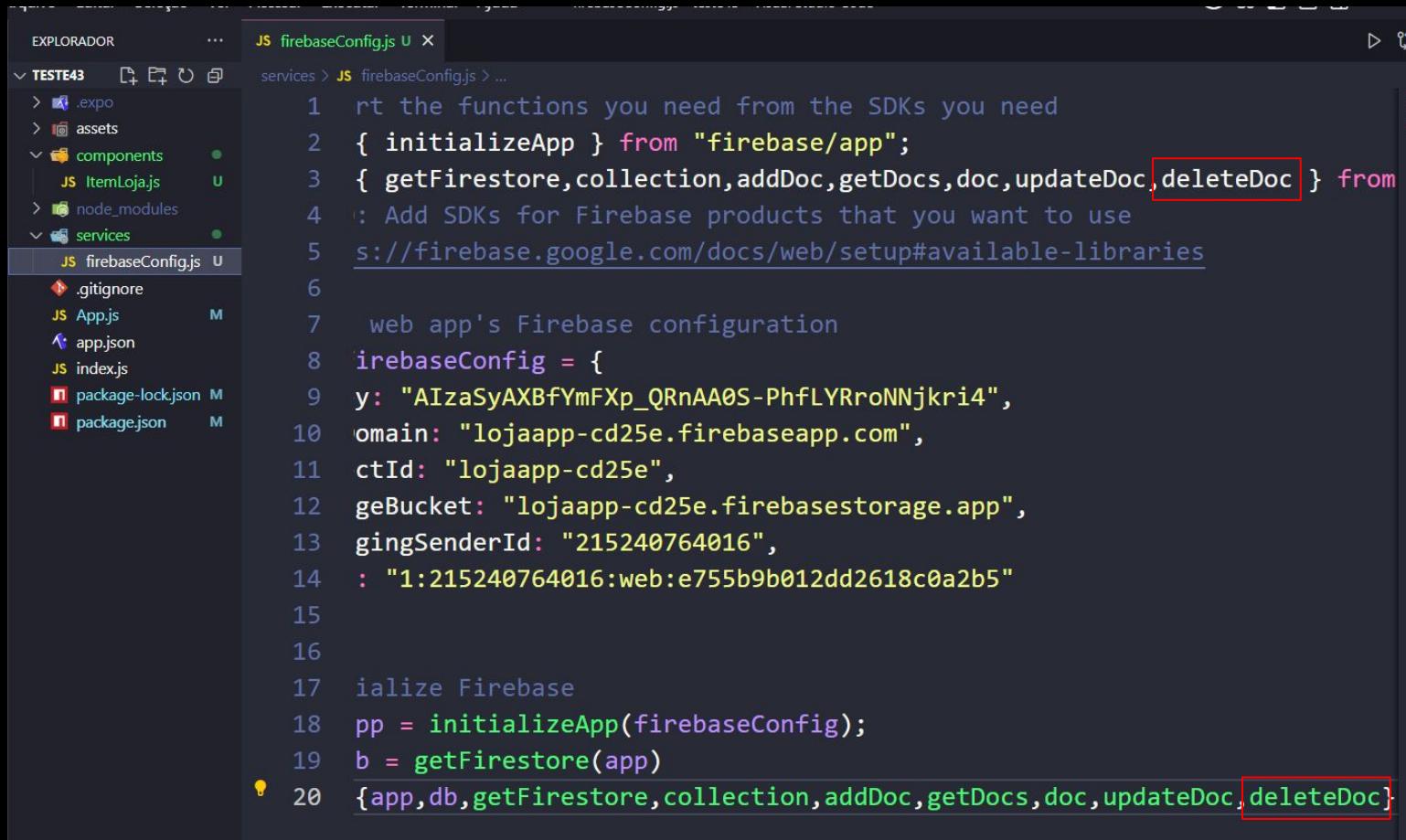
12         // Set the "capital" field of the city 'DC'
13         await updateDoc(itemRef, {
14             isChecked: isChecked,
15         })
16     }

18     useEffect(() => {
19         updateIsChecked();
20     }, [isChecked]);

22     return (
23         <View style={styles.container}>
24             <Pressable onPress={() => setIsChecket(!isChecked)}>
```

Desenvolvendo o delete

FIAP

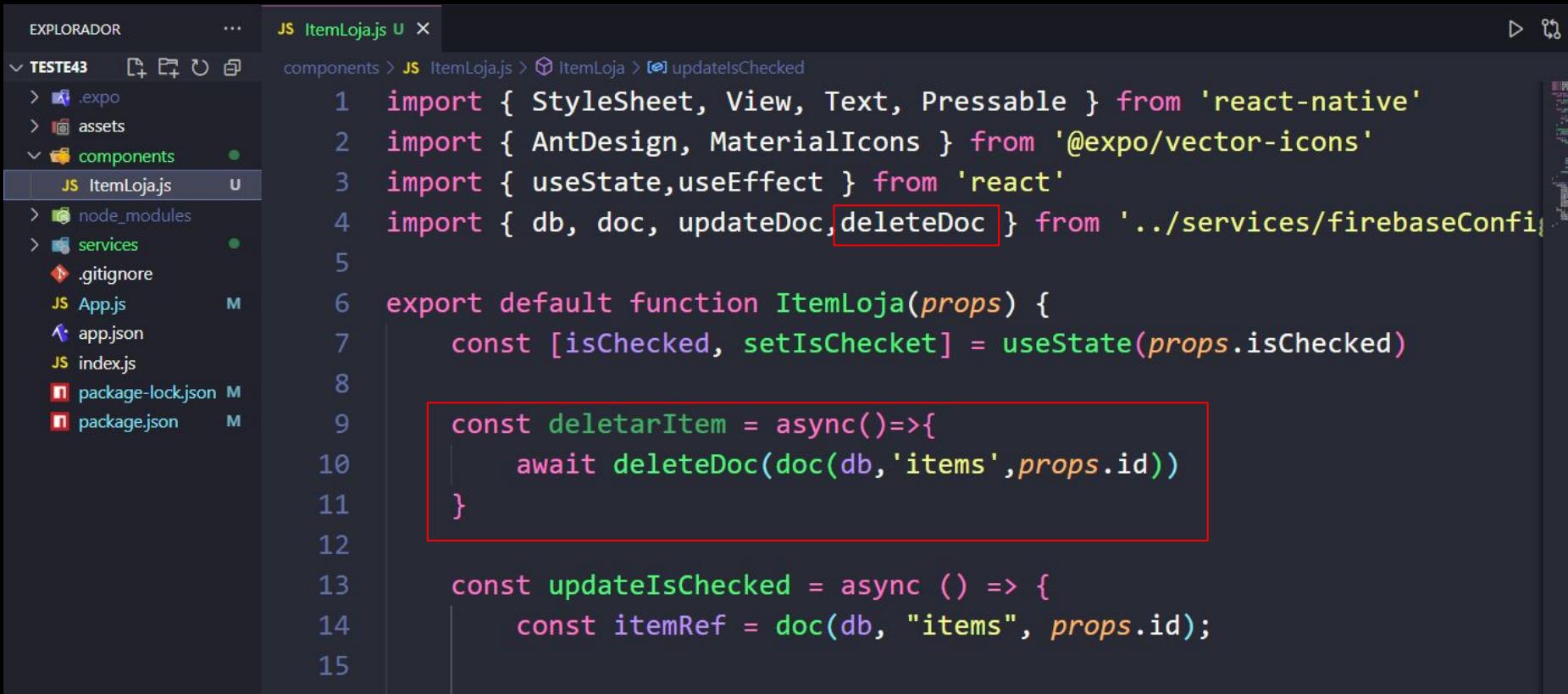


```
EXPLORADOR JS firebaseConfig.js X
TESTE43 .expo assets components ItemLoja.js node_modules services
JS firebaseConfig.js gitignore App.js M app.json index.js package-lock.json M package.json M

services > JS firebaseConfig.js > ...
1  rt the functions you need from the SDKs you need
2  { initializeApp } from "firebase/app";
3  { getFirestore,collection,addDoc,getDocs,doc,updateDoc,deleteDoc } from
4  : Add SDKs for Firebase products that you want to use
5  s://firebase.google.com/docs/web/setup#available-libraries
6
7  web app's Firebase configuration
8  iirebaseConfig = {
9  y: "AIzaSyAXBfYmFXp_QRnAA0S-PhfLYRroNNjkri4",
10 omain: "lojaapp-cd25e.firebaseio.com",
11 ctId: "lojaapp-cd25e",
12 geBucket: "lojaapp-cd25e.firebaseiostorage.app",
13 gingSenderId: "215240764016",
14 : "1:215240764016:web:e755b9b012dd2618c0a2b5"
15
16
17 ialize Firebase
18 pp = initializeApp(firebaseConfig);
19 b = getFirestore(app)
20 {app,db,getFirestore,collection,addDoc,getDocs,doc,updateDoc,deleteDoc}
```

Função deletar

FIA P



The screenshot shows a VS Code editor window with the following details:

- EXPLORADOR**: Shows the project structure: TESTE43, components (containing ItemLoja.js), node_modules, services, .gitignore, App.js, app.json, index.js, package-lock.json, and package.json.
- ItemLoja.js** (selected tab):

```
1 import { StyleSheet, View, Text, Pressable } from 'react-native'
2 import { AntDesign, MaterialIcons } from '@expo/vector-icons'
3 import { useState, useEffect } from 'react'
4 import { db, doc, updateDoc, deleteDoc } from '../services/firebaseConfig'
5
6 export default function ItemLoja(props) {
7     const [isChecked, setIsChecket] = useState(props.isChecked)
8
9     const deletarItem = async()=>{
10         await deleteDoc(doc(db,'items',props.id))
11     }
12
13     const updateIsChecked = async () => {
14         const itemRef = doc(db, "items", props.id);
15     }
}
```

The code highlights the `deleteDoc` import and the `deletarItem` function body with red boxes, indicating they are the focus of the current discussion.

Passando a leitura via props

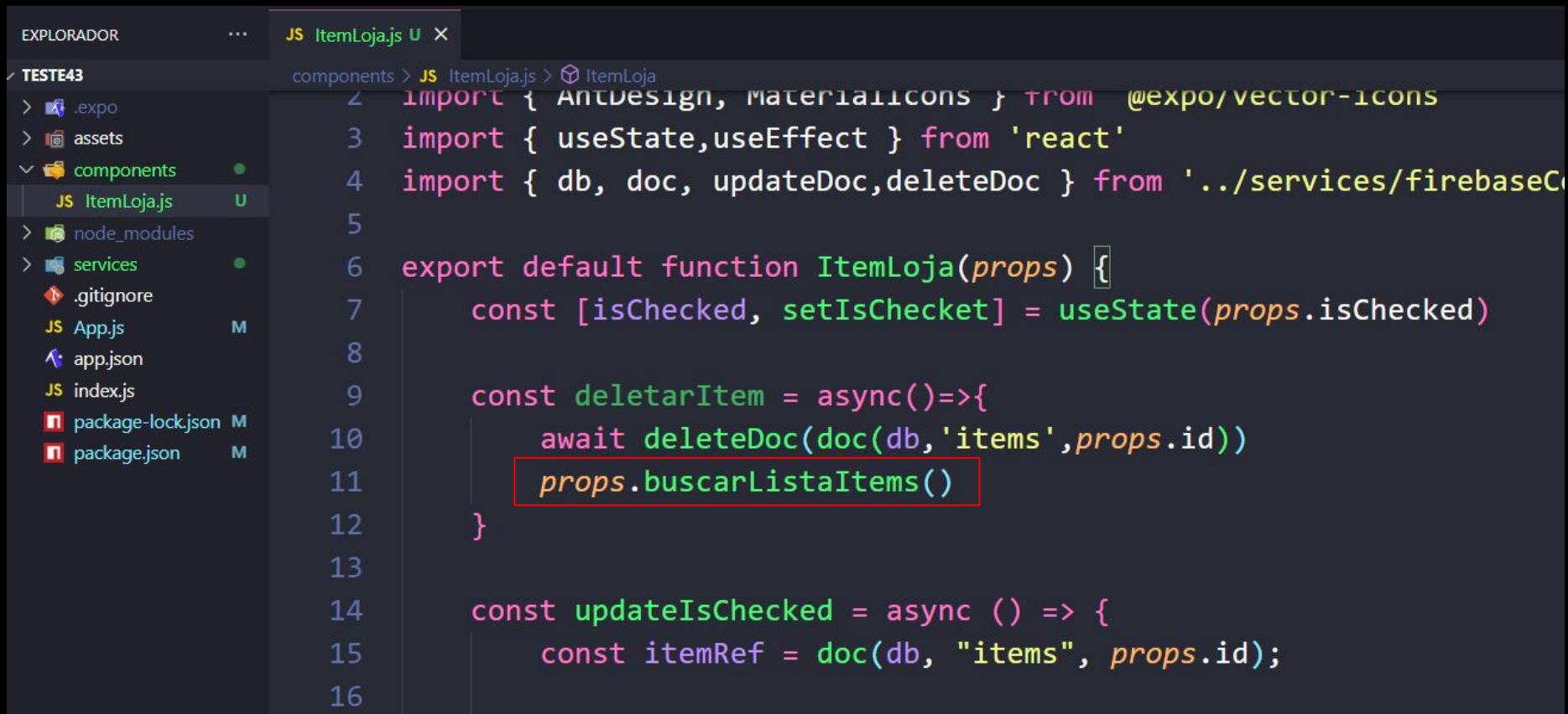
FIAP

The screenshot shows a code editor interface with a dark theme. On the left is the file explorer (EXPLORADOR) showing files like .expo, assets, components (with ItemLojajs selected), node_modules, services, .gitignore, App.js (selected), app.json, index.js, package-lock.json, and package.json. The main area is a code editor titled 'JS App.js' under 'App'. The code is:

```
1  export default function App() {
2      <ActivityIndicator />
3
4      ) : (
5          <FlatList
6              keyExtractor={item=>item.id}
7              data={listItems}
8              renderItem={({ item }) => <ItemLoja
9                  title={item.title}
10                 isChecked={item.isChecked}
11                 id={item.id}
12                 buscarListaItems={buscarListaItems}
13             />}
14         )
15     }
16 }
```

The line 'buscarListaItems={buscarListaItems}' is highlighted with a red border.

Fará buscar dados após a exclusão do item F|A|P



```
EXPLORADOR ... JS ItemLoja.js ×
TESTE43 components > JS ItemLoja.js > ItemLoja
1 import { AntDesign, MaterialIcons } from '@expo/vector-icons'
2
3 import { useState, useEffect } from 'react'
4 import { db, doc, updateDoc, deleteDoc } from '../services.firebaseioC
5
6 export default function ItemLoja(props) {
7     const [isChecked, setIsChecked] = useState(props.isChecked)
8
9     const deletarItem = async () => {
10         await deleteDoc(doc(db, 'items', props.id))
11         props.buscarListaItems()
12     }
13
14     const updateIsChecked = async () => {
15         const itemRef = doc(db, "items", props.id);
16     }
}
```

Acionando a função no ícone

FIAP

```
EXPLORADOR ... JS ItemLoja.js ✎
TESTE43
> .expo
> assets
> components
  > ItemLoja.js ✎
> node_modules
> services
  > .gitignore
JS App.js M
app.json
JS index.js
package-lock.json M
package.json M

components > JS ItemLoja.js ✎ ItemLoja
6  export default function ItemLoja(props) {
7    ...
8
9    useEffect(() => {
10      updateIsChecked();
11    }, [isChecked]);
12
13    return (
14      <View style={styles.container}>
15        <Pressable onPress={() => setIsChecket(!isChecked)}>
16          {isChecked ? (
17            <AntDesign name='checkcircle' color='black' size={24} />
18          ) : (
19            <AntDesign name='checkcircleo' color='black' size={24} />
20          )}
21
22        </Pressable>
23        <Text style={styles.title}>{props.title}</Text>
24        <Pressable onPress={deletarItem}>
25          <MaterialIcons name='delete' size={24} color='black' />
26        </Pressable>
27
28      </View>
29    )
30  )
31
32  const styles = StyleSheet.create({
33    container: {
34      flex: 1,
35      padding: 16,
36      ...
37    },
38    title: {
39      ...
40    }
41  })
42
```

Dúvidas?