

Semana Integrada da Computação 2025

# React Native

---

Introdução ao desenvolvimento de aplicativos mobile  
com React Native

# Roteiro

- Apresentação
- React Native
- Expo
- Ferramentas e instalação
- Componentes
- Propriedades
- Estado
- Navegação
- Bibliotecas
- EAS e Build

## Apresentação **Vinicius Stefanes**

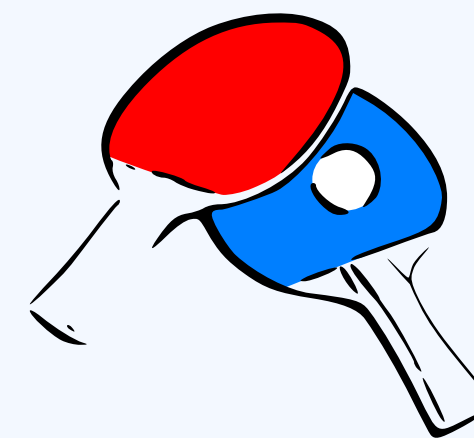


<https://github.com/viniciusSt1>



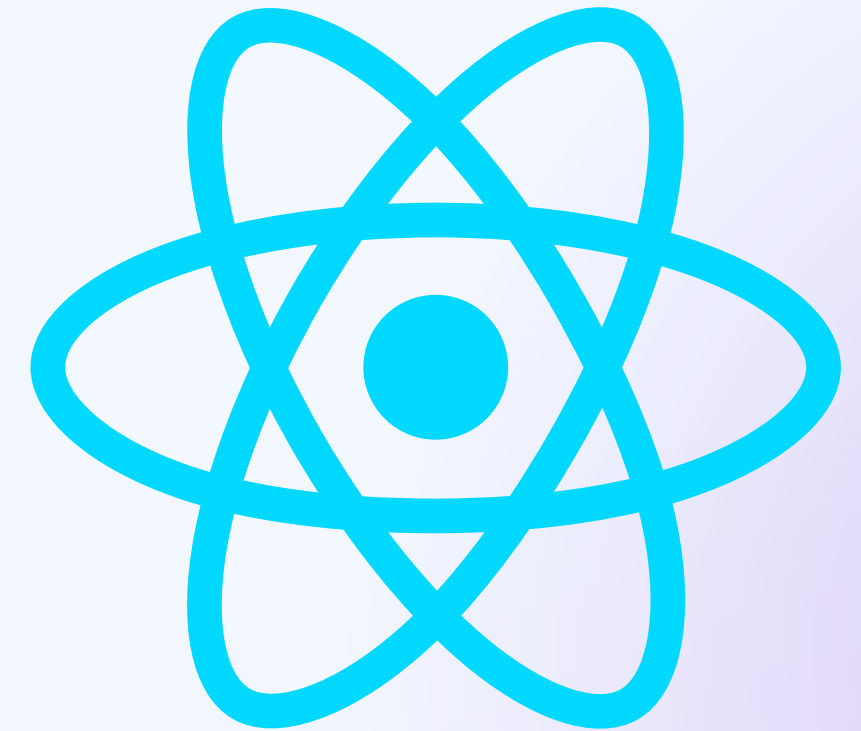
<https://www.linkedin.com/in/vinicius-stefanes-73a502189/>

- Décimo semestre de Engenharia de Computação
- Pesquisador Criptografia e Blockchain CPQD / PUCPR
- Desenvolvedor de software back/front-end
- CC's Pongas Player



# React Native ??

React Native lets you create truly native apps and doesn't compromise your users' experiences. It provides a core set of platform agnostic native components like View, Text, and Image that map directly to the platform's native UI building blocks.



<https://reactnative.dev/>

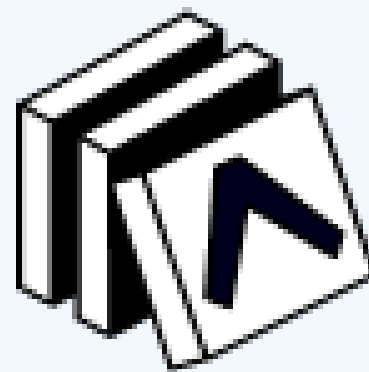


# Expo ??

We're a full ecosystem of tools that help you write, build, update, submit, and monitor mobile apps. App store submission a pain? We've got you. Want developer tools in your app? Done. Our application services are built to help you ship faster. And if you get stuck, join our community of developers or say hello and meet our team.



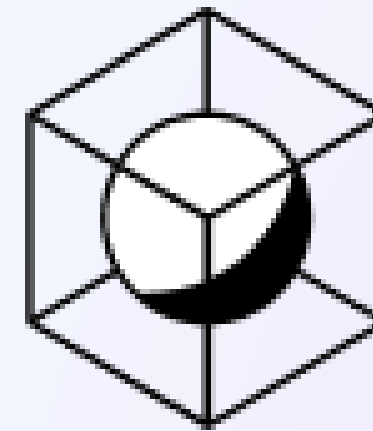
<https://expo.dev/>



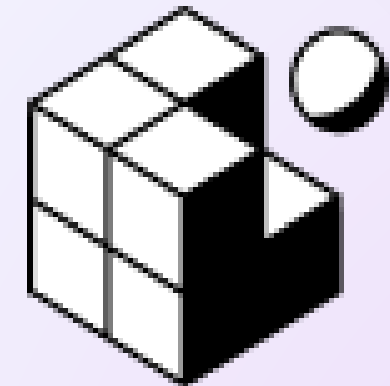
SDK



Go

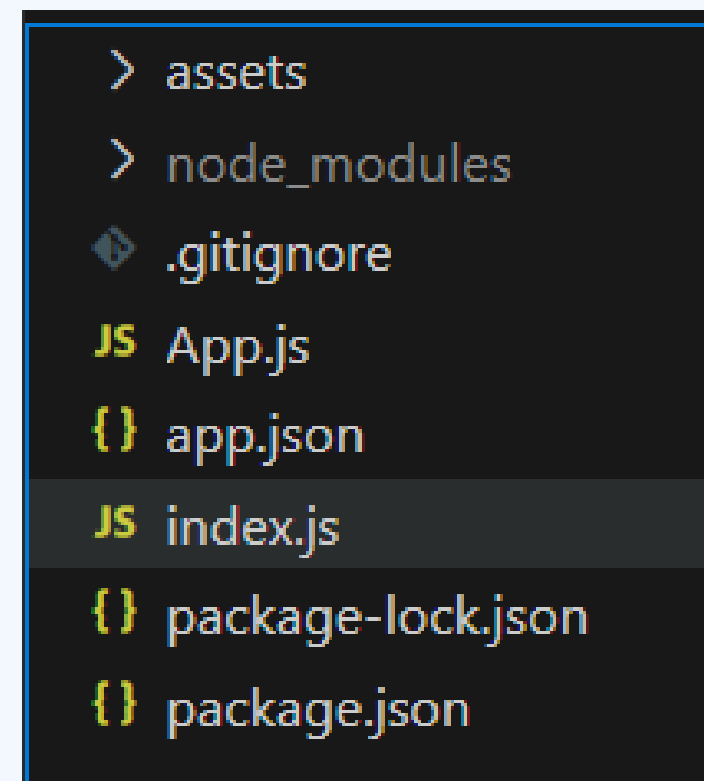
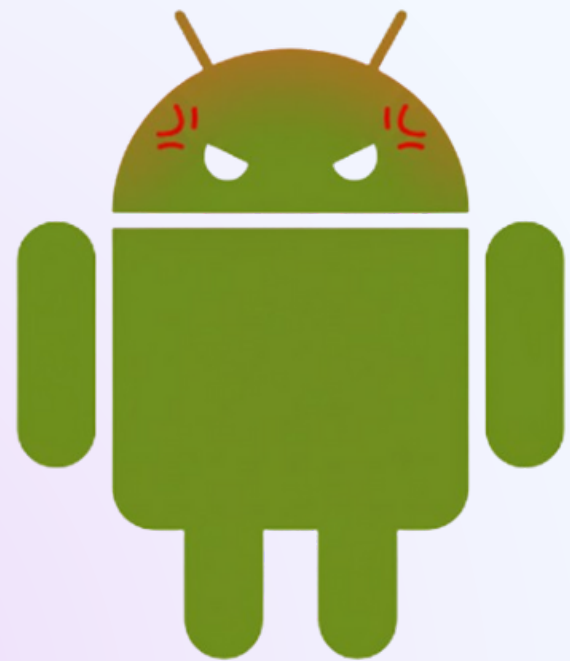


Build

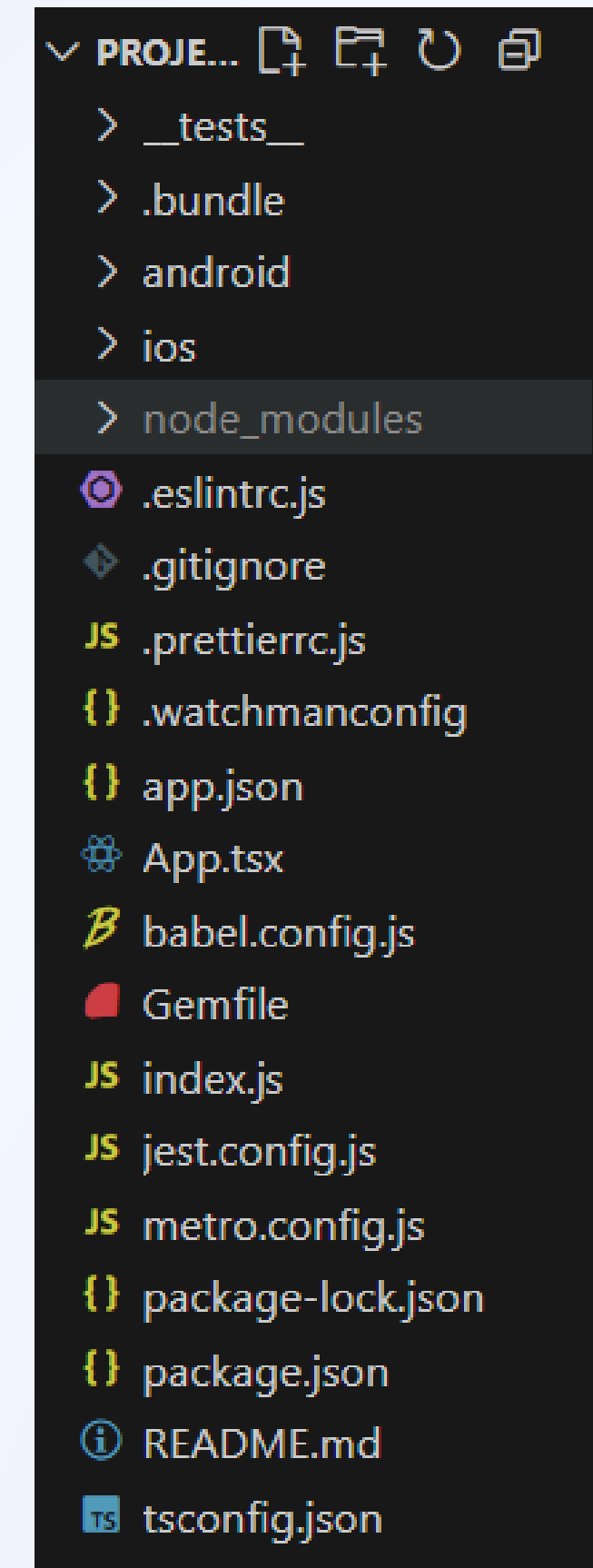


Snacks

# Utilizar Expo ou React Native “Puro” ?



Expo



React Native



# Instalações e primeiro programa

1

Baixar o Node JS

2

Criar um novo projeto Expo

3

Definir ambiente a ser utilizado

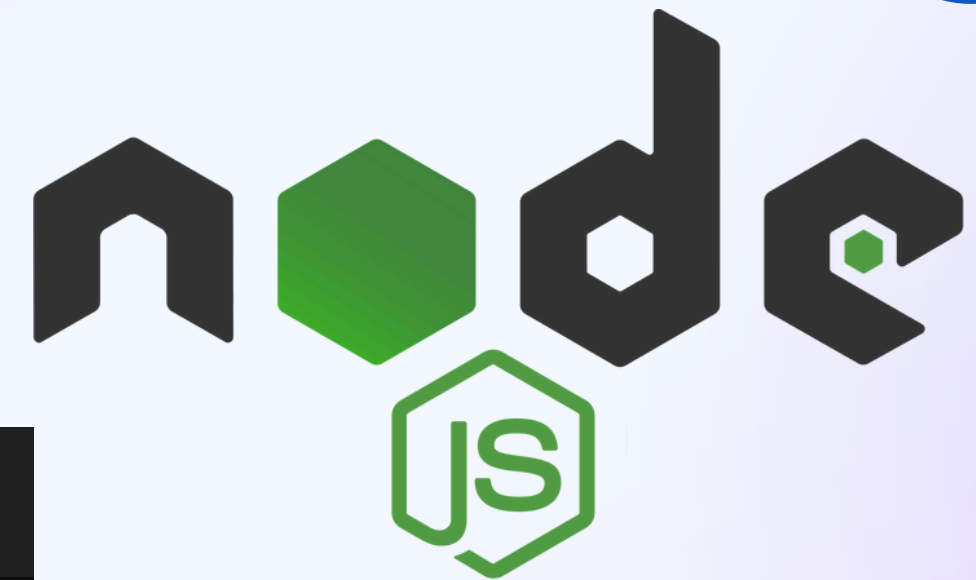
4

Rodar seu primeiro programa

# Primeiro instale o Node JS



<https://nodejs.org/en/download>



```
Selecionar Administrador: Prompt de Comando

C:\Windows\System32>node --version
v22.19.0

C:\Windows\System32>npm --version
10.9.3

C:\Windows\System32>npx --version
10.9.3
```



# Criando o primeiro projeto Expo



Execute o seguinte comando no seu CMD:

```
C:\Users\stefa>npx create-expo-app Primeiro_projeto --template blank|
```

Se estiver utilizando o VSCode, é possível acessar os códigos facilmente com os seguintes comandos

```
C:\Users\stefa>cd Primeiro_projeto
```

```
C:\Users\stefa\Primeiro_projeto>code .
```



Dica: Você pode instalar o VSCode diretamente pelo Microsoft Store

# Definindo ambiente a ser utilizado



Há 4 maneiras de executar seu primeiro código:

## Utilizando Android Studio:

- Virtual Device Manager (VDM)
  - Baixe e configure o Android Studio <https://developer.android.com/studio>
  - Crie um smartfone e uma versão do android desejável
- Cabo conectado em modo deputação
  - Ative o modo de desenvolvedor de seu smartfone
  - Ative o modo depuração USB

## Sem Android Studio:

- Web (Expo Snacks)
  - É necessário instalar umas bibliotecas extras:
  - `npx expo install react-dom react-native web @expo/metro-runtime`
- Expo Go
  - É necessário instalar o Expo Go no celular a partir da PlayStore ou AppStore
  - Se os dispositivos estiverem conectados em redes diferentes é necessário rodar o projeto em modo tunelamento

# Rodando o primeiro programa



- Web (Expo Snacks)

```
C:\Users\stefa\Primeiro_projeto>npx expo start --web
```

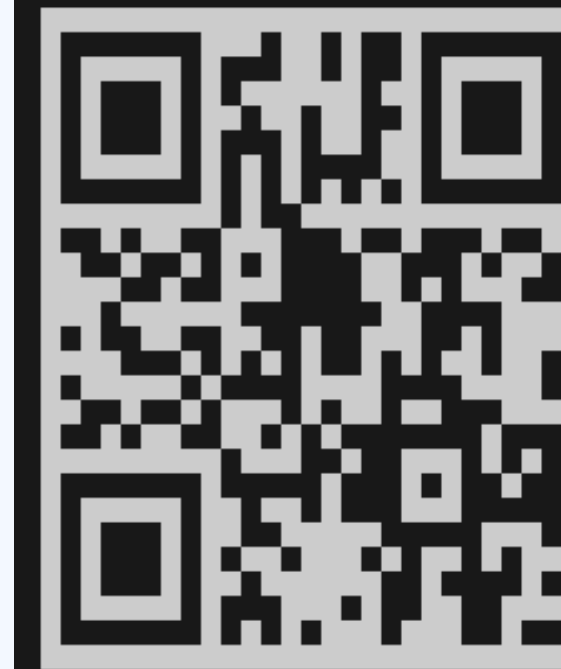
- Acesse <http://localhost:8081/>

- Smartfone (emulado ou depurado)
  - É necessário estar com o dispositivo android/ios conectado seja emulado ou modo depuração
  - Podemos verificar se existem dispositivos identificados setando Path e rodando: adb devices

```
C:\Users\stefa\Primeiro_projeto>npx expo start --android
```

- Expo Go

```
C:\Users\stefa\Primeiro_projeto>npx expo start  
Starting project at C:\Users\stefa\Primeiro_projeto  
Starting Metro Bundler
```



- Após rodar o comando basta Scanear o QRCode através do App Expo Go
- Caso esteja em redes diferentes rode:  
npx expo start --tunnel

Fundamentos

# Aprendendo React Native



Componentes



Hooks



Propriedade



Navegação



Estado



Build com EAS

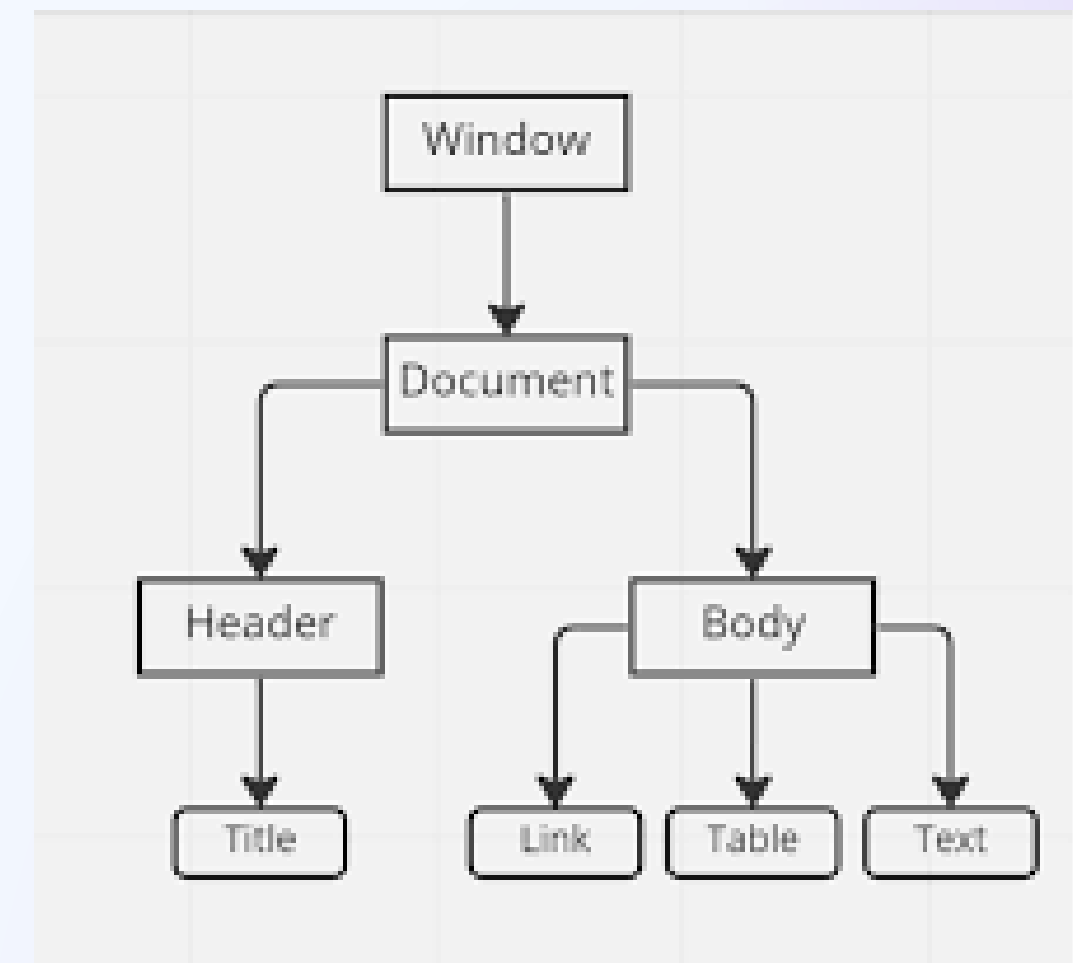
# Componentes

No React Native, os componentes são os blocos fundamentais da interface. Eles podem ser **nativos** (built-in) ou **personalizados**.

Componentes nativos são importados diretamente do React Native, e são fundamentais para criarmos nossos próprios componentes

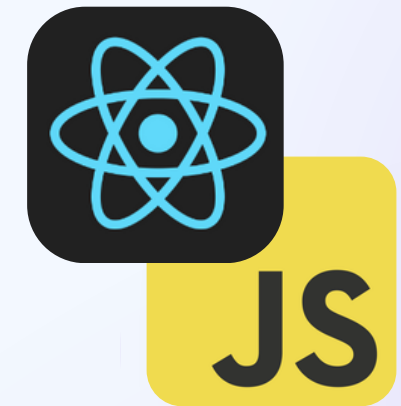
```
import { Text, View, Button, TextInput,  
        Pressable, SafeAreaView, ScrollView,  
        Image, Alert, StyleSheet  
} from 'react-native';
```

Com o desenvolvimento de nosso aplicativo, estaremos formando uma estrutura aninhada em formato de árvore, o que chamamos de **árvore de componentes** da aplicação



# Exemplo criando o primeiro componente

Para criar um componente basta criar uma função em javascript que retorna um trecho de JSX



```
//import React from "react";
import { Text, View, Button, TextInput } from 'react-native';

function MeuComponente(){
  return (
    <View>
      <Text>Digite Algo</Text>
      <TextInput></TextInput>
      <Button title="Meu Botao"></Button>
    </View>
  )
}

export default MeuComponente
```

Podemos salvar o arquivo tanto com a extensão .js quanto com extensão .jsx

```
import MeuComponente from './MeuComponente';

export default function App() {
  return (
    <View style={styles.container}>
      <MeuComponente></MeuComponente>
    </View>
  );
}
```



# Propriedades

Propriedades em React (ou lembrado como props) são como parâmetros que passamos para componentes. Elas permitem reutilizar componentes em diferentes contextos.

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

export default props => {
  return (
    <View>
      <Text>Componente numero {props.num}</Text>
      <TextInput></TextInput>
      <Button title={props.propriedade}></Button>
    </View>
  )
}
```

```
<MeuComponente
  propriedade = "Texto1"
  num={1} />
<MeuComponente
  propriedade = "Texto2"
  num={2} />
<MeuComponente
  propriedade = "Texto3"
  num={3} />
```

Os atributos de props podem assumir diferentes e diversos tipos de dados (inclusive **funções** e **componentes**), onde cabe o programador gerenciar através de convenções ou adoção do **typescript**.



# Estado

O estado representa dados internos de um componente, que podem mudar ao longo do tempo e causam re-renderização da interface.

```
const [numero, setNumero] = useState(0)

function acrescentar() {
  setNumero(numero + 1)
}
function decrescentar() {
  setNumero(numero - 1)
}

return (
  <View style={styles.container}>
    <Text>Contador: {numero}</Text>
    <View style={styles.contador}>
      <Button title='+' onPress={acrescentar} />
      <Button title='-' onPress={decrescentar} />
    </View>
  </View>
)
```

Nesse exemplo queremos atualizar o valor de uma variável e que seu valor seja apresentado na tela. Porém para realizar isso precisamos de uma re-renderização do componente. A Hook **useState** permite que utilizamos uma função que atualiza esse valor juntamente com a renderização do componente.

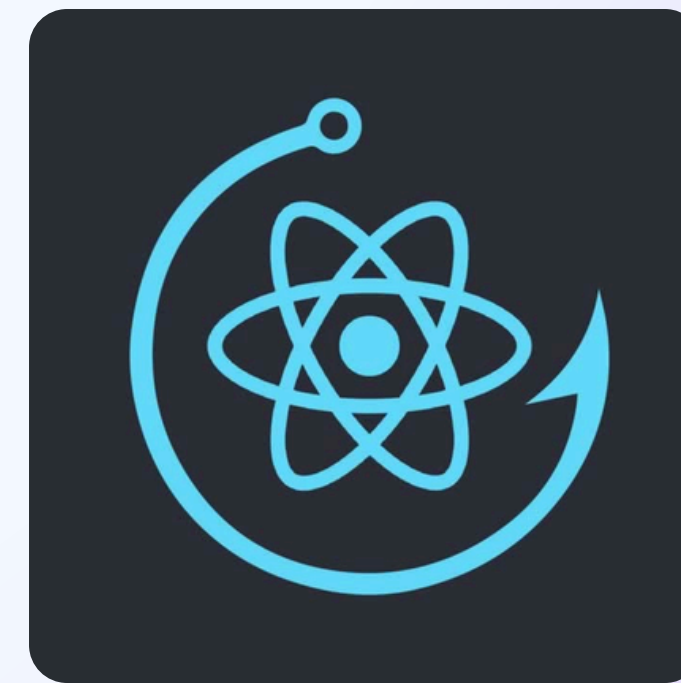
```
import { useState } from 'react';
```

# Hooks

Implementadas a partir do React versão 16.8, Hooks são funções especiais do React que permitem usar estado, ciclo de vida e outros recursos em componentes funcionais.

## Alguns exemplos de Hooks mais utilizadas

- useState
- useColorScheme
- useRouter
- useNavigation
- useEffect
- useContext
- useFonts
- useRef
- entre outros ...



# Navegação

Em uma aplicação completa temos diversas e diferentes telas, utilizamos a navegação para transitar entre essas telas.

Há diferentes bibliotecas para navegação, e também diferentes modos de navegação, como **Stack**, **Tabs** e **Drawer**.



Principais Bibliotecas:

- Expo Router - <https://docs.expo.dev/router/introduction/>
- React Navigation - <https://reactnavigation.org/docs/getting-started>

No Expo Router organizamos nossas rotas através de diretórios e nomes de arquivos, enquanto no React Navigation organizamos definindo as telas como componentes.

```
C:\Users\stefa\Primeiro_projeto>npx expo install expo-router react-native-safe-area-context react-native-screens expo-linking expo-constants expo-status-bar
```

# Navegação - Expo Router

Após a instalação da biblioteca, o primeiro passo para definir as telas utilizando o Expo Router é estruturando de forma correta os arquivos e diretórios

No package.json: `"main": "expo-router/entry",`

```
src > app > JS _layout.js > ...
1  import { Stack, Tabs, Slot } from 'expo-router';
2
3  export default function StackLayout() {
4    return <Stack />
5  }
6
```

Podemos navegar usando **Link** e **useRouter**

```
src
├── app
│   ├── _layout.js
│   ├── index.js
│   ├── Tela1.js
│   ├── Tela2.js
│   ├── Tela3.js
│   └── Tela4.js
├── components
│   └── MeuCompon...
├── .gitignore
├── app.json
├── package-lock.json
└── package.json
```

**\_layout.js:** Definição do tipo de rota  
**index.js:** Tela inicial  
Demais: rotas de navegação

# Algumas Bibliotecas Interessantes

- @expo/vector-icons - <https://icons.expo.fyi/Index>
- React Native Elements - <https://reactnativeelements.com/docs>
- Gluestack - <https://gluestack.io/ui/docs/home/overview/quick-start>
- react-native-calendars - <https://github.com/quidone/react-native-calendars>
- @react-native-community/datetimepicker - <https://docs.expo.dev/versions/latest/sdk/date-time-picker/>
- expo-notifications - <https://docs.expo.dev/versions/latest/sdk/notifications/>

**Banco de dados:** <https://rnfirebase.io/>



# Build da aplicação

Uma vantagem da utilização do Expo é que temos uma ferramenta facilitada para buildar o aplicativo para possível publicação. Utilizamos o EAS - Expo Application Service. É necessário fazer o cadastro na plataforma do expo e seguir os passos da seguinte documentação: <https://docs.expo.dev/build/setup/>

Faça seu cadastro/login em: <https://expo.dev/signup>

```
- npm install -g eas-cli
```

```
- eas login
```

```
- eas build:configure
```

Um arquivo eas.json será criado, onde utilizamos para definir configurações e versionamento. Após isso, para o primeiro build execute e selecione android/ios

```
C:\Users\stefa\Primeiro_projeto>eas build|
```

Nosso arquivo será disponibilizado acessando sua conta Expo, onde podemos configurar os próximos builds



# Do mais é isso...

---

## Contato

viniciusst1.dev@gmail.com

67999030325