

Guia Prático: Integrando Google Maps em um App React Native com Expo

Introdução ao Google Maps

O **Google Maps** é uma das plataformas de mapeamento mais utilizadas do mundo, oferecendo funcionalidades como geolocalização, navegação, visualização de mapas em tempo real, markers (marcadores), rotas e muito mais.

Exemplos de uso em apps:

- Aplicativos de **entrega** (mostrar localização do entregador e cliente)
- Apps de **turismo** (exibir pontos turísticos, rotas etc.)
- Aplicativos de **serviços** (mostrar prestadores próximos ao usuário)
- Apps de **monitoramento de frotas** ou **localização de amigos em tempo real**

Passo a Passo: Criando o Projeto

Pré-requisitos:

- Node.js instalado
- Expo CLI (usaremos o `npx`)
- Conta no Google Cloud Platform com uma API key do Google Maps habilitada

Passo 1: Criar o projeto com Expo

```
npx create-expo-app my-maps-app  
cd my-maps-app
```

Passo 2: Instalar dependências necessárias

`npx expo install react-native-maps`

O react-native-maps é uma biblioteca amplamente utilizada para integração de mapas em React Native. O Expo facilita essa instalação.

Passo 3: Editar o App.js para exibir o mapa

// App.js

```
1  import React from 'react';
2  import { StyleSheet, View, Dimensions } from 'react-native';
3  import MapView, { Marker } from 'react-native-maps';
4
5  export default function App() {
6    return (
7      <View style={styles.container}>
8        <MapView
9          style={styles.map}
10         initialRegion={{
11           latitude: -23.55052,
12           longitude: -46.633308,
13           latitudeDelta: 0.0922,
14           longitudeDelta: 0.0421,
15         }}
16       >
17         <Marker
18           coordinate={{ latitude: -23.55052, longitude: -46.633308 }}
19           title={"São Paulo"}
20           description={"Capital financeira do Brasil"}
21         />
22       </MapView>
23     </View>
24   );
25 }
26
27 const styles = StyleSheet.create({
28   container: {
29     flex: 1,
30   },
31   map: {
32     width: Dimensions.get('window').width,
33     height: Dimensions.get('window').height,
34   },
35 });
36
37
```

Explicações:

- initialRegion: Define o ponto inicial do mapa.
- Marker: Adiciona um marcador no mapa (como um "pin").
- latitudeDelta e longitudeDelta: controlam o zoom.

Passo 4: Configurar a API Key do Google Maps (Android e iOS)

1. Obter chave da API

- Vá até <https://console.cloud.google.com/>
- Crie um projeto, habilite **Maps SDK for Android/iOS**
- Gere uma API Key

2. Configurar no projeto

Crie ou edite o arquivo app.json:

```
1  {
2    "expo": {
3      "name": "my-app-maps",
4      "slug": "my-app-maps",
5      "version": "1.0.0",
6      "orientation": "portrait",
7      "icon": "./assets/icon.png",
8      "userInterfaceStyle": "light",
9      "newArchEnabled": true,
10     "splash": {
11       "image": "./assets/splash-icon.png",
12       "resizeMode": "contain",
13       "backgroundColor": "#ffffff"
14     },
15     "ios": {
16       "supportsTablet": true,
17       "config": {
18         "googleMapsApiKey": "SUA_CHAVE_API_GOOGLE_MAPS"
19       }
20     },
21     "android": {
22       "adaptiveIcon": {
23         "foregroundImage": "./assets/adaptive-icon.png",
24         "backgroundColor": "#ffffff"
25       },
26       "config": {
27         "googleMaps": {
28           "apiKey": "SUA_CHAVE_API_GOOGLE_MAPS"
29         }
30       },
31       "edgeToEdgeEnabled": true
32     },
33     "web": {
34       "favicon": "./assets/favicon.png"
35     }
36   }
37 }
```

Passo 5: Rodar o app

npx expo start