



Deploy de apps React Native na Google Play Store

Material de apoio da aula sobre deploy de aplicativos na Google Play Store

[Assista ao vídeo](#)

License MIT

Deploy na Google Play Store

Partiremos do pressuposto que já possuímos a aplicação backend publicada.

Google Maps Key

OBS.: Caso seu app não utilize o google maps, pule para o **Build com o Expo** ou para o **Build em app criados com a CLI**

☐ No caso do Dev Radar, precisaremos gerar a key do google maps (basta acessar: <https://console.developers.google.com/apis/dashboard?pli=1>)

☐ Com as chaves em mãos, adicione um novo objeto com a chave “android”, com a seguinte config, no app.json do seu app Expo:

```
"config": {  
  "googleMaps": {  
    "apiKey": "SUA_CHAVE_DE_API"  
  }  
}
```

Build com o Expo

☐ Verificar urls externas no seu aplicativo. Usar variáveis de ambiente é o ideal nesse caso.

☐ Configurar ícone e splash screen do app. Em apps expo basta alterar as imagens placeholder.

☐ Mude o valor **backgroundColor** do seu app.json para o a cor do background da sua splash screen. Ex.:
“backgroundColor”: “#7159c1”

☐ Adicione o package do seu app no objeto android:

```
"package": "com.rocketseat.devradar",
```

☐ Adicione o `adaptiveIcon` no objeto `Android` do seu `app.json`, com o `backgroundColor` do seu ícone e o caminho do seu ícone no seu `foregroundImage`.

```
"adaptiveIcon": {  
  "backgroundColor": "#7159c1",  
  "foregroundImage": "./assets/icon.png"  
}
```

☐ Com tudo pronto, rode o comando `expo build:android`. Será necessário criar uma conta no [site do expo](#)

Build em app criados com a CLI

☐ Para configurar a splash screen e o ícone do seu app: <https://www.youtube.com/watch?v=3Gf9yb53bJM>

☐ Para o build de projetos com a CLI a gente precisa acessar a pasta `android/app`, e em seguida rodar o comando abaixo:

```
keytool -genkeypair -v -keystore my-upload-key.keystore -alias my-key-  
alias -keyalg RSA -keysize 2048 -validity 10000
```

☐ Algumas perguntas serão feitas, basta respondê-las corretamente e confirmar tudo no final.

Em seguida, volte para a pasta principal do seu app e abra ela em um editor de texto.

☐ No final do arquivo `gradle.properties`, da pasta `android`, adicione o código abaixo:

```
MYAPP_UPLOAD_STORE_FILE=my-upload-key.keystore  
MYAPP_UPLOAD_KEY_ALIAS=my-key-alias  
MYAPP_UPLOAD_STORE_PASSWORD=123456  
MYAPP_UPLOAD_KEY_PASSWORD=123456
```

OBS.: Lembre-se de mudar `123456` pelo senha que você escolheu no passo anterior, caso você não tenha definido uma senha, deixe onde está `123456` vazio.

☐ No arquivo `build.gradle` da pasta `android/app`, busque pela abertura do `android {}`, e dentro dessas chaves, busque pela `signingConfigs {}`.

☐ Dentro da `signingConfigs {}`, adicione o código abaixo:

```
release {  
  if (project.hasProperty('MYAPP_UPLOAD_STORE_FILE')) {
```

```
        storeFile file(MYAPP_UPLOAD_STORE_FILE)
        storePassword MYAPP_UPLOAD_STORE_PASSWORD
        keyAlias MYAPP_UPLOAD_KEY_ALIAS
        keyPassword MYAPP_UPLOAD_KEY_PASSWORD
    }
}
```

☐ Em seguida, na pasta android, rode o comando `./gradle bundleRelease` (caso você esteja no windows, o comando deve ser `gradlew bundleRelease`).

☐ Quando o processo terminar, você poderá encontrar seu app no seguinte caminho:
`android/app/build/outputs/bundle/release/app.aab`.

Deploy na Google Play Store

☐ É necessário adquirir uma conta de desenvolvedor na Google Play Store

☐ Acesse a sua conta da Google Play Console

☐ Em seguida, crie um novo aplicativo através do botão **CRIAR APLICAÇÃO**.

O restante da publicação se dá em preenchimento de informação, você precisa garantir que, no menu lateral, os 5 itens abaixo estejam corretamente preenchidos e com o **check** verde.

☐ Lançamento de aplicações

☐ Ficha de loja

☐ Classificação de conteúdo

☐ Conteúdo da aplicação

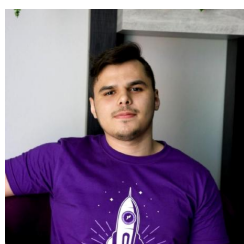
☐ Preço e distribuição

Pra finalizar, acesse **Lançamento de Aplicações**, selecione o rascunho criado e clique em **INICIAR IMPLEMENTAÇÃO PARA A PRODUÇÃO**.

Com isso feito o seu processo de publicação terá sido concluído.

OBS.: Pode levar alguns dias pro seu aplicativo ser aprovado. Futuras atualizações costumam demorar algumas horas apenas.

Autor



Carlos Levir

Licença

This project is licensed under the MIT License - see the [LICENSE](#) page for details.