

SENAI “Roberto Mange”

Gustavo Bruno de Paiva Florentino

**Como Implementar um aplicativo na Play Store**

Como exportar um APK para o próprio telefone

Campinas,

2025

## **Sumário**

1. Objetivo.....	3
2. Pré-requisitos .....	3
3. Preparação do Aplicativo Flutter para Publicação.....	3
3.1 Configuração da versão e do nome .....	3
3.2 Configuração do ícone do aplicativo .....	4
3.3 Gerar a chave de assinatura (Android Keystore).....	4
3.4 Configurar a assinatura no Gradle .....	4
4. Geração do Build Final (APK ou AAB).....	5
5. Publicação na Google Play Store.....	6
5.1 Criar o App no Google Play Console .....	6
5.2 Preencher a Ficha de Loja .....	6
5.3 Política de Privacidade.....	6
5.4 Submeter o AAB .....	6
5.5 Revisão e Publicação.....	7
6. Boas Práticas Recomendadas .....	7
7. Conclusão .....	7

## 1. Objetivo

Este relatório descreve, de forma técnica e operacional, o processo necessário para preparar, assinar, validar e publicar um aplicativo desenvolvido em Flutter na plataforma Google Play Store. O objetivo é fornecer um guia claro que permita a implantação segura e conforme as políticas da Google.

## 2. Pré-requisitos

Antes de iniciar o processo de publicação, é necessário:

- Conta de Desenvolvedor Google Play
- Acessar <https://play.google.com/console>
- Criar uma conta de desenvolvedor.
- Efetuar o pagamento da taxa única (25 USD).
- Código-fonte Flutter finalizado
- O app deve estar funcional, testado e com os erros corrigidos.
- As dependências devem estar atualizadas.
- Certificados e chaves
- Geração de uma chave de assinatura (keystore)
- Configuração do key.properties.
- Informações de Produto
- Nome do app.
- descrição curta.
- descrição longa.
- Screenshots obrigatórios (mínimo: 2 da interface).
- Ícone de alta resolução (512×512).
- Feature Graphic (1024×500).
- Política de Privacidade com link público.

## 3. Preparação do Aplicativo Flutter para Publicação

### 3.1 Configuração da versão e do nome

No arquivo pubspec.yaml:

**version: 1.0.0+1**

A primeira parte (1.0.0) é a versão visível ao usuário.

A segunda (1) é o código interno exigido pela Play Store (incrementado a cada release).

### 3.2 Configuração do ícone do aplicativo

Utilizar o package flutter\_launcher\_icons ou definir manualmente em:

android/app/src/main/res/mipmap-\*/ic\_launcher.png

### 3.3 Gerar a chave de assinatura (Android Keystore)

Executar no terminal:

```
keytool -genkey -v -keystore release-keystore.jks -keyalg RSA -keysize 2048 -  
validity 10000 -alias upload
```

Copiar o arquivo .jks para:

android/app/

Criar o arquivo key.properties em android/:

```
storePassword=*****  
keyPassword=*****  
keyAlias=upload  
storeFile=release-keystore.jks
```

### 3.4 Configurar a assinatura no Gradle

No arquivo android/app/build.gradle, adicionar:

```
signingConfigs {  
    release {  
        keyAlias keystoreProperties['keyAlias']  
        keyPassword keystoreProperties['keyPassword']  
        storeFile file(keystoreProperties['storeFile'])  
        storePassword keystoreProperties['storePassword']  
    }  
}
```

E configurar o build:

```
buildTypes {  
    release {  
        signingConfig signingConfigs.release  
        shrinkResources true  
        minifyEnabled true  
    }  
}
```

#### 4. Geração do Build Final (APK ou AAB)

A Google exige AAB para publicação.

Gerar com o Flutter:

```
flutter build appbundle --release
```

Arquivo final será criado em:

```
build/app/outputs/bundle/release/app-release.aab
```

Caso necessário APK:

`flutter build apk --release`

## 5. Publicação na Google Play Store

### 5.1 Criar o App no Google Play Console

No painel:

- Clicar em “Criar Aplicativo”.
- Selecionar idioma padrão.
- Informar título.
- Declarar se o app é de produção, teste interno ou rascunho.

### 5.2 Preencher a Ficha de Loja

Campos necessários:

- Nome interno e público do app.
- Descrição curta (80 caracteres).
- Descrição detalhada.
- Imagens:
- Mínimo de duas screenshots.
- Ícones e banners conforme especificações.
- Classificação indicativa (questionário obrigatório).
- Categorias (ex.: Ferramentas, Produtividade).

### 5.3 Política de Privacidade

O link deve estar ativo e público (ex.: página em domínio próprio, Notion, GitHub Pages).

### 5.4 Submeter o AAB

Menu Produção > Criar nova versão:

Enviar o arquivo .aab.

Preencher o "Notas da versão".

Salvar e revisar.

## 5.5 Revisão e Publicação

Após o envio:

O Google revisará o app (tempo médio: 24 horas, podendo chegar a 7 dias).

Em caso de pendências, o console indicará corretamente.

Quando aprovado, o aplicativo será disponibilizado na loja.

## 6. Boas Práticas Recomendadas

Incrementar versionCode em todo novo envio.

Evitar permissões desnecessárias.

Adicionar descrição clara de uso das permissões no Manifest quando necessário.

Utilizar ProGuard e shrinkResources para reduzir tamanho.

Manter a política de privacidade atualizada.

Testar em vários dispositivos reais e em emuladores.

## 7. Conclusão

Este relatório consolidou o processo completo para preparar e publicar um aplicativo Flutter na Play Store, desde os pré-requisitos até as ações necessárias de build, assinatura, submissão e conformidade com as políticas do Google. Seguindo essas etapas, o desenvolvedor assegura que o aplicativo será enviado corretamente e estará apto à aprovação.