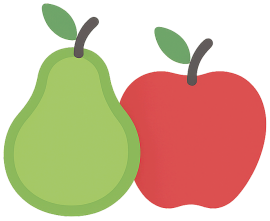
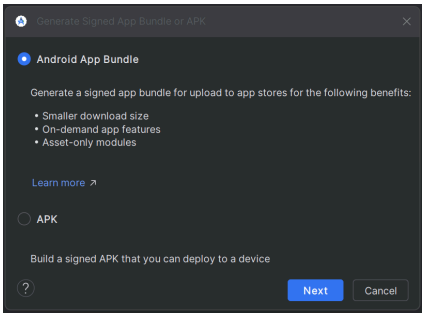
🚀 AE4-ABPRO1 Release AAB (Android App Bundle) Firmado



Guía paso a paso para generar un **Android App Bundle (.aab)** firmado, optimizado y listo para distribuir (Play Store) o probar localmente con `bundletool`.



Generación del empaquetamiento de aplicaciones para Android (AAB)

1) Resumen rápido

- Generar o usar keystore existente.
- Configurar `signingConfig` en `app/build.gradle`.
- Habilitar ofuscación y reducción (R8/ProGuard).
- Generar `bundleRelease` (`.aab`).
- Verificar firma y probar con `bundletool` en dispositivo/emulador.
- Documentar y añadir capturas.

---

2) Generación del keystore

(Ver guía APK) — usar `keytool` si no tienes keystore:

```
keytool -genkeypair -v -keystore release-key.jks -alias my_app_key -keyalg RSA -
keysize 2048 -validity 10000
Espacio para captura (keystore)
```

---

3) Configurar app/build.gradle (Groovy)

La misma configuración de signingConfig y buildTypes que para APK. Gradle firmará el AAB si signingConfig está presente.

```
android {
    compileSdkVersion 34

    signingConfigs {
        release {
            storeFile file(RELEASE_STORE_FILE ?: "release-key.jks")
            storePassword RELEASE_STORE_PASSWORD
            keyAlias RELEASE_KEY_ALIAS
            keyPassword RELEASE_KEY_PASSWORD
        }
    }

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

```
            shrinkResources true
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'),
'proguard-rules.pro'
        }
    }
}
```

---

4) Optimización (R8 / ProGuard)

- minifyEnabled true activa R8 (reemplaza ProGuard).
- shrinkResources true elimina recursos no usados.
- Mantén reglas específicas en proguard-rules.pro.
- Revisión de tamaño: usa Analyze > APK/Bundle en Android Studio o du -h en la carpeta output.

---

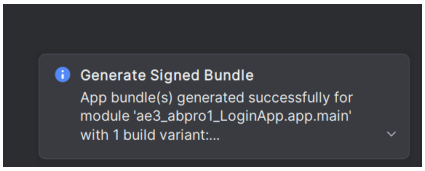
5) Generar el AAB (Android App Bundle)

Desde Android Studio:

- Build > Generate Signed Bundle / APK... → Seleccionar Android App Bundle → Release → Finish.

Desde terminal (Gradle):

```
./gradlew clean bundleRelease
```



El AAB estará en:

```
    ```bash
    app/build/outputs/bundle/release/app-release.aab
    ```
    </td>
</tr>
```

Verificar y firmar (Gradle lo firma si signingConfig está presente)

- Gradle al usar signingConfig firmará el AAB automáticamente. AUN ASÍ puedes verificar el keystore con keytool y los metadatos del bundle con bundletool.

Probar el AAB localmente (bundletool)

- Para instalar una AAB en un dispositivo necesitas bundletool para generar un .apks con APKs específicos para el dispositivo.

---

6) Descargar bundletool

Descarga bundletool.jar desde el repositorio de Google y colócalo en una carpeta (p.ej tools/).

---

7) Generar archivos .apks

Reemplaza rutas/alias/passwords según corresponda

```
java -jar bundletool.jar build-apks
--bundle=app/build/outputs/bundle/release/app-release.aab
--output=app-release.apks
--ks=release-key.jks
--ks-pass=pass:TuPasswordKeystore
--ks-key-alias=my_app_key
--key-pass=pass:TuPasswordKey
```

8) Instalar .apks en el dispositivo

```
java -jar bundletool.jar install-apks --apks=app-release.apks
```

bundletool, detecta el dispositivo conectado y genera los APKs correctos para ese dispositivo.

---

9) Verificar la firma del AAB

AAB es un zip; el cual se puede inspeccionar para comprobar que los archivos se generaron correctamente y que bundletool pudo instalar el APKs. Para verificar el certificado del APK, generado por bundletool, extrae el APK dentro del .apks (zip) o usa bundletool para generar los APKs en una carpeta:

```
java -jar bundletool.jar build-apks --bundle=app-release.aab --output=app-release.apks --mode=universal --ks=release-key.jks --ks-pass=pass:... --ks-key-alias=my_app_key --key-pass=pass:...
unzip app-release.apks -d apks_unzipped
```

10) Encontrar app-release-unsigned.apk o similar

- Verificar con apksigner:

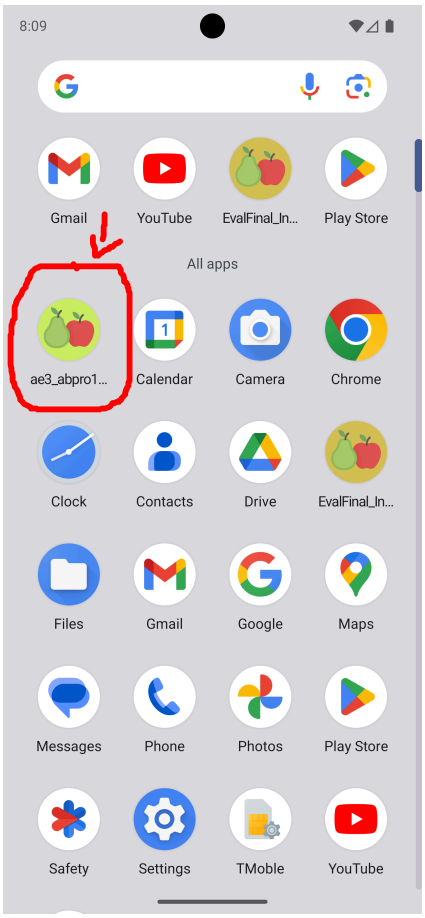
```
apksigner verify --print-certs apks_unzipped/\*.apk
```

---

11) Instalar y probar en dispositivo

- Usa **bundletool install-apks** o instala el APK resultante con adb install si generaste un APK universal.

Verifica funcionalidad: login, navegación, rendimiento y errores.



Luego de instalado se puede observar el ícono en el dispositivo

---

12) Buenas prácticas y optimización

- Usa Android App Bundle para subir a Play Store (reduce tamaño para usuarios).
- Mantén minifyEnabled true y shrinkResources true.
- Conserva mapping.txt para desofuscación de crash reports.
- Usa Play App Signing (recomendado) cuando publiques en Google Play: subirás la AAB y Google lo re-firmará con su key de distribución si eliges Play App Signing.

---

13) Checklist (documentación)

- Keystore generado y guardado seguro
- signingConfig configurado en build.gradle
- minifyEnabled y shrinkResources activados

- bundleRelease generado correctamente (app-release.aab)
- bundletool instaló la app en dispositivo de prueba
- Tests manuales y automáticos ejecutados en release

---

14) Errores comunes y soluciones

- Bundletool no puede firmar: asegúrate de pasar --ks y --ks-pass.
- Bundletool install-apks da error: comprueba adb devices y que el dispositivo esté conectado.
- App falla sólo en release: revisar mapping.txt y proguard-rules.pro para mantener clases necesarias.
