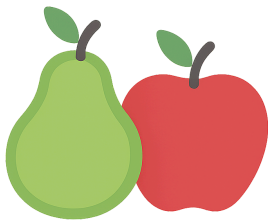
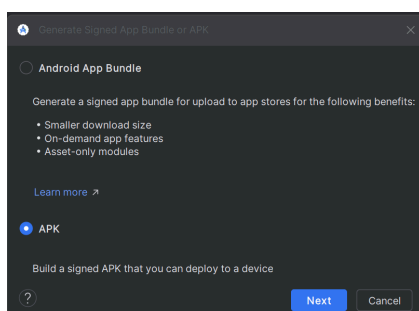


- 🚀 AE4-ABP1 Release APK (Android Package Kit) Firmado
 - 1) Resumen rápido
 - 2) Generación del keystore (firma)
 - 3) Guardar credenciales seguras (opcional y recomendado)
 - 4) Configurar app/build.gradle (Groovy)
 - 5) Configurar reglas ProGuard / R8
 - 6) Consejos para optimización:
 - 7) Generar el APK de release
 - 8) Verificar firma del APK
 - 9) Instalar y probar en dispositivo / emulador
 - 10) Comprobaciones finales y buenas prácticas
 - 11. Checklist (para documentar)
 - 12) Problemas comunes y soluciones
 - 13) Referencias

🚀 AE4-ABP1 Release APK (Android Package Kit) Firmado



Guía paso a paso para generar un APK de release **firmado**, optimizado y listo para distribuir o probar en dispositivos reales.



Generación del kit de empaquetamiento para Android (APK)

1) Resumen rápido

- Generar **keystore** (si no existe).
 - Configurar **signingConfig** en **app/build.gradle**.
 - Habilitar ofuscación y reducción (R8/ProGuard).
 - Generar **assembleRelease**.
 - Verificar firma y probar en dispositivo/emulador.
 - Documentar.
-

2) Generación del keystore (firma)

- **No** compartir el keystore ni las contraseñas en el repositorio. Se debe guardar en un lugar seguro.
- Use **keytool** (JDK) para crear un keystore:

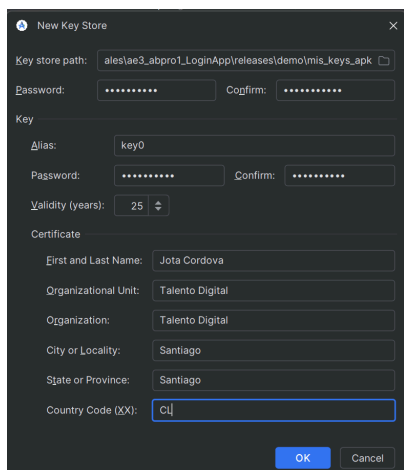
```
keytool -genkeypair \  
-v \  
-keystore release-key.jks \  
-alias aqui_dale_un_nombre \  
-keyalg RSA \  
-keysize 2048 \  
-validity 10000
```

Notas:

release-key.jks: Nombre del archivo del keystore.

aqui_dale_un_nombre: Alias de la clave.

Sigue completando las preguntas: contraseña, nombre, organización, país, etc.



Generación de las claves y almacenamiento seguro

3) Guardar credenciales seguras (opcional y recomendado)

- No pongas contraseñas en build.gradle.
- Añádelas a gradle.properties (local) o variables de entorno.
- Archivo ~/.gradle/gradle.properties (o android/gradle.properties)

```
RELEASE_STORE_FILE=/ruta/a/release-key.jks
RELEASE_STORE_PASSWORD=TuPasswordKeystore
RELEASE_KEY_ALIAS=my_app_key
RELEASE_KEY_PASSWORD=TuPasswordKey
```

- Verificar que gradle.properties local no esté en git (.gitignore).

4) Configurar app/build.gradle (Groovy)

- En app/build.gradle añade signingConfigs y buildTypes:

```
android {
    compileSdkVersion 34

    defaultConfig {
        applicationId "com.tu.organizacion.app"
        minSdkVersion 21
        targetSdkVersion 34
    }
}
```

```

        versionCode 1
        versionName "1.0"
    }

    signingConfigs {
        release {
            // Lee del gradle.properties
            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 // Habilita R8/ProGuard
            shrinkResources true // Elimina recursos no usados
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'),
'proguard-rules.pro'
            // Opcional: deshabilitar debuggable, enable proguard mapping
            debuggable false
        }
        debug {
            // configuración debug
        }
    }
}

```

5) Configurar reglas ProGuard / R8

- Archivo app/proguard-rules.pro
- Mantener los modelos Gson (si los usas)
- `keep class com.tu.organizacion.*_ { _; }`
- Mantener clases usadas por reflection si aplica

```

keepattributes _Annotation_
dontwarn okhttp3
dontwarn retrofit2

```

- Ajusta según bibliotecas utilizadas.
-

6) Consejos para optimización:

- Usar `minifyEnabled true` y `shrinkResources true`.
 - Revisar `mapping.txt` (en `app/build/outputs/mapping/release/`) para depurar stack traces ofuscados.
 - Analizar el APK con Android Studio > Build > Analyze APK... para ver tamaño y recursos.
-

7) Generar el APK de release

- Desde Android Studio:

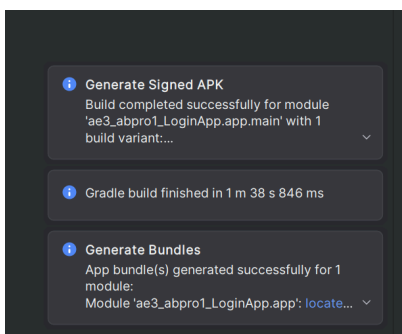
```
***Build > Generate Signed Bundle / APK... → Seleccionar APK → Elegir  
keystore y alias → Release → Finish.***
```

- Desde terminal (Gradle wrapper):

```
***./gradlew clean assembleRelease***
```

- Por defecto, el APK firmado se encontrará en:

```
***app/build/outputs/apk/release/app-release.apk***
```



Generación del APK

8) Verificar firma del APK

- Usar apksigner (Android SDK Build Tools) para verificar:
- Verifica la firma y muestra los certificados:

```
$ANDROID_SDK_ROOT/build-tools/`<version>`/apksigner verify --print-certs  
app/build/outputs/apk/release/app-release.apk
```

Ejemplo:

```
apksigner verify --print-certs app-release.apk
```

- También puedes comprobar el keystore:

```
keytool -list -v -keystore release-key.jks
```

- Opcional: Alinear el APK con zipalign — ya lo hace Gradle, pero si lo haces manual:

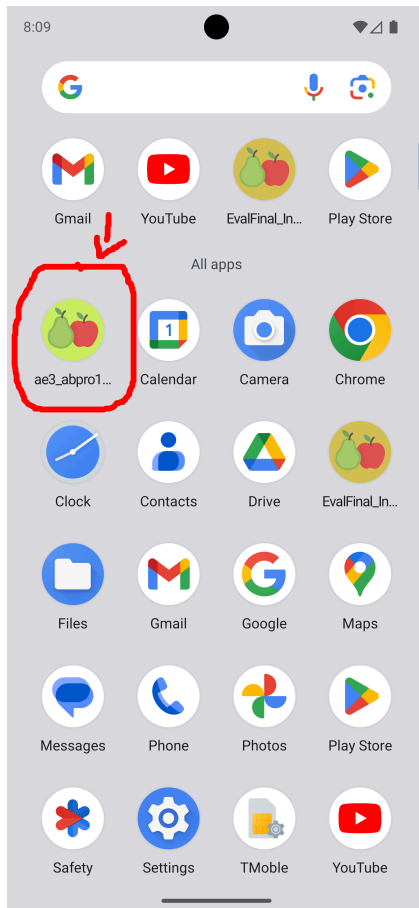
```
zipalign -v -p 4 app-release-unsigned.apk app-release-aligned.apk  
apksigner sign --ks release-key.jks --out app-release-signed.apk app-release-  
aligned.apk
```

9) Instalar y probar en dispositivo / emulador

- Conectar dispositivo y habilitar **Depuración USB**.
- Instalar:

```
adb install -r app/build/outputs/apk/release/app-release.apk
```

Nota: "-r" reemplaza la app existente.



Aplicación instalada en dispositivo emulador (APK)

10) Comprobaciones finales y buenas prácticas

- Revisar mapping.txt para depuración si usa R8.
- No subir release-key.jks ni contraseñas al repositorio.
- Automatiza firma con CI (GitHub Actions/Bitrise) usando secretos.
- Probar en varios dispositivos y APIs.

11. Checklist (para documentar)

- Keystore generado y guardado en lugar seguro

- **build.gradle** configurado con ***signingConfig***
 - **minifyEnabled** y reglas de **ProGuard** definidas
 - **assembleRelease** produce APK firmado
 - **apksigner verify** OK
 - APK instalado y funcional en dispositivo real
 - Keystore creation
 - Configuración build.gradle
 - Resultado Build > APK
 - apksigner verify
 - APK instalado en dispositivo
-

12) Problemas comunes y soluciones

- APK no firmado: revisar signingConfig y que las variables de contraseña existan.
 - Errores R8/ProGuard: revisar reglas y añadir -keep cuando faltan clases por reflexion.
 - APK grande: revisar dependencias, usar shrinkResources true, usar implementation en vez de api en Gradle.
-

13) Referencias

- Android Docs — Sign your app
 - Android Docs — Reduce app size
 - keytool, apksigner, zipalign, bundletool (ver guía AAB)
-