

Guía de Release de App: Inventaris

1. Configuración previa:

1. Configurar `build.gradle.kts`:

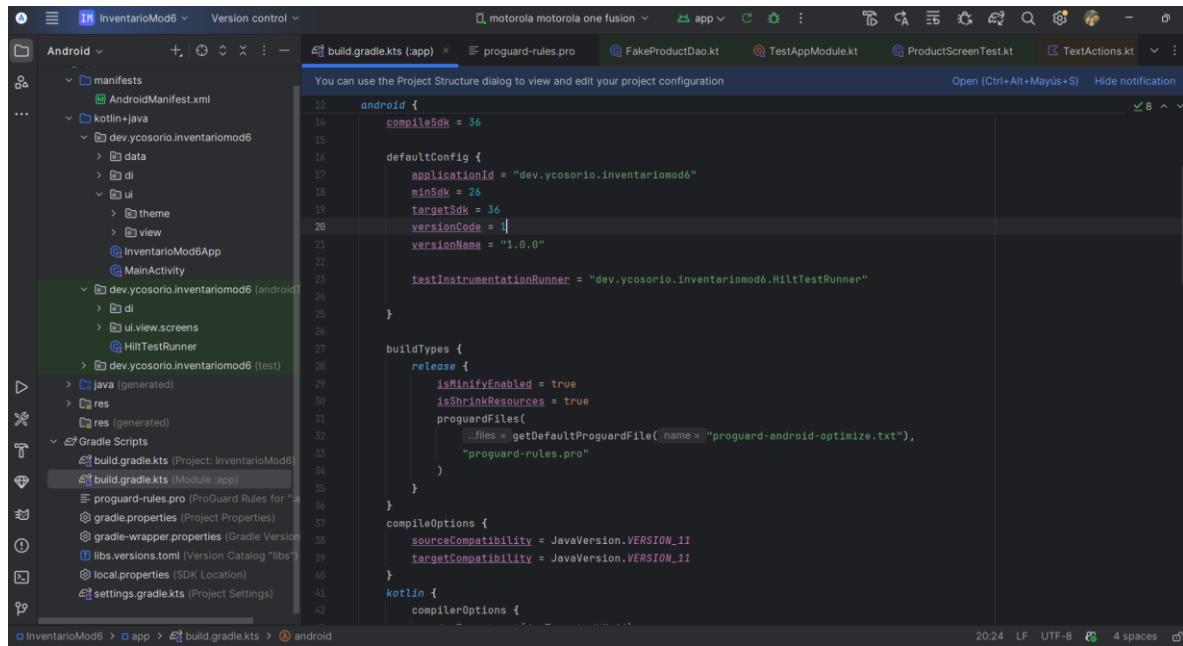
- En `defaultConfig`, verificar:

```
versionCode = X  
versionName = "X.X.X"
```

- En el `buildType de release`:

```
isMinifyEnabled = true  
isShrinkResources = true.
```

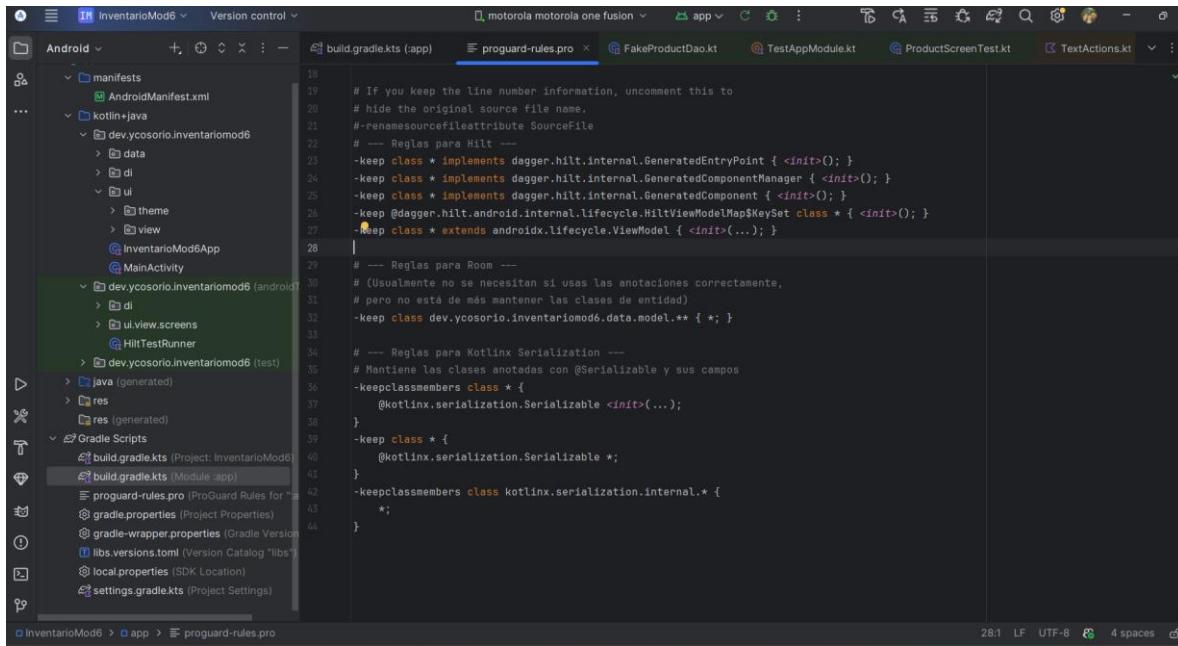
- Habilitar `proguardFiles()`...



Captura 1. Configuración del `build.gradle.kts` (módulo App) abierto en el editor de Android Studio.

2. Configurar `proguard-rules.pro`:

- Este archivo es la *consecuencia* de haber puesto `isMinifyEnabled = true`.
- Se añaden excepciones para que R8 no rompa la app (especialmente importante para Compose y modelos de datos).



```
# If you keep the line number information, uncomment this to
# hide the original source file name.
#-renamesourcefileattribute SourceFile
# --- Reglas para Hilt ---
-keep class * implements dagger.hilt.internal.GeneratedEntryPoint { <init>(); }
-keep class * implements dagger.hilt.internal.GeneratedComponentManager { <init>(); }
-keep class * implements dagger.hilt.internal.GeneratedComponent { <init>(); }
-keep @dagger.hilt.android.lifecycle.HiltViewModelMap$KeySet class * { <init>(); }
-keep class * extends androidx.lifecycle.ViewModel { <init>(...); }

# --- Reglas para Room ---
# (Usualmente no se necesitan si usas las anotaciones correctamente,
# pero no está de más mantener las clases de entidad)
-keep class dev.ycosorio.inventariomod6.data.model.* { *; }

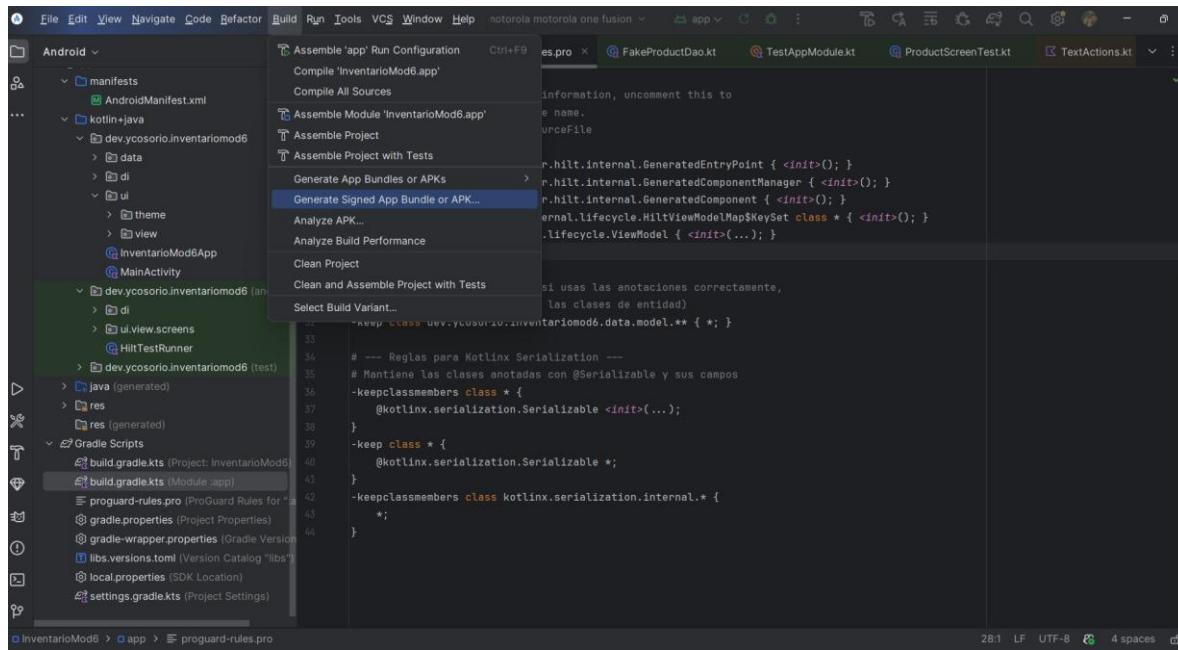
# --- Reglas para Kotlinx Serialization ---
# Mantiene las clases anotadas con @Serializable y sus campos
-keepclassmembers class * {
    @kotlinx.serialization.Serializable <init>(...);
}
-keep class * {
    @kotlinx.serialization.Serializable *;
}
-keepclassmembers class kotlinx.serialization.internal.* {
    *;
}
```

Captura 2. El archivo `proguard-rules.pro` (ubicado en la raíz del módulo `app/`). Se muestra un ejemplo de regla de R8, como '`-keep class...`', usada para prevenir que se eliminen clases necesarias durante la ofuscación.

2. Pasos para Generar la APK de Release Firmada

1. Abrir el Asistente de Generación

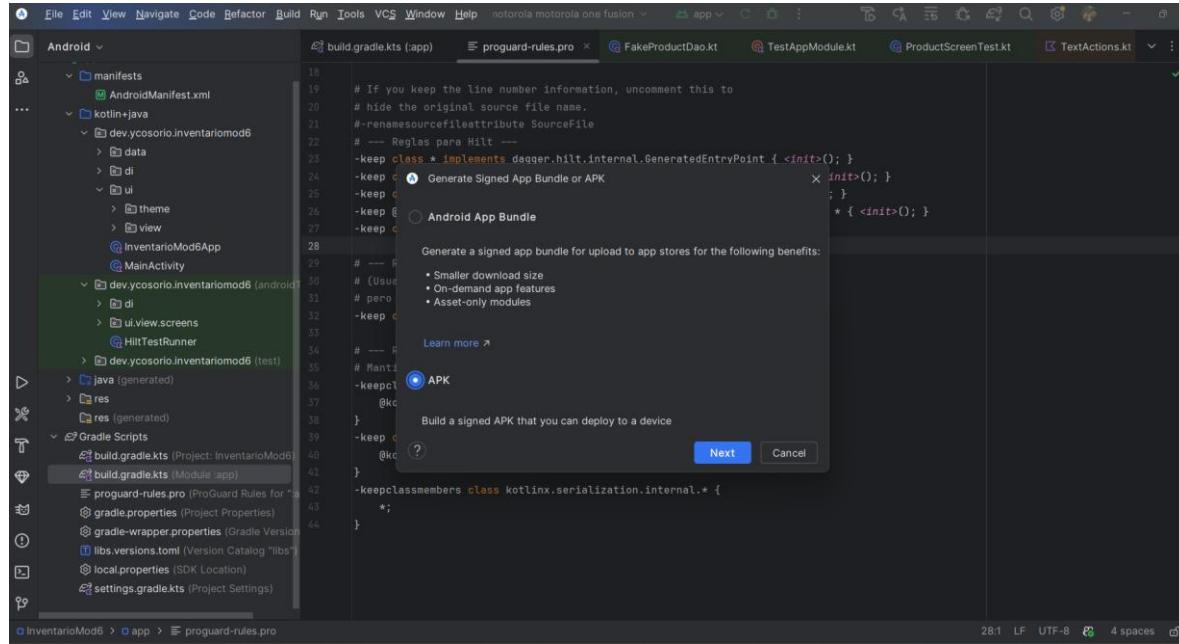
- En el menú superior de Android Studio, ve a **Build > Generate Signed App Bundle or APK...**



Captura 3. Menú de Android Studio con la opción 'Build' desplegada y la sub-opción 'Generate Signed App Bundle or APK...' resaltada.

2. Elegir el formato APK

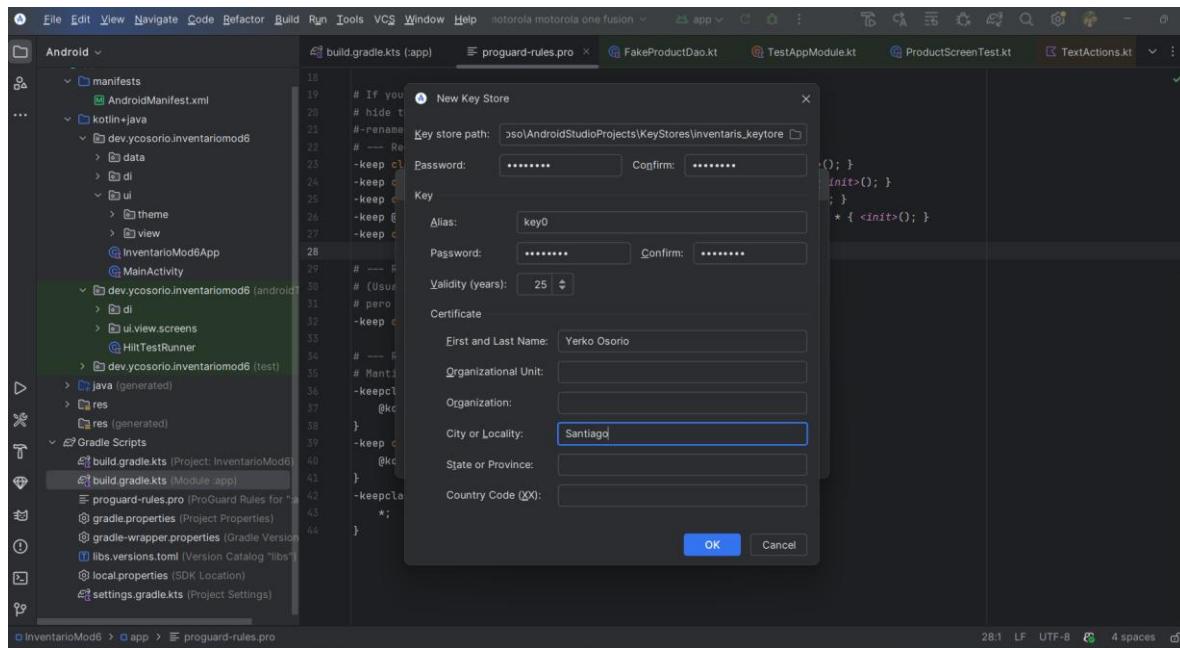
- El asistente te preguntará si quieres generar un "Android App Bundle" (AAB) o un "APK".
- Selecciona la opción **APK** y haz clic en "Next".



Captura 4. Diálogo de selección inicial del asistente, con las opciones 'Android App Bundle' y 'APK'. La opción 'APK' está seleccionada.

3. (Solo si es nueva) Crear la Clave Keystore

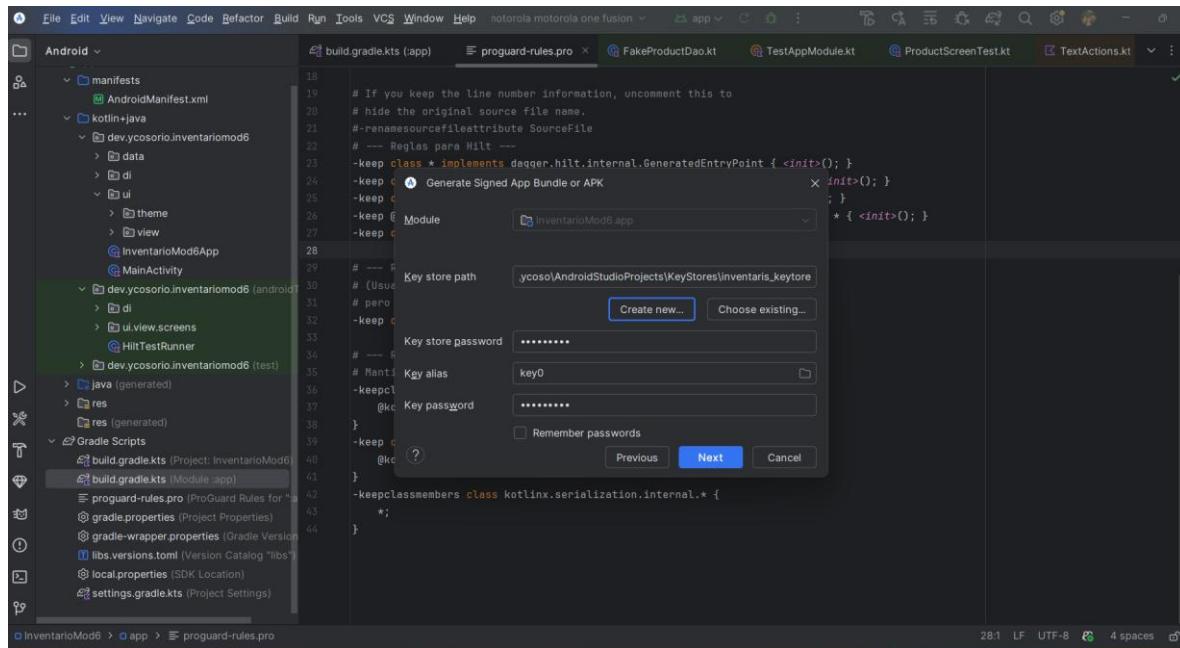
- Este es el paso más importante. Necesitas una clave digital (un archivo .jks o .keystore) para firmar tu app. Esto prueba que tú eres el desarrollador.
- Si NO tienes una clave (primera vez):**
 - Haz clic en el botón "Create new...".
 - Key store path:** Haz clic en el ícono de carpeta y elige dónde guardarás tu archivo de clave. ¡Guarda este archivo y sus contraseñas en un lugar seguro (como un gestor de contraseñas)! Si lo pierdes, no podrás actualizar tu app.
 - Password / Confirm:** La contraseña para el archivo .jks completo.
 - Key - Alias:** Un nombre para identificar esta clave específica (ej. mi-alias-release).
 - Password / Confirm:** La contraseña para ese alias (puede ser la misma que la anterior).
 - Validity (years):** Pon un número alto, como 25 o más.
 - Certificate:** Rellena al menos uno de los campos (nombre, organización, etc.).
 - Haz clic en "OK".



Captura 5. Diálogo 'New Key Store' para crear una clave nueva. Muestra los campos para la ruta del archivo, contraseñas, alias, validez (años) y los campos del certificado.

4. Seleccionar la Clave de Firma (Keystore)

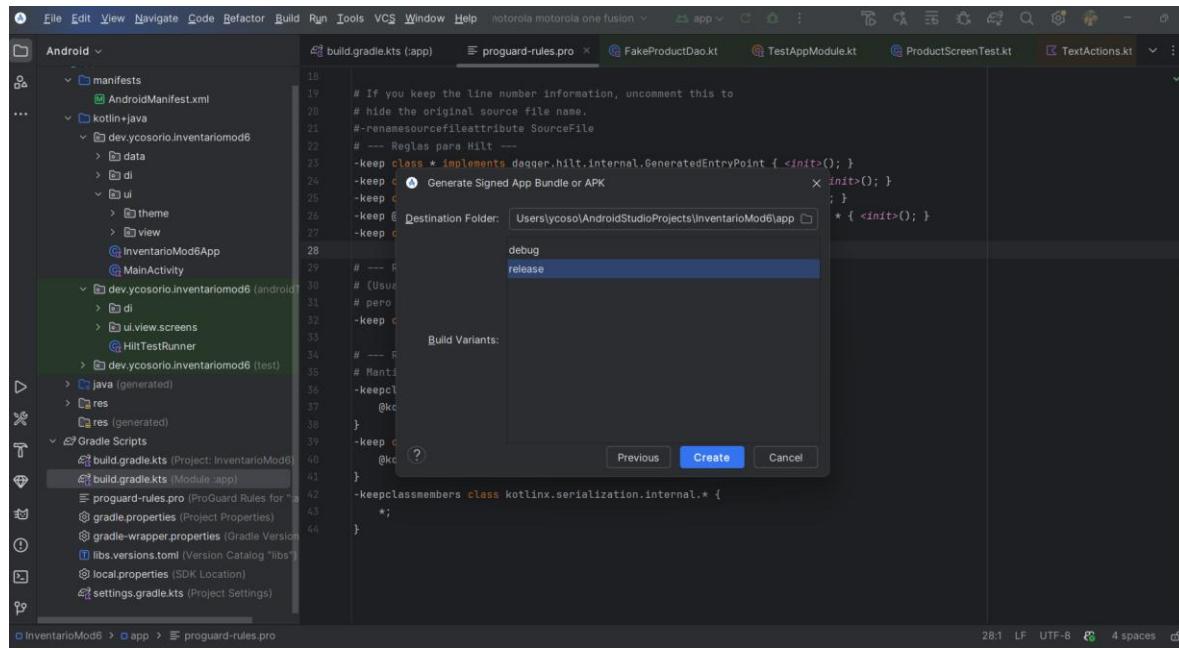
- Si ya tienes una clave:
 - En "Key store path", busca tu archivo .jks.
 - Rellena "Key store password", "Key alias" y "Key password".
 - Volverás a la pantalla del paso 3, pero ahora los campos estarán llenos con la información de la clave que acabas de crear (o la que ya tenías).
 - Asegúrate de que todo esté correcto y haz clic en "Next".



Captura 6. Diálogo 'Generate Signed Bundle or APK' con todos los campos de 'Key store path', 'Key store password', 'Key alias' y 'Key password' completados.

6. Seleccionar la Build Variant y Finalizar

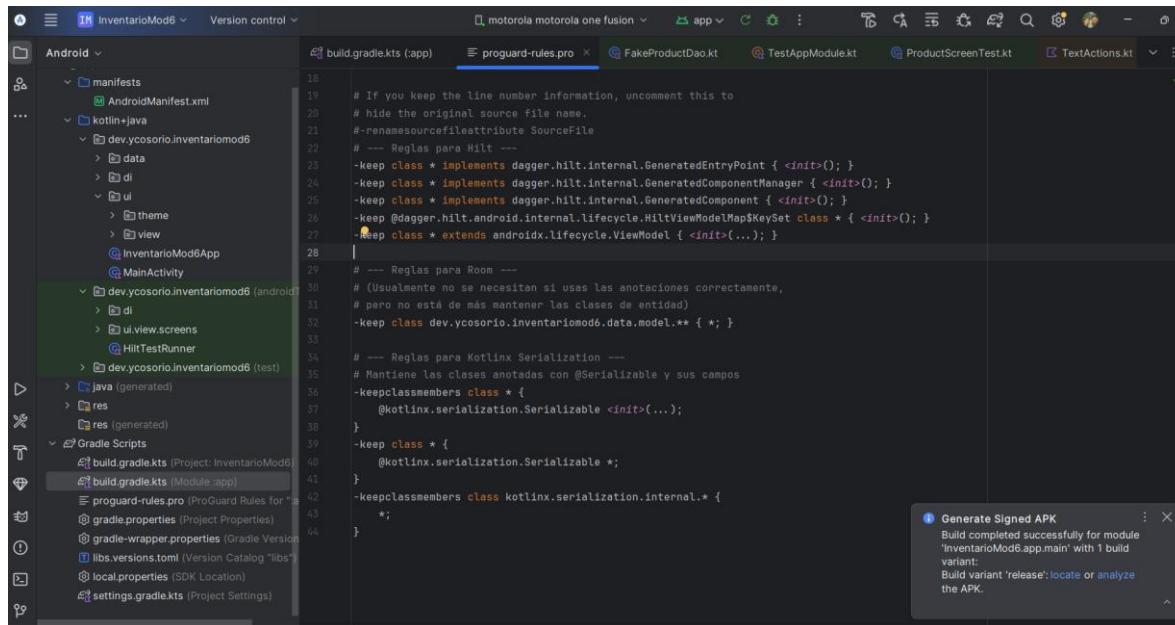
- En la última pantalla, debes decirle a Android Studio *qué* versión compilar.
- **Destination Folder:** Elige dónde se guardará la APK final.
- **Build Variant(s):** ¡Muy importante! Asegúrate de marcar la variante que dice **reléase**.
- Haz clic en "Create".



Captura 7. Diálogo final 'Generate Signed APK(s)' mostrando la carpeta de destino, la lista de 'Build Variants' (con 'release' seleccionado).

7. Localizar la APK

- Android Studio comenzará a compilar la app (la barra de "Gradle Build Running" en la parte inferior comenzará a cargar).
- Cuando termine (puede tardar unos minutos), aparecerá una notificación en la esquina inferior derecha.
- Haz clic en el enlace "**locate**" en esa notificación.
- Se abrirá tu explorador de archivos directamente en la carpeta (usualmente `tu-proyecto/app/release/`) donde encontrarás tu `app-release.apk`, firmada y lista.



Captura 8. Notificación 'popup' de Android Studio en la esquina de la pantalla. Indica 'APK(s) generated successfully' y tiene dos enlaces: 'locate' y 'Analyze'.