

PDM Avanzado Empaquetado e distribución

Sumario

Introdución

Modificando a versión compilada de debug a release

Graficamente

Dende consola

Obtendo a pegada dixital SHA1 dun almacén para utilizar con GoogleMap

Firmando automaticamente una aplicación tanto en modo debug como release

Introdución

Unha vez temos a aplicación rematada chega o momento de, ou ben pasala ó/ós usuarios que van facer uso dela ou ben subila ó Market de Android para a súa comercialización.

Para poder realizar este paso é necesario xerar o APK da aplicación cun certificado válido.

Temos que xerar o arquivo apk da aplicación pero coa aplicación 'firmada' cun keystore que ven ser un almacén de claves.

Cando xeremos o apk primeiro daranos a opción de crear un novo almacén ou utilizar un xa existente.

Nun almacén imos poder 'gardar' moitas aplicacións. Cada almacén terá unha clave.

Así mesmo, cada aplicación terá outra clave asociada o seu certificado.

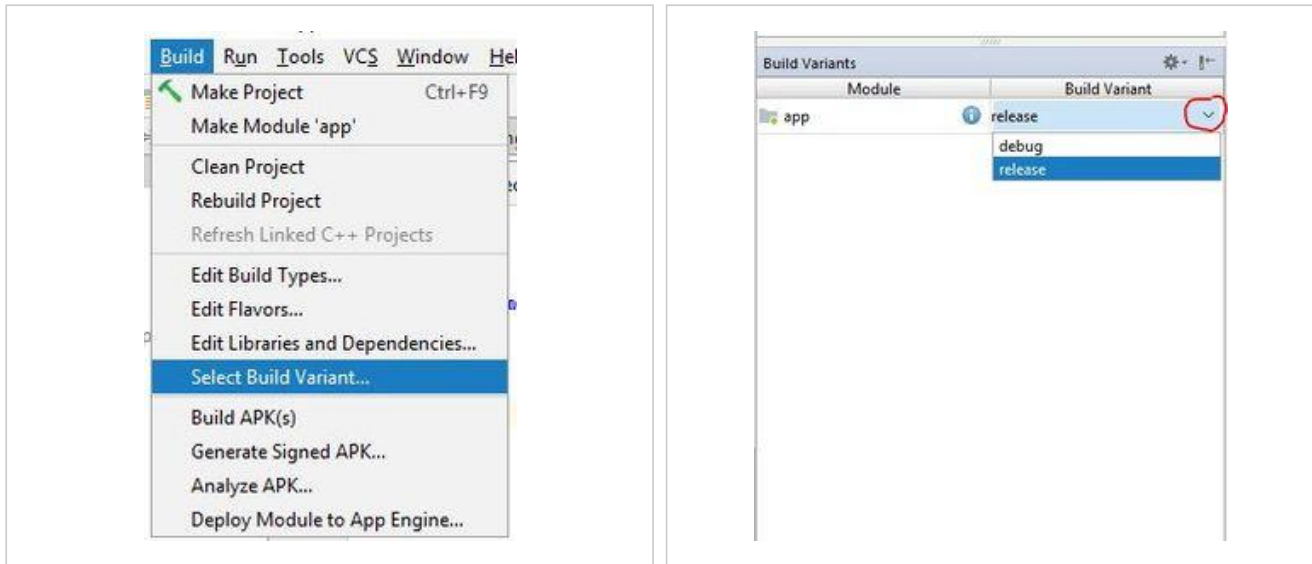
Polo tanto cando xeremos o apk imos ter que gardar dúas claves: unha para o almacén e outra para cada unha das aplicacións gardadas en dito almacén.

A firma é moi importante xa que cada vez que cambiemos de versión teremos que firmala có mesmo certificado para que o S.O. Android o interprete como unha actualización da aplicación.

Modificando a versión compilada de debug a release

- Máis información neste enlace (<https://developer.android.com/studio/run/index.html>).

Modificando a versión release

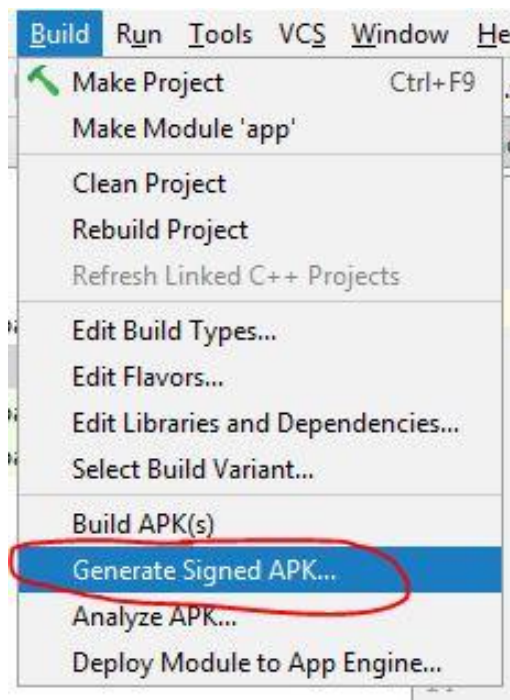


Escollemos a opción de menú **Build => Select Build Variant**. Lembrar que xa comentamos algo sobre como xerar diferentes versións no punto [Compilando diferentes versións](http://wiki.cifprodolfoucha.es/index.php?title=Prog_Android_Compilando_diferentes_versi%C3%B3ns) (http://wiki.cifprodolfoucha.es/index.php?title=Prog_Android_Compilando_diferentes_versi%C3%B3ns) desta Wiki.

Aparecerá unha nova ventá onde escolleremos se queremos xerar o apk na versión debug (para facer probas, a opción por defecto) ou release.

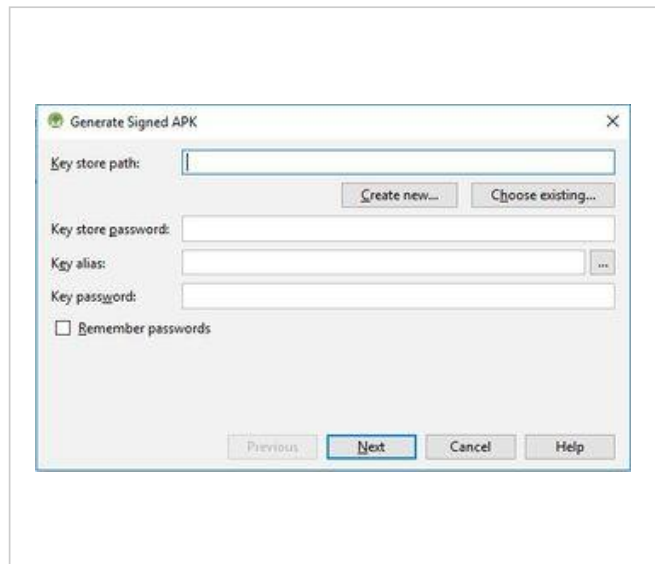
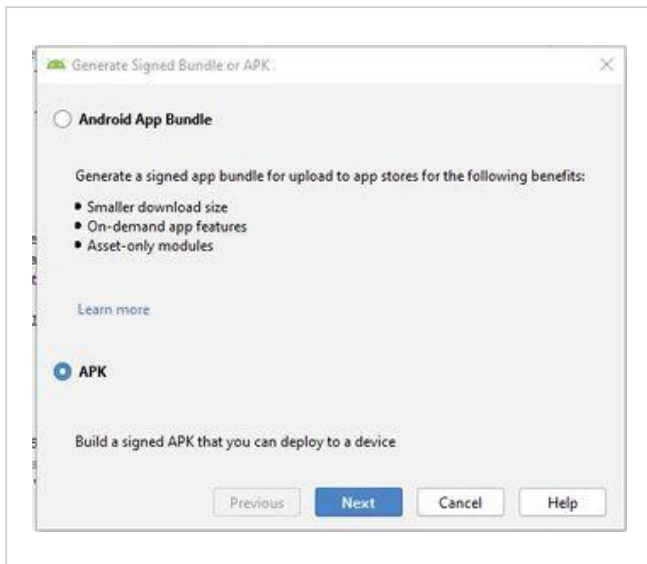
Graficamente

Simplemente imos a opción de menú **Build** e escollemos **Generated Signed Apk**.



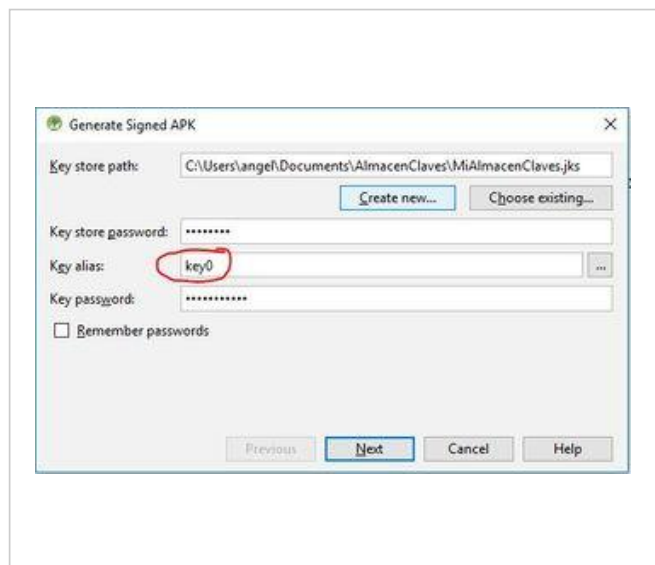
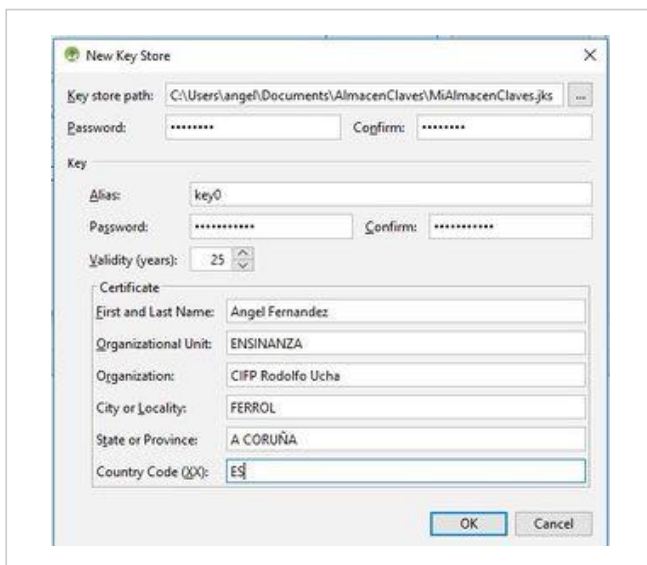
Aparece un asistente:

Asistente de xeración de apk



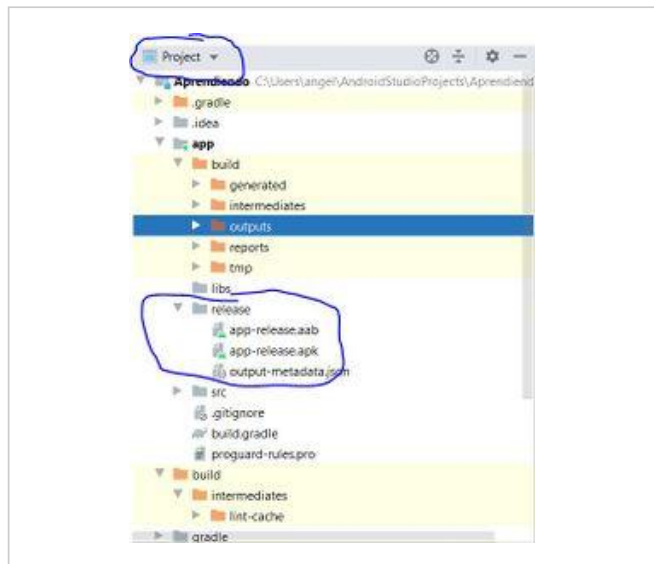
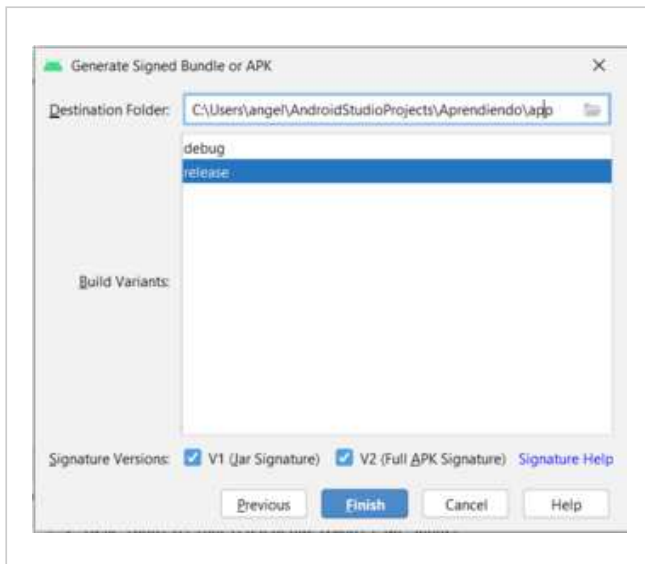
A partires da versión 3.2 podemos elixir xerar a aplicación en un formato aab, que é outra forma de subir a aplicación a Google Play. Este formato 'trocea' a aplicación en trozos para poder xerar o apk personalizado para cada tipo de dispositivo permitindo aforrar espazo. Máis información neste enlace (https://developer.android.com/guide/app-bundle?utm_source=android-studio).

Nesta pantalla podemos indicar se xa temos un keystore (almacén de claves). Se o tivéssemos deberíamos seleccionalo. Inicialmente imos crear un novo premendo **Create New**.



Prememos nos puntos suspensivos e indicamos o **nome e localización do almacén de claves**. Asociamos unha password a dito almacén. Nesta mesma pantalla imos crear unha key (lembrar que nun almacén podemos ter moitas keys diferentes). Temos que indicar: **Alias**: Un alias para o key. Pode ser o mesmo que o nome ou unha abreviación do mesmo. **Password**: Novamente asinámoslle unha contrasinal **pero esta vez á key**. Ten que ter o lo menos 6 caracteres. Esta vai ser a contrasinal da key da aplicación. **Validity (years)**: Aquí indicamos o tempo que vai ser válida a nosa key en anos. Os seguintes campos fan referencia a información persoal e da organización. O campo de Country Code, se pode consultar no listado da ISO 3166-1 (http://es.wikipedia.org/wiki/ISO_3166-1). No noso caso sería ES.

Podemos comprobar como está posto o nome do alias da key do paso anterior xunto có password da key.



Como paso final temos que indicar como firmar a aplicación. A O apk xérase no cartafol **release** partires do Android 7.0 aparece unha nova forma de 'firmar' máis segura (v2). Podemos marcar as dúas opcións. Se temos problemas de instalación en dispositivos cunha versión Android 7.0 ou superior podemos desmarcar a opción v2. Fixarse que estamos xerando a versión **release**.

Agora xa podemos instalar o apk en calquera dispositivo con Android.

- Nota: Indicar que cando creamos o almacén de claves, pode saír o seguinte aviso de erro: 'el almacén de claves jks utiliza un formato propietario. se recomienda migrar a pkcs12'

O almacén funciona correctamente, pero se queremos pasalo ao formato recomendado temos que executar a seguinte orde:

```
1 keytool -importkeystore -srckeystore [MY_KEYSTORE.jks] -destkeystore [MY_FILE.p12] -srcstoretype JKS -deststoretype PKCS12
```

keytool é un executable que se atopa no cartafol /bin/ do Java SDK instalado.

Vos poderá introducir o password do novo almacén (formato p12), o password do almacén anterior (formato jsk) e o password do certificado/s gardados no almacén.

Dende consola

Esta información tamén se pode obter dende unha consola ou terminal. En Windows debemos utilizar unha consola con permisos administrativos (executar como administrador).

Debemos situarnos coa orde **cd** (se non o temos no path) no cartafol onde estea instalado o JDK e dentro deste no cartafol /bin/.

Nese cartafol se atopa o executable keytool.

- LINUX:

```
1 ./keytool -genkey -v -keystore almacen.keystore -alias clavealmacen -keyalg RSA -keysize 2048 -validity 10000
```

- WINDOWS:

```
1 keytool -genkey -v -keystore almacen.keystore -alias clavealmacen -keyalg RSA -keysize 2048 -validity 10000
```

Neste caso estaríamos a crear un almacén e un certificado (key) válido por 10.000 días.

Ó darlle a 'enter' pediranos o password do almacén (mínimo 6 caracteres) e a información que preguntou antes graficamente para o certificado (key):

```
¿Cuáles son su nombre y su apellido?
(Unknown): ANGEL
¿Cuál es el nombre de su unidad de organización?
(Unknown): ENSINANZA
¿Cuál es el nombre de su organización?
(Unknown): PLATEGA
¿Cuál es el nombre de su ciudad o localidad?
(Unknown): FERROL
¿Cuál es el nombre de su estado o provincia?
(Unknown): CORUÑA
¿Cuál es el código de país de dos letras de la unidad?
(Unknown): ES
¿Es correcto CN=ANGEL, OU=ENSINANZA, O=PLATEGA, L=FERROL, ST=CORUÑA, C=ES?
(Unknown): S

Generando par de claves RSA de 2.048 bits para certificado autofirmado (SHA256withRSA) con una validez de 10.000 días
para: CN=ANGEL, OU=ENSINANZA, O=PLATEGA, L=FERROL, ST=CORUÑA, C=ES
```

Unha vez feito xa temos o almacén creado.

Obtendo a pegada dixital SHA1 dun almacén para utilizar con GoogleMap

Relacionado coa Unidade vista anteriormente de Google Map (http://wiki.cifprodolfoucha.es/index.php?title=PDM_Avanzado_GoogleMap_s#Obtenci.C3.B3n_da_pegada_dixital_SHA1_Produci.C3.B3n) temos a opción de obter a clave SHA1 dun almacén de datos, necesario para poder ver o mapa de Google Map nunha aplicación que queiramos dar ós usuarios.

Para obter o SHA1 debemos escribir dende consola:

```
1 keytool -list -keystore almacen.keystore
```

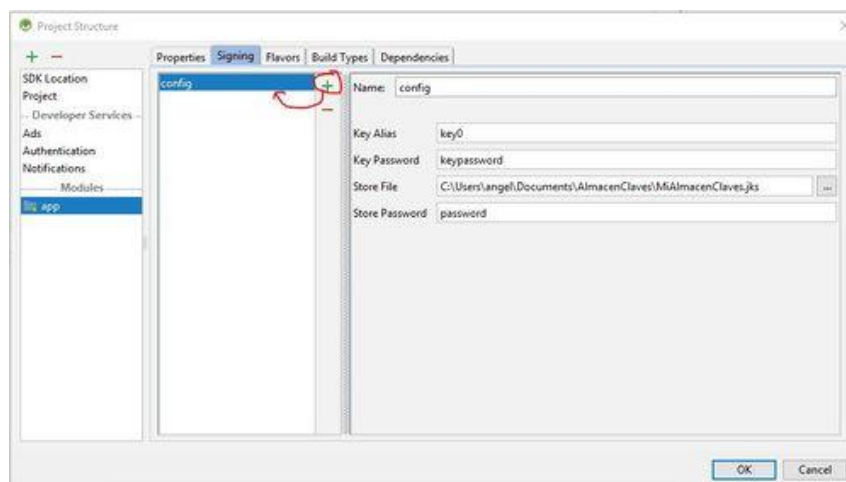
Sendo almacen.keystore o almacén onde se atopa a key que imos utilizar para firmar a nosa aplicación.

Firmando automaticamente una aplicación tanto en modo debug como release

- Podemos facer que de forma automática o Android Studio 'firme' a aplicación antes de instala no emulador / dispositivo.

Para facelo, imos a ventá de **Project Structure**:

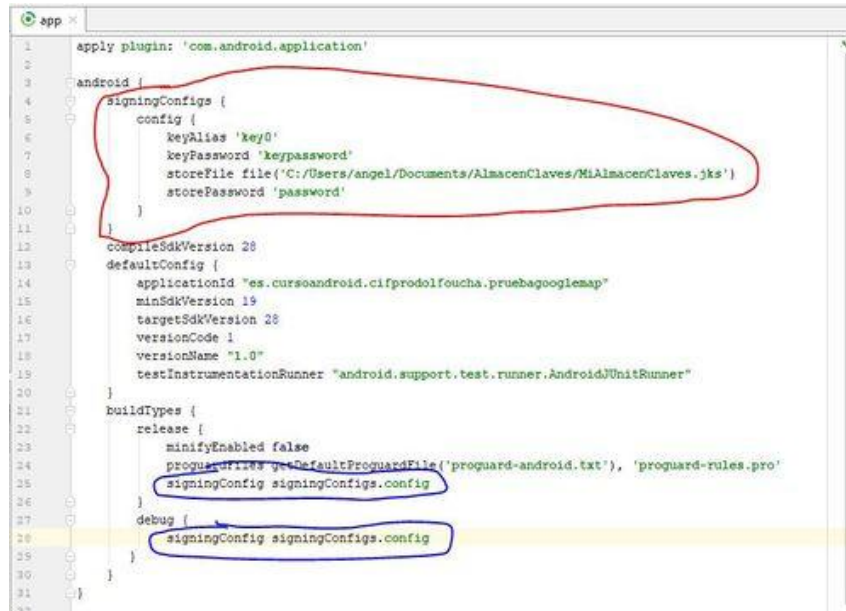
Imos a lapela de Signing e prememos o símbolo 'máis'. Indicamos os datos do almacén de claves creado previamente.



- Agora debemos editar o arquivo **build.gradle** a nivel de módulo.

Nese arquivo estará a configuración creada no paso anterior.

Temos que modificar as entradas 'release' e 'debug' para que fagan uso do almacén de claves configurado anteriormente.



[Enlace a la página principal de la UD8 \(https://wiki.cifprodolfoucha.es/index.php?title=Programaci%C3%B3n_de_dispositivos_m%C3%B3viles#UNIDADE_8:_Empaquetado_e_distribuci.C3.B3n._Xeraci.C3.B3n_de_arquivos_APK.\)](https://wiki.cifprodolfoucha.es/index.php?title=Programaci%C3%B3n_de_dispositivos_m%C3%B3viles#UNIDADE_8:_Empaquetado_e_distribuci.C3.B3n._Xeraci.C3.B3n_de_arquivos_APK.)

[Enlace a la página principal del curso \(https://wiki.cifprodolfoucha.es/index.php?title=Programación_de_dispositivos_móviles\)](https://wiki.cifprodolfoucha.es/index.php?title=Programación_de_dispositivos_móviles)

-- Ángel D. Fernández González e Carlos Carrión Álvarez -- (2015).

Obtenido de «https://wiki.cifprodolfoucha.es/index.php?title=PDM_Avanzado_Empaquetado_e_distribución&oldid=12095»

Esta página se editó por última vez el 14 ene 2021 a las 18:43.