INDICE

1.1. Introducción

Una vez tenemos la aplicación rematada llega el momento de, o bien pasarla a lo/s usuario/s que van a hacer uso de ella o bien subirla al Market de Android para su comercialización.

Para poder realizar este paso es necesario generar el APK de la aplicación con un certificado válido.

Tenemos que generar el archivo de la apk de la aplicación pero con la aplicación 'firmada' con un keystore que viene a ser un almacén de claves.

Cuando generemos el apk primero nos dará la opción de crear un nuevo almacén o utilizar uno ya existente.

En un almacén vamos a poder 'guardar' muchas aplicaciones. Cada almacén tendrá una clave.

Así mesmo, cada aplicación tendrá otra clave asociada a su certificado.

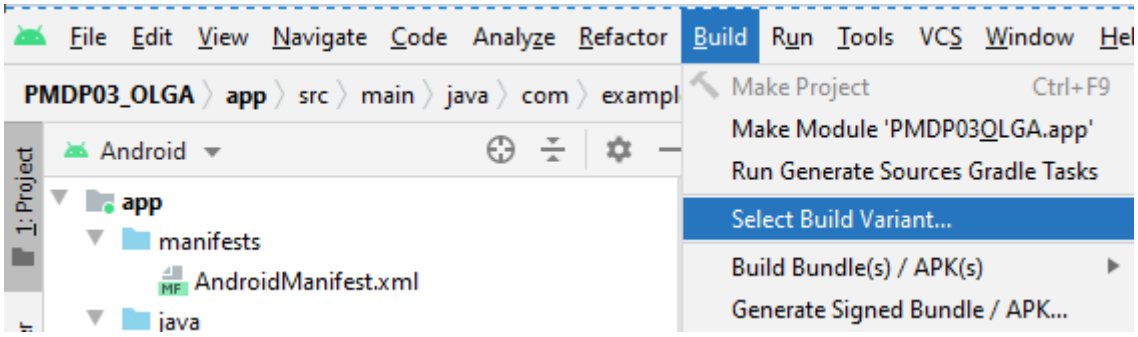
Por lo tanto cuando generemos el apk vamos a tener que guardar dos claves: una para el almacén y otra para cada una de las aplicaciones guardadas en dicho almacén.

La firma es muy importante ya que cada vez que cambiemos de versión tendremos que firmarla con el mismo certificado para que el S.O. Android lo interprete como una actualización de la aplicación.

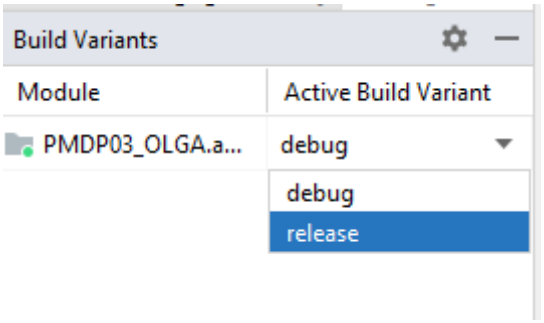
1.2. Modificación a la versión compilada de debug a release

- ✓ Más información en este enlace:

Modificando la versión release



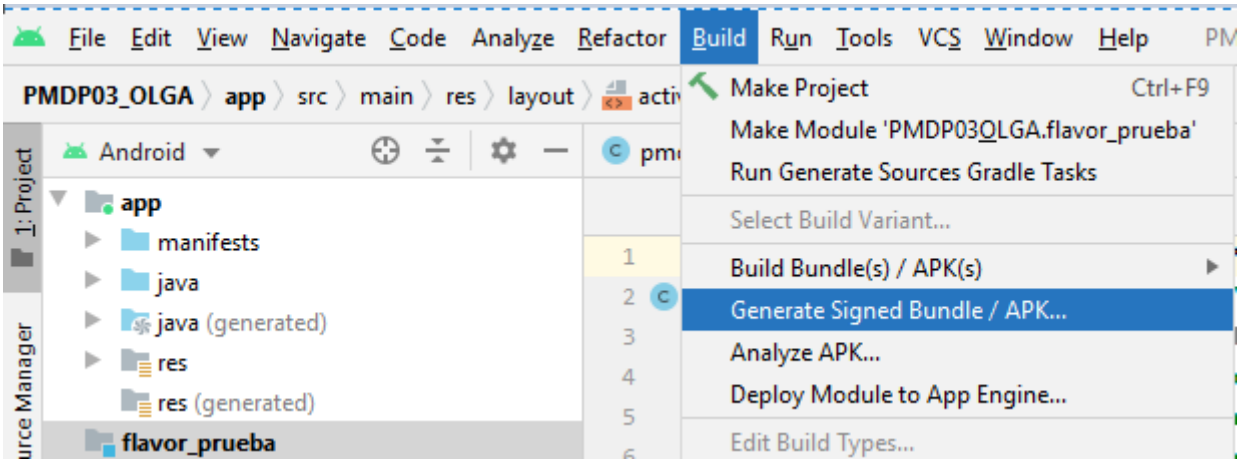
Escogemos la opción del menú **Build** => **Select Build Variant**.



Aparecerá una nueva ventana ddonde escogeremos si queremos generar el apk en la versión *debug* (para hacer pruebas, la opción por defecto) o *release*.

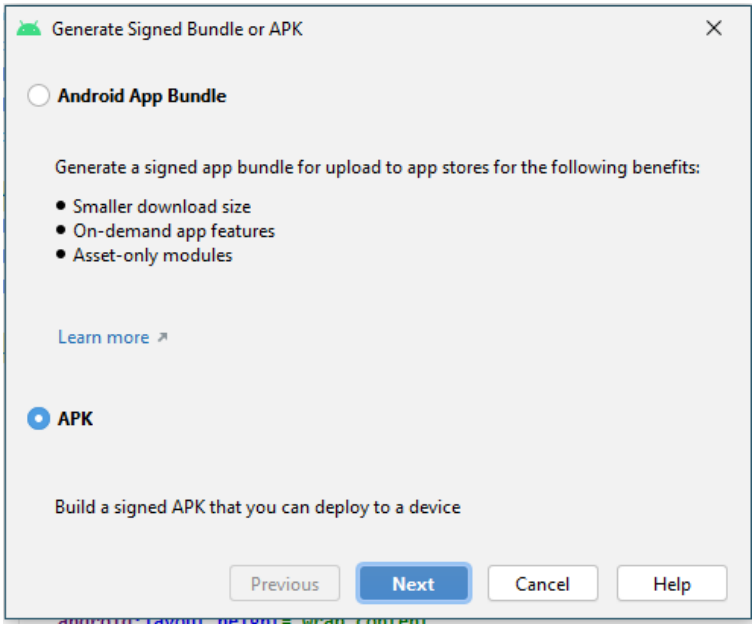
1.3. Gráficamente

Simplemente vamos a la opción de menú **Build** y escogemos **Generated Signed Apk**.

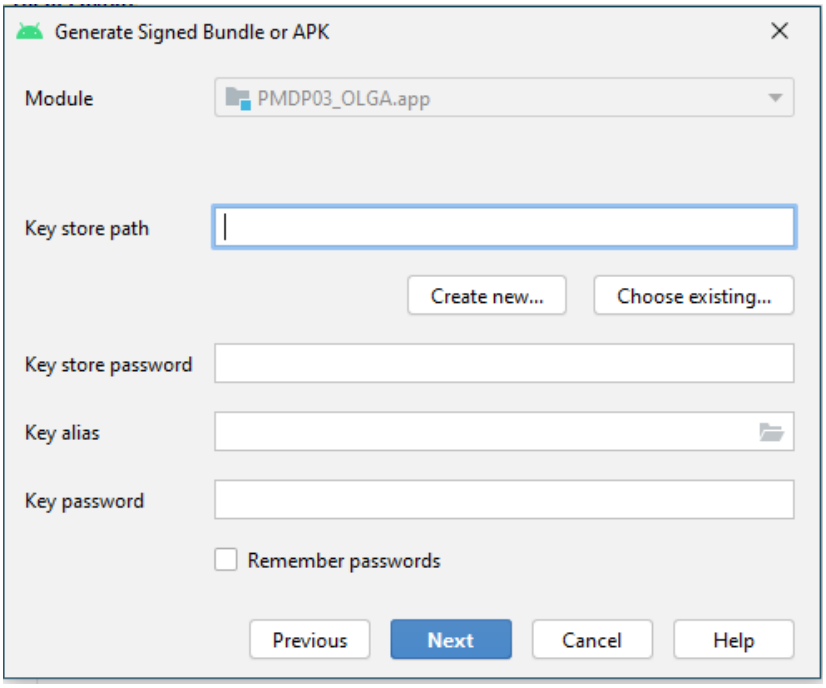


Aparece un asistente:

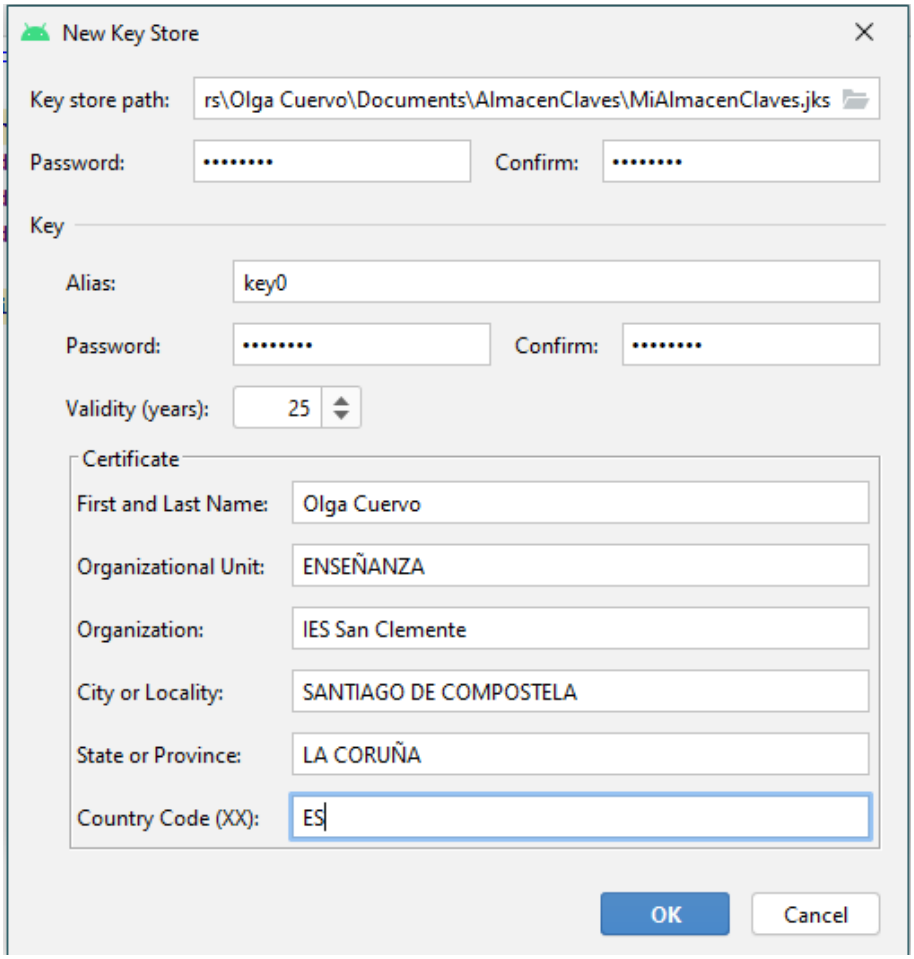
Asistente de generación de apk



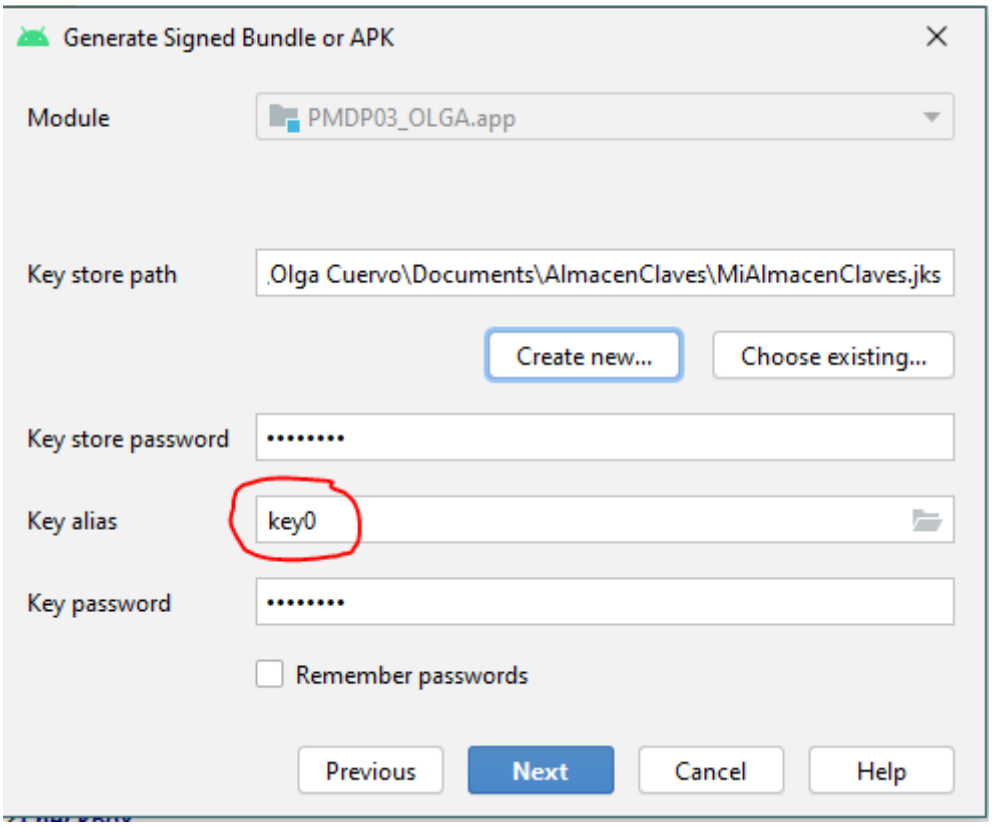
A partir de la versión 3.2 podemos elegir generar la aplicación en un formato aab, que es otra forma de subir la aplicación a Google Play. Este formato 'trocea' la aplicación en trozos para poder generar el apk personalizado para cada tipo de dispositivo permitiendo ahorrar espacio. Más información en este enlace.



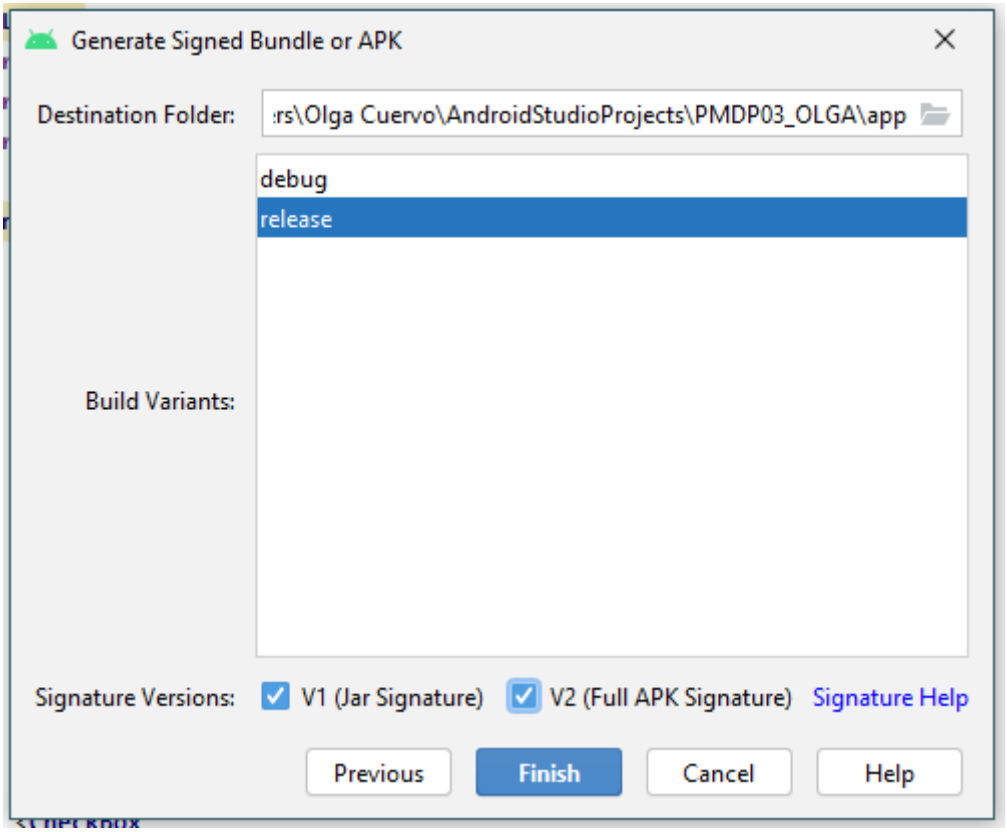
En esta pantalla podemos indicar si ya tenemos un keystore (almacén de claves). Si lo tuviésemos deberíamos seleccionarlo. Inicialmente vamos a crear uno nuevo haciendo clic en **Create New**.



Hacemos clic en los puntos suspensivos e indicamos el **nombre y localización del almacén de claves**. Asociamos una password a dicho almacén. En esta misma pantalla vamos a crear una key (recordar que en un almacén podemos tener muchas keys diferentes). Tenemos que indicar: **Alias:** Un alias para la key. Puede ser el mismo que el nombre o una abreviación del mismo. **Password:** Nuevamente asignamos una contraseña **pero esta vez a la key**. Tiene que tener por lo menos 6 caracteres. Esta va a ser una contraseña de la aplicación. **Validity (years)**: Aquí indicamos el tiempo que va a ser válida nuestra osa key en años. Los siguientes campos hacen referencia a la información personal y de la organización. El campo de Country Code, se puede consultar en el listado de la ISO 3166-1. En nuestro caso sería ES.

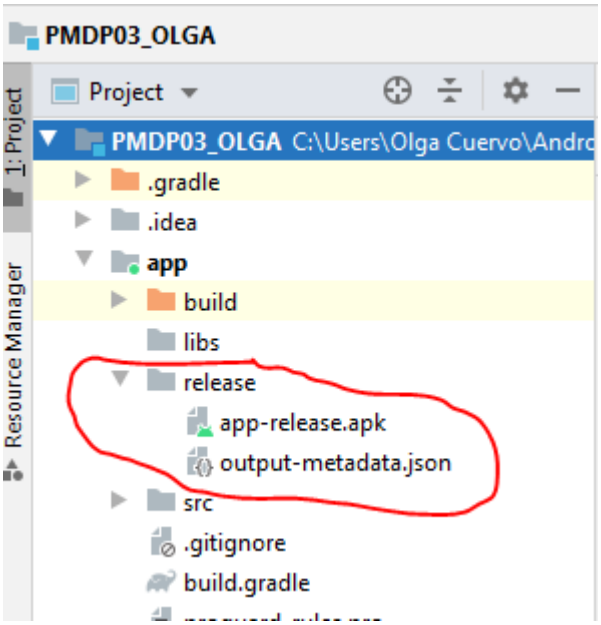


Podemos comprobar cómo está puesto el nombre del alias de la key del paso anterior junto con el password **de la key**.



Como paso final tenemos que indicar como firmar la aplicación. A partir del Android 7.0 aparece una nueva forma de 'firmar' más segura (v2). Podemos marcar las dos opciones. Si tenemos problemas de instalación en dispositivos con una versión

Android 7.0 o superior podemos desmarcar a opción v2. Fijarse que estamos generando una versión **release**.



El apk se genera en la carpeta **release**

Ahora ya podemos instalar el apk en cualquier dispositivo con Android.

1.4. Desde consola

Esta información también se puede obtener desde una consola o terminal. En Windows debemos utilizar una consola con permisos administrativos (ejecutar como administrador).

Debemos situarnos con la orde **cd** (si no lo tenemos en el path) en la carpeta donde este instalado el JDK y dentro de este en la carpeta /bin/.

En esta carpeta se encuentra el ejecutable keytool.

- ✓ **LINUX:**

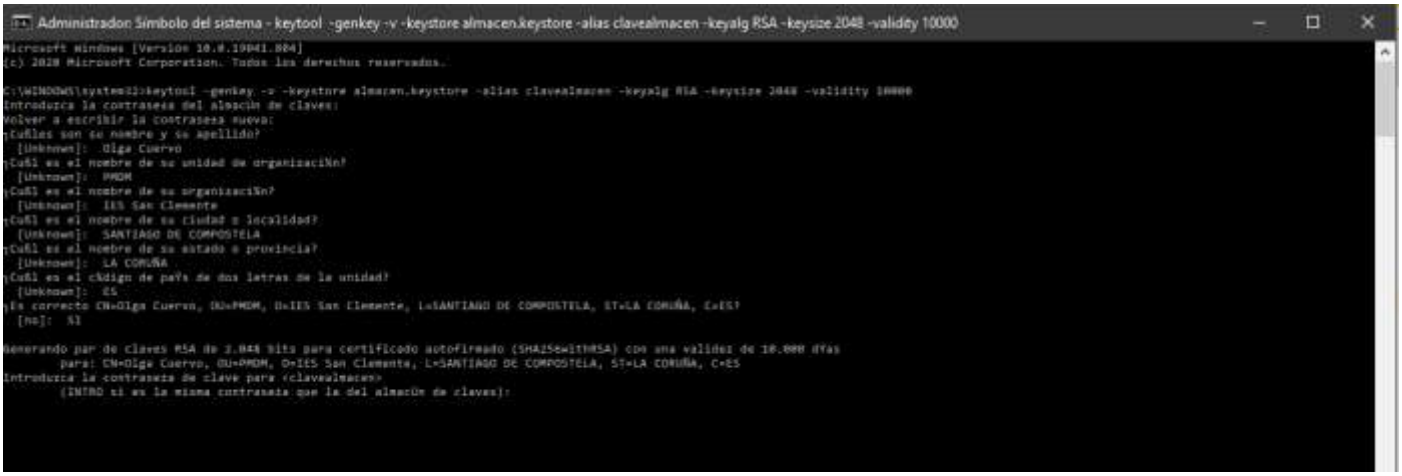
```
/keytool -genkey -v -keystore almacen.keystore -alias clavealmacen -keyalg RSA -keysize
2048 -validity 10000
```

- ✓ **WINDOWS:**

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

En este caso estariamos creando un almacén y un certificado (key) válido por 10.000 días.

Al pulsar *'enter'* nos pedirá el password del almacén (mínimo 6 caracteres) y la información que pregunto antes graficamente para el certificado (key):



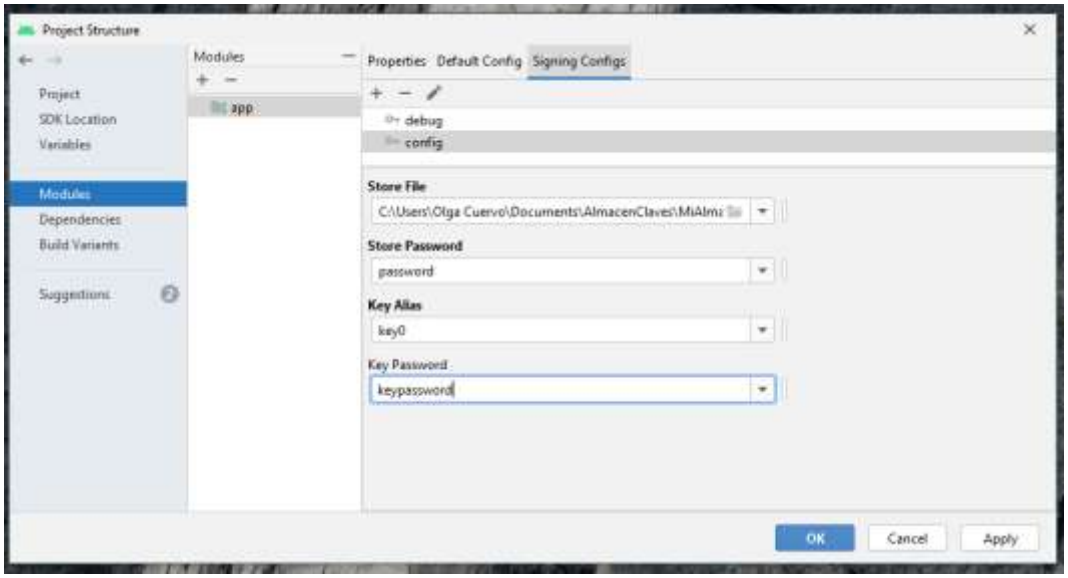
Una vez hecho, ya tenemos el almacén creado.

1.5. Firmando automáticamente una aplicación tanto en modo debug como release

- ✓ Podemos hacer que de forma automática el Android Studio *'firme'* la aplicación antes de instalarla en el emulador / dispositivo.

Para hacerlo, vamos a la ventana de **Project Structure → Modules**:

Vamos a la ventana **Signing Configs** y pulsamos el símbolo '**+**'. Indicamos los datos del almacén de claves creado previamente.



- ✓ Ahora debemos editar el archivo **build.gradle** a nivel de módulo.

- ✓ En este archivo estará la configuración creada en el paso anterior.
- ✓ Tenemos que modificar las entradas *'release'* y *'debug'* para que hagan uso del almacén de claves configurado anteriormente.