

Tutorial

Firmar el *.apk*



ANDROID PARTY 2014-07-10

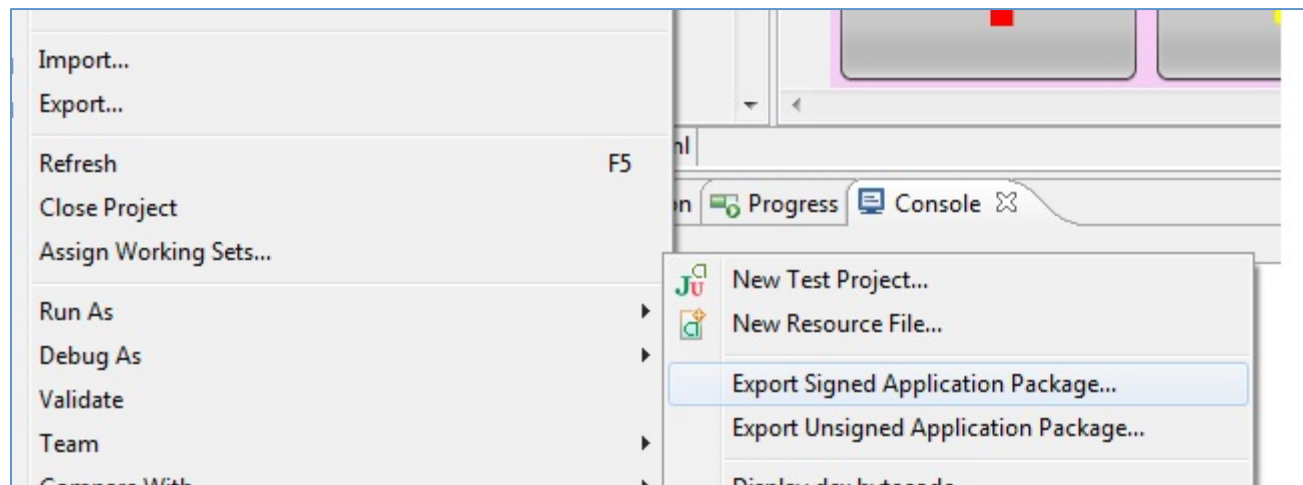


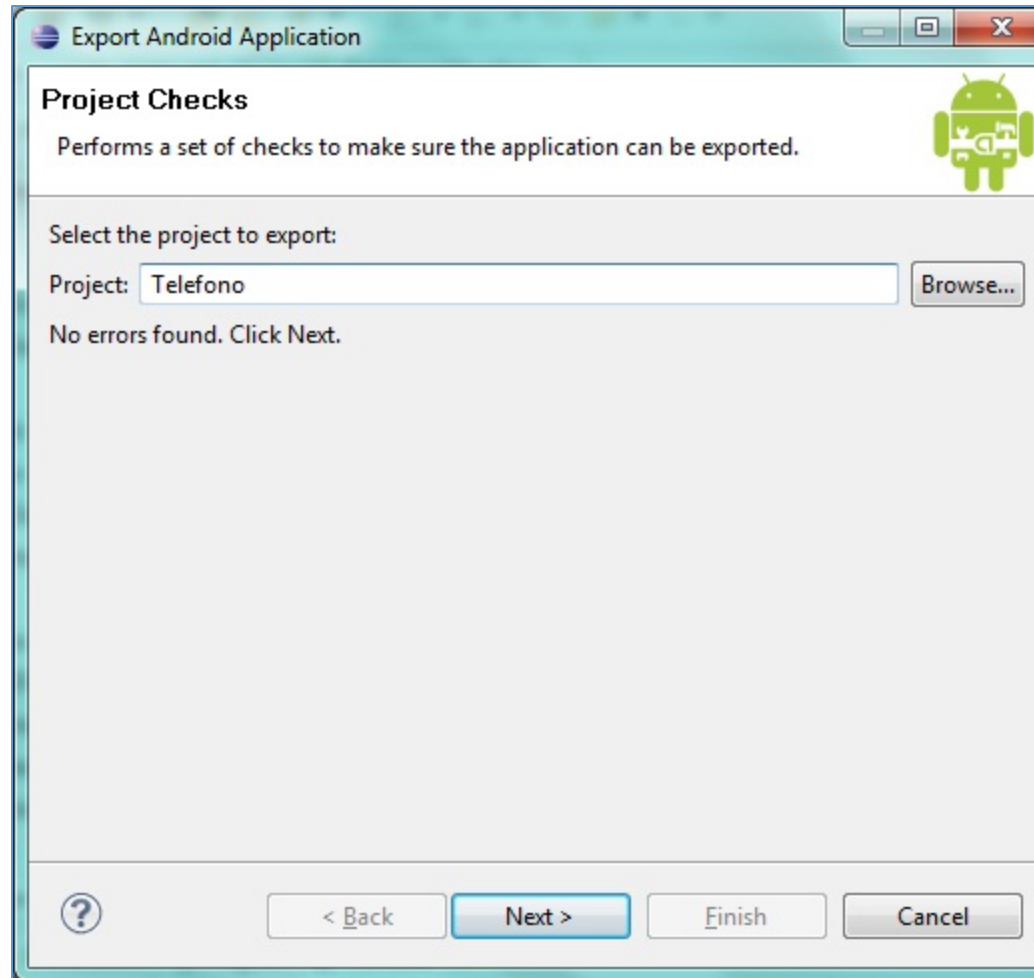
¿Qué es firmar el .apk?

- Se trata de crear el paquete **.apk** de nuestra aplicación para su posterior publicación y distribución desde el **market** de Android (**Google Play**).
- En este tutorial vamos a explicar como se realiza este procedimiento desde Eclipse.

- Al compilar nuestra aplicación Android desde Eclipse se genera el paquete .apk en "debug mode" y se le añade una key de depuración. Este paquete no es válido para su distribución en el market. Si queremos generar un paquete que esté firmado correctamente debemos seguir los siguientes pasos:


- 1) Pulsad con el botón derecho del ratón sobre el proyecto
- 2) Seleccionad "Android Tools" > "Export Signed Application Package"
- 3) Completad los datos de creación de la "key".





Export Android Application

Key Creation



Alias: pepe

Password: ●●●●●●

Confirm: ●●●●●●

Validity (years): 25

First and Last Name: pepe perez


Organizational Unit:

Organization:

City or Locality:

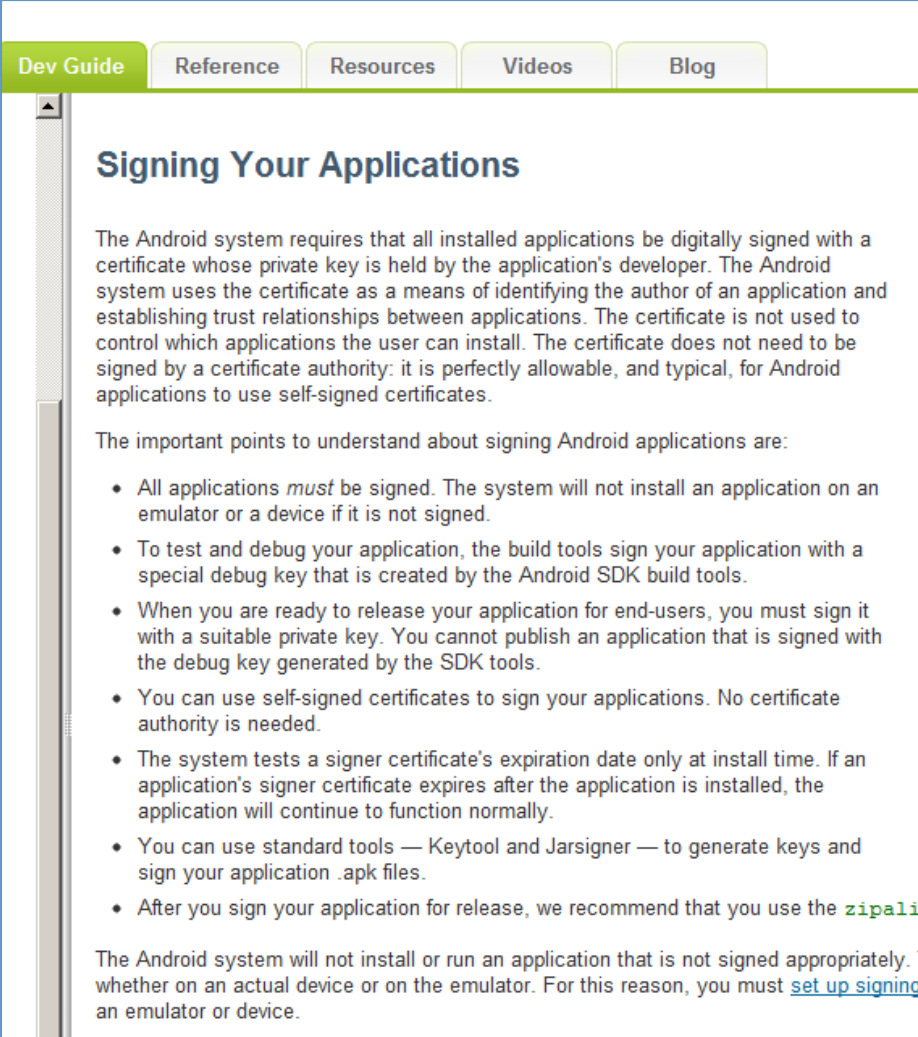
State or Province:

Country Code (XX):



Más información en:

<http://developer.android.com/guide/publishing/app-signing.html>



The screenshot shows a web page with a navigation bar at the top containing links for 'Dev Guide', 'Reference', 'Resources', 'Videos', and 'Blog'. The 'Dev Guide' link is highlighted in green. Below the navigation bar, the page title 'Signing Your Applications' is displayed in a large, bold, blue font. The main content area contains a paragraph explaining that the Android system requires all installed applications to be digitally signed with a certificate whose private key is held by the application's developer. It also mentions that the certificate is used to identify the author and establish trust relationships, but it is not used to control which applications the user can install. The certificate does not need to be signed by a certificate authority; it is perfectly allowable and typical for Android applications to use self-signed certificates.

The important points to understand about signing Android applications are:

- All applications *must* be signed. The system will not install an application on an emulator or a device if it is not signed.
- To test and debug your application, the build tools sign your application with a special debug key that is created by the Android SDK build tools.
- When you are ready to release your application for end-users, you must sign it with a suitable private key. You cannot publish an application that is signed with the debug key generated by the SDK tools.
- You can use self-signed certificates to sign your applications. No certificate authority is needed.
- The system tests a signer certificate's expiration date only at install time. If an application's signer certificate expires after the application is installed, the application will continue to function normally.
- You can use standard tools — Keytool and Jarsigner — to generate keys and sign your application .apk files.
- After you sign your application for release, we recommend that you use the `zipalign` tool to align the application.

The Android system will not install or run an application that is not signed appropriately. This applies whether on an actual device or on the emulator. For this reason, you must [set up signing](#) before you can run an application on an emulator or device.