This document assumes an Eclipse version with the Android Development plugin (SDK) already installed, which can be obtained from here: `http://developer.android.com/tools/sdk/eclipse-adt.html`; and also Android's Native Development Kit (NDK), which can be obtained here: `https://developer.android.com/tools/sdk/ndk/index.html`

The application code is easy to compile, once the tools are installed. It has to be imported to the workspace as an Android project (otherwise, the necessary libraries have to be manually referenced). That option is found in *File > Import*, selecting the option *Android/Existing Android code into workspace* (see 1).

To run the code on an emulator, we need to previously create its profile: *Window > Android Virtual Device Manager*, select *New*, and choose the features (see 2).

Once the profile is set up, compiling the code will automatically run it on the emulator.

Compiling the code for a real device is also easy, but it requires some previous steps.

First, the device must be enabled for application debugging. That option is found in Settings > Developer options (or, in Android versions pre-3.2, Settings > Applications > Developers), and enable USB debugging.

**Note**: in recent versions of Android (4.2 and ahead), the Developer options menu is hidden by default. To show it, go to Settings > Device information, and tap seven times on the *Build number* field.

If everything went well, in a command line window, from the path where the Android SDK is installed, enter the folder platform-tools, and run `adb devices`. In the displayed list, the device should appear as *device* and an identifier (fig 3).

In case it does not show, it may be necessary to install de drivers for the specific device, available from the website of the manufacturer.

Once these steps are done, pressing the Run button in Eclipse, with the device plugged in, will show a selector allowing to run the code in the phone.

**Note**: If the code fails to run due to ”not finding a compatible target”, change in the file *AndroidManifest.xml*, the line
`android:targetSdkVersion="17"` to the version the phone uses. Still, we remind that the application relies on features of Android 14, so it will not work on devices previous to 4.0.

Compiling the native side of the code is a bit more complex, as this feature is not yet integrated with Eclipse.

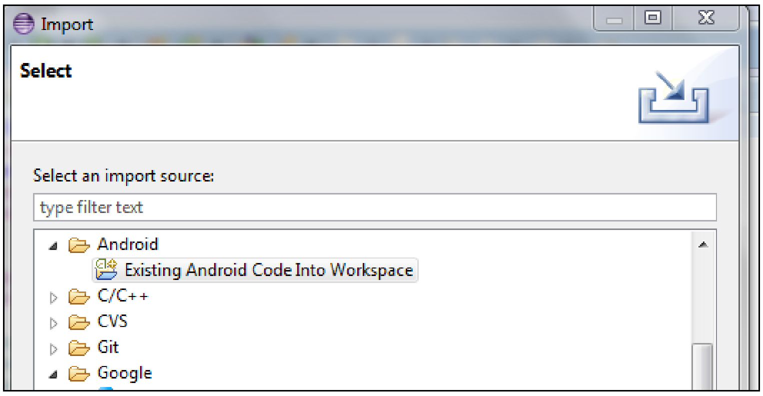
The Java side uses the native code as a precompiled binary file. This means that, as long as the C part remains unmodified, it is not necessary to recompile it to run the project.

The Android native code compiler uses a syntax similar to Makefile's; the compilation directives are, inside the project, in the path *jni/Android.mk*. This file should not be modified unless more native code packages were necessary.

From the command line (figure 4; pictured: Windows' cmd), launch the ndk-build program, available in the NDK intall folder, from the main directory of the project.

This generates again the binary files, which are automatically created in the format required by the Java side.

**Note**: for some OSs (especially older versions of Windows) the compilation from the default command line may provoke some issues. In that case, we recommend using a Linux-style command window, such as Cygwin (available here: `http://www.cygwin.com/`).

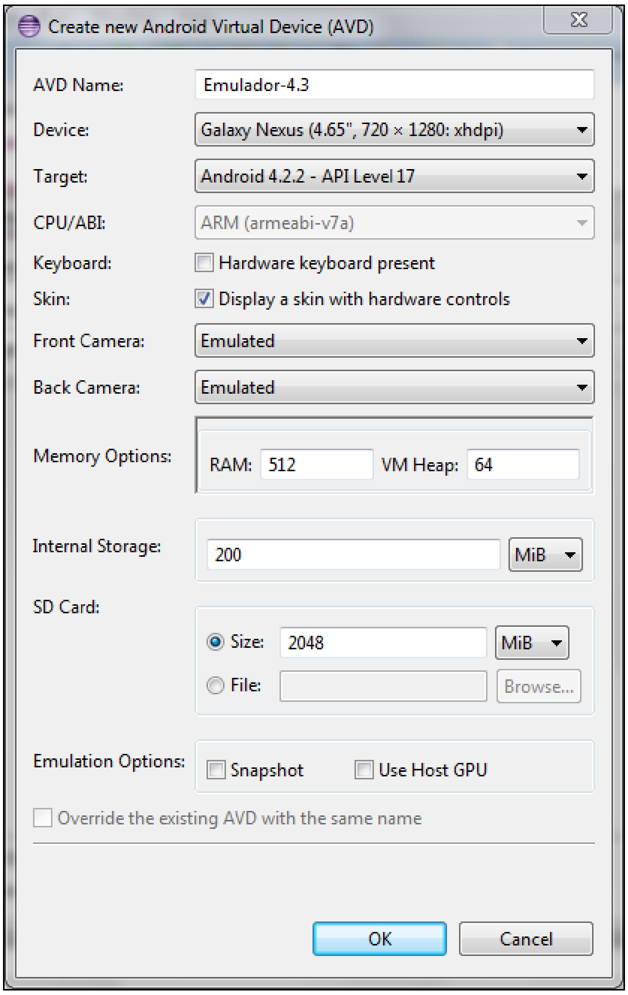


Select an import source:

type filter text

- Android
  - Existing Android Code Into Workspace
- C/C++
- CVS
- Git
- Google

Figura 1: Importing an Android project



AVD Name: Emulador-4.3

Device: Galaxy Nexus (4.65", 720 × 1280: xhdpi)

Target: Android 4.2.2 - API Level 17

CPU/ABI: ARM (armeabi-v7a)

Keyboard: ☐ Hardware keyboard present

Skin: ☑ Display a skin with hardware controls

Front Camera: Emulated

Back Camera: Emulated

Memory Options: RAM: 512 VM Heap: 64

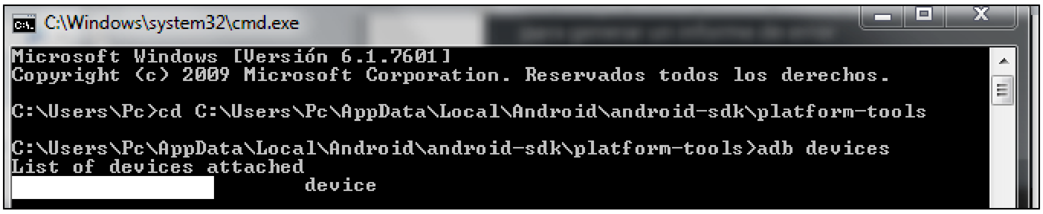
Internal Storage: 200 MiB

SD Card:
● Size: 2048 MiB
○ File: Browse...

Emulation Options: ☐ Snapshot ☐ Use Host GPU

☐ Override the existing AVD with the same name

Figura 2: Setting the emulator profile

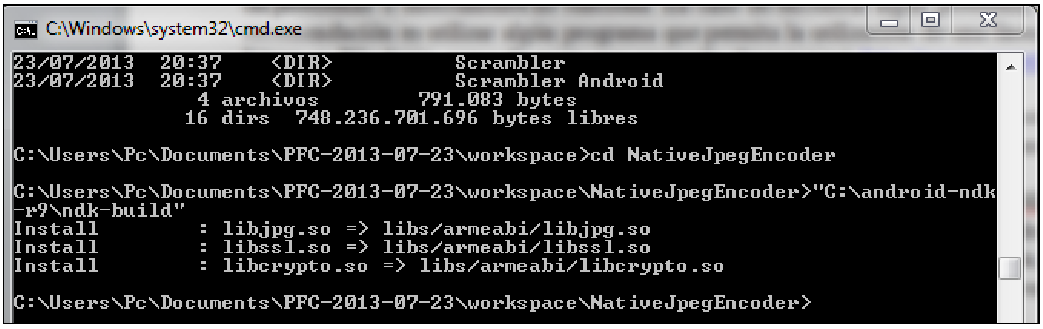


```
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\Pc>cd C:\Users\Pc\AppData\Local\Android\android-sdk\platform-tools

C:\Users\Pc\AppData\Local\Android\android-sdk\platform-tools>adb devices
List of devices attached
[redacted]         device
```

Figura 3: Output of *adb devices*



```
23/07/2013  20:37    <DIR>          Scrambler
23/07/2013  20:37    <DIR>          Scrambler Android
               4 archivos        791.083 bytes
              16 dirs  748.236.701.696 bytes libres

C:\Users\Pc\Documents\PFC-2013-07-23\workspace>cd NativeJpegEncoder

C:\Users\Pc\Documents\PFC-2013-07-23\workspace\NativeJpegEncoder>"C:\android-ndk
-r9\ndk-build"
Install        : libjpg.so => libs/armeabi/libjpg.so
Install        : libssl.so => libs/armeabi/libssl.so
Install        : libcrypto.so => libs/armeabi/libcrypto.so

C:\Users\Pc\Documents\PFC-2013-07-23\workspace\NativeJpegEncoder>
```

Figura 4: Running *ndk-build*