



[www.centic.es](http://www.centic.es)

# Curso Android Edición 2013



CENTRO TECNOLÓGICO DE LAS TECNOLOGÍAS DE  
LA INFORMACIÓN Y LAS COMUNICACIONES



# Introducción

## ¿Qué es Android?

- Android es un conjunto de software.
- Incluye un sistema operativo.
- El SDK dispone de herramientas y APIs para el desarrollo de aplicaciones.
- Se programa «usando» lenguaje JAVA
- Ejecuta aplicaciones sobre una maquina virtual. «DALVIK»

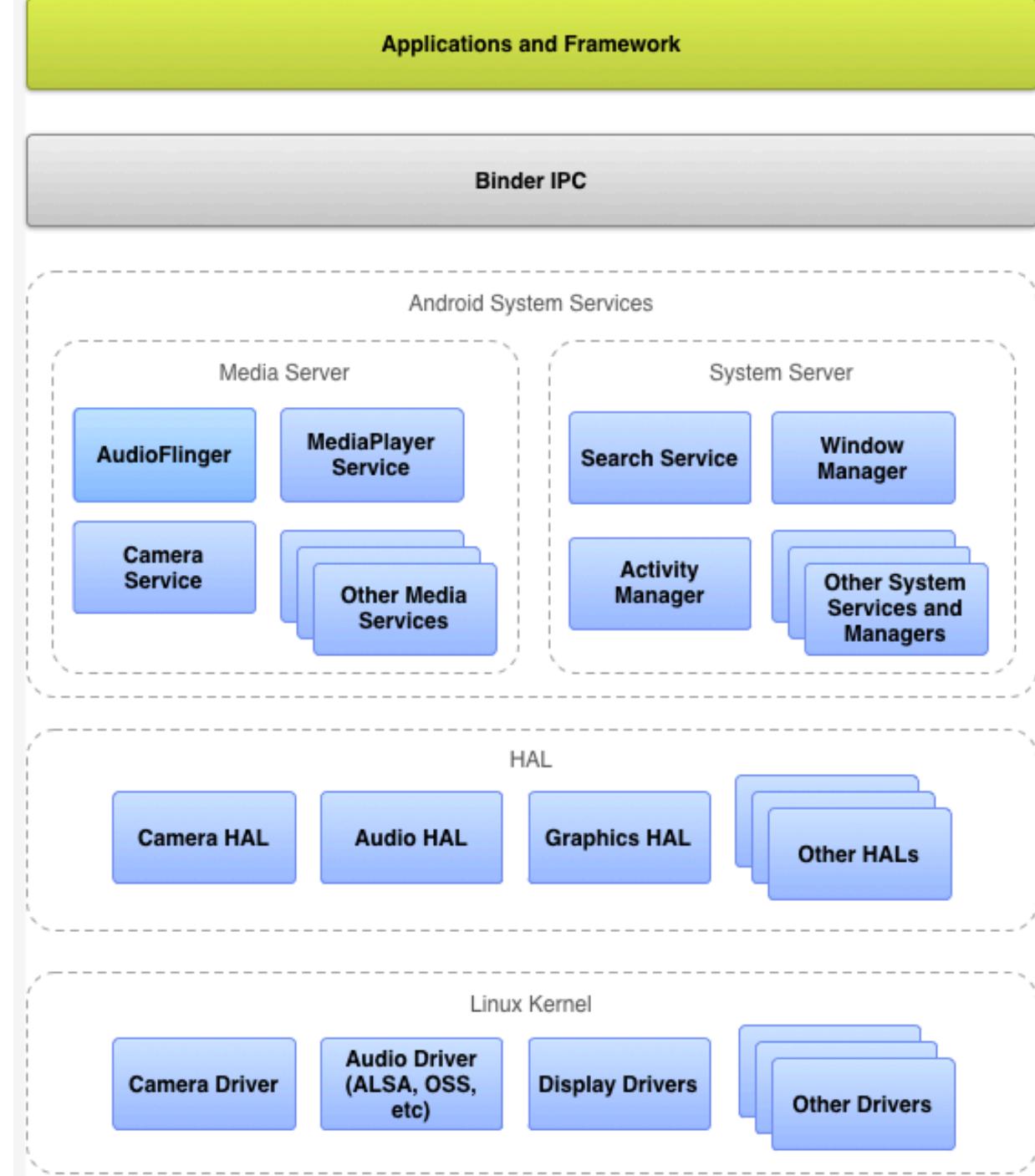
## Features

- Application framework enabling reuse and replacement of components
- Dalvik virtual machine optimized for mobile devices
- Integrated browser based on the open source [WebKit](#) engine
- Optimized graphics powered by a custom 2D graphics library; 3D graphics based on the OpenGL ES 1.0 specification (hardware acceleration optional)
- SQLite for structured data storage
- Media support for common audio, video, and still image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- GSM Telephony (hardware dependent)
- Bluetooth, EDGE, 3G, and WiFi (hardware dependent)
- Camera, GPS, compass, and accelerometer (hardware dependent)
- Rich development environment including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the Eclipse IDE



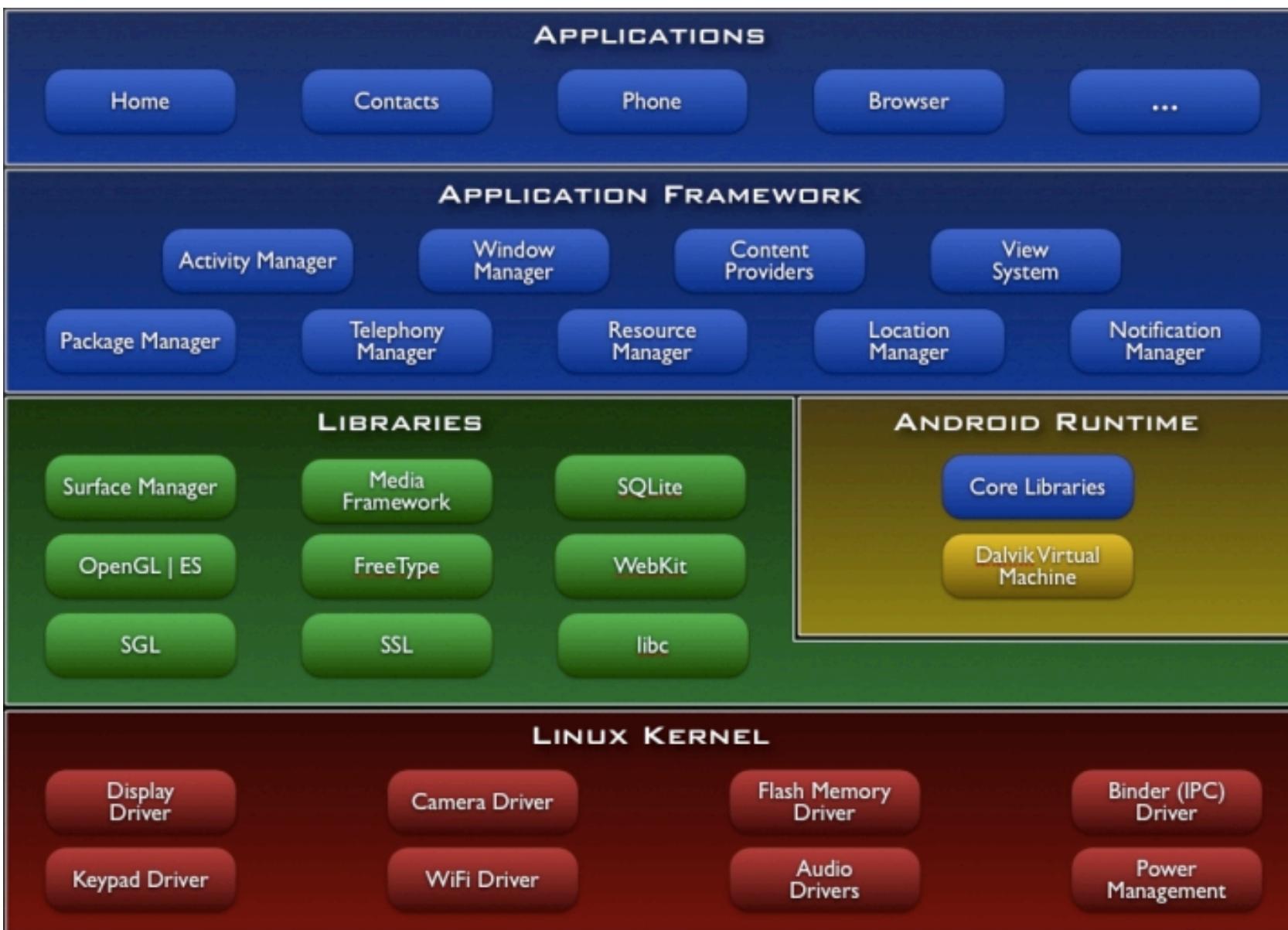
# Introducción

## Arquitectura



# Introducción

## Arquitectura



# Introducción

## Arquitectura

Framework de Aplicaciones.

- Views
- Content Providers
- Resource Manager
- Notification Manager
- Activity Manager



# Introducción

## Arquitectura

### Bibliotecas.

- System C Library.
- Bibliotecas multimedia «Opencore»
- Surface Manager
- LibWebCore
- SGL «Gráficos 2D»
- OpenGL ES 1.1 / 2.0 «Gráficos 3D»
- FreeType
- SQLite



# Introducción

## Arquitectura

### Entorno de Ejecución

- Android incluye un conjunto de bibliotecas que proporcionan la mayor parte de la funcionalidad disponible en las bibliotecas del núcleo del lenguaje de programación Java.
- Cada aplicación Android se ejecuta en su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito para que un dispositivo pueda ejecutar múltiples máquinas virtuales de manera eficiente.
- La máquina virtual Dalvik ejecuta archivos ejecutables en formato dex, este formato está optimizado para consumo de memoria mínima.
- El VM es basado en registros, y corre clases compiladas por un compilador de lenguaje Java que se han convertido en formato dex. El compilador «DX» va incluido en el «SDK».
- La máquina virtual Dalvik se basa en el kernel de Linux.



# Introducción

## Arquitectura

### Kernel de Linux 2.6/3.0/3.3/3.4/¿3.8?

- Android se basa en el de Linux para el sistema de servicios básicos, tales como la seguridad, la gestión de memoria, gestión de procesos, pila de red, y el modelo del controlador.
- El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software



# Introducción

## Versiones

Code name	Version	API level
(no code name)	1.0	API level 1
(no code name)	1.1	API level 2
Cupcake	1.5	API level 3, NDK 1
Donut	1.6	API level 4, NDK 2
Eclair	2.0	API level 5
Eclair	2.0.1	API level 6
Eclair	2.1	API level 7, NDK 3
Froyo	2.2.x	API level 8, NDK 4
Gingerbread	2.3 - 2.3.2	API level 9, NDK 5
Gingerbread	2.3.3 - 2.3.7	API level 10
Honeycomb	3.0	API level 11
Honeycomb	3.1	API level 12, NDK 6
Honeycomb	3.2.x	API level 13
Ice Cream Sandwich	4.0.1 - 4.0.2	API level 14, NDK 7
Ice Cream Sandwich	4.0.3 - 4.0.4	API level 15, NDK 8
Jelly Bean	4.1.x	API level 16
Jelly Bean	4.2.x	API level 17

# Introducción

## ADB – Android Debug Bridge

Sintaxis:

```
adb [-d|-e|-s <serialNumber>] <command>
```

-d | -e | -s Son necesarios cuando tenemos mas de un dispositivo conectado y necesitamos especificar cual de ellos es el destino del comando.

**-d** Cuando es el único dispositivo USB.

**-e** Cuando solo hay un Emulador.

**-s** El dispositivo cuyo número de serie es el q especificamos.



# Introducción

## ADB – Android Debug Bridge

Ejemplos:

adb devices, lista los dispositivos conectados. [serialNumber] [state]

state, offline indica desconectado o sin respuesta, device indica que el servidor está operativo, no device, no hay ningún emulador/dispositivo conectado.

```
adb devices
List of devices attached
emulator-5554    device
emulator-5556    device
emulator-5558    device
```



# Introducción

## ADB – Android Debug Bridge

Ejemplos:

adb –s<serial> <command>, Permite ejecutar comandos en un dispositivo.

adb –s<serial> install myfirstapp.apk, instala la aplicación myfirstapp en el dispositivo.

adb –s<serial> pull <remote> <local>, copia el fichero o directorio remote, situado en la raiz del dispositivo en local del ordenador.

adb –s<serial> push, inversa de pull.

adb –s<serial> shell, abre una linea de comandos dentro del dispositivo.



# Introducción

## ADT – Android Developer Toolkit

Provee de una interfaz gráfica para realizar las tareas de prototipado, diseño y desarrollo de las interfaces de usuario de las aplicaciones Android.

Integración del SDK de Android:

- Traceview.
- Android.
- Hierarchy Viewer.
- Pixel Perfect.
- DDMS.
- Adb.
- ProGuard



# Introducción

ADT – Android Developer Toolkit

Editor de Código.

- Editor Gráfico.
- Editor Android Manifest
- Editor Menú.
- Editor de Recursos.
- Editor XML.

Otras acciones.

- Distribución.
- Compilar.
- Firmar.
- Debuguear
- Testear.



# Introducción

Android. (Integrada en ADT)

Esta herramienta nos permite:

- Gestionar ADVs, Android Virtual Devices.
- Crear y actualizar proyectos.
- Actualizar el SDK

## Syntax

```
android [global options] action [action options]
```

## Global Options

**-s**

Silent mode: only errors are printed out

**-h**

Usage help

**-v**

Verbose mode: errors, warnings and informational messages are printed.

# Introducción

bgmr. (Backup Manager)

Esta herramienta nos permite:

Gestionar el gestor de copias de seguridad de los dispositivos android, evitando tener que realizar las tareas de limpieza de estos datos cada vez que queremos realizar una prueba en el dispositivo.

Ejecutar una copia.

```
adb shell bmgr backup <packagename>
```

Ejecutar restauración.

```
adb shell bmgr restore <packagename>
```

Eliminar datos.

```
adb shell bmgr wipe <packagename>
```



# Introducción

## Draw 9-patch

Esta herramienta nos permite:

Crear gráficos NinePatch usando un editor WYSIWYG.

Un gráfico NinePatchDrawable es una imagen de mapa de bits “estirable”, a la que Android va a cambiar automáticamente el tamaño para acomodar su contenido a la vista en la que se ha colocado.

El borde se utiliza para definir las zonas “estirables” y estáticas de la imagen.

Se indica una sección extensible dibujando una (o más) línea negra de 1 píxel de ancho en la parte izquierda y la parte superior de la frontera



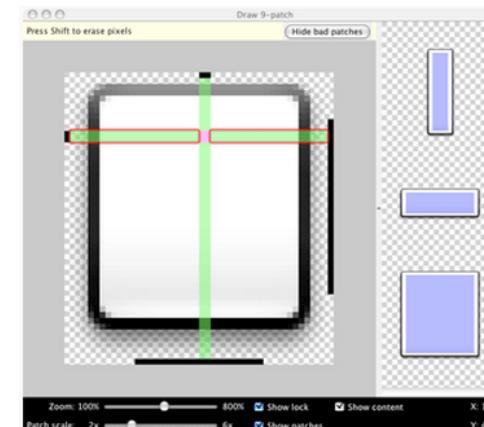
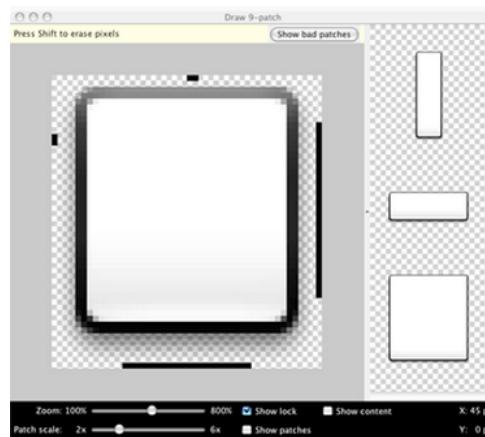
# Introducción

## Draw 9-patch

Esta herramienta nos permite:

Los demás píxeles del borde deben ser totalmente transparentes o blancos.

Su tamaño relativo se mantiene igual, por lo que las secciones más importantes son siempre las más grandes.



# Introducción

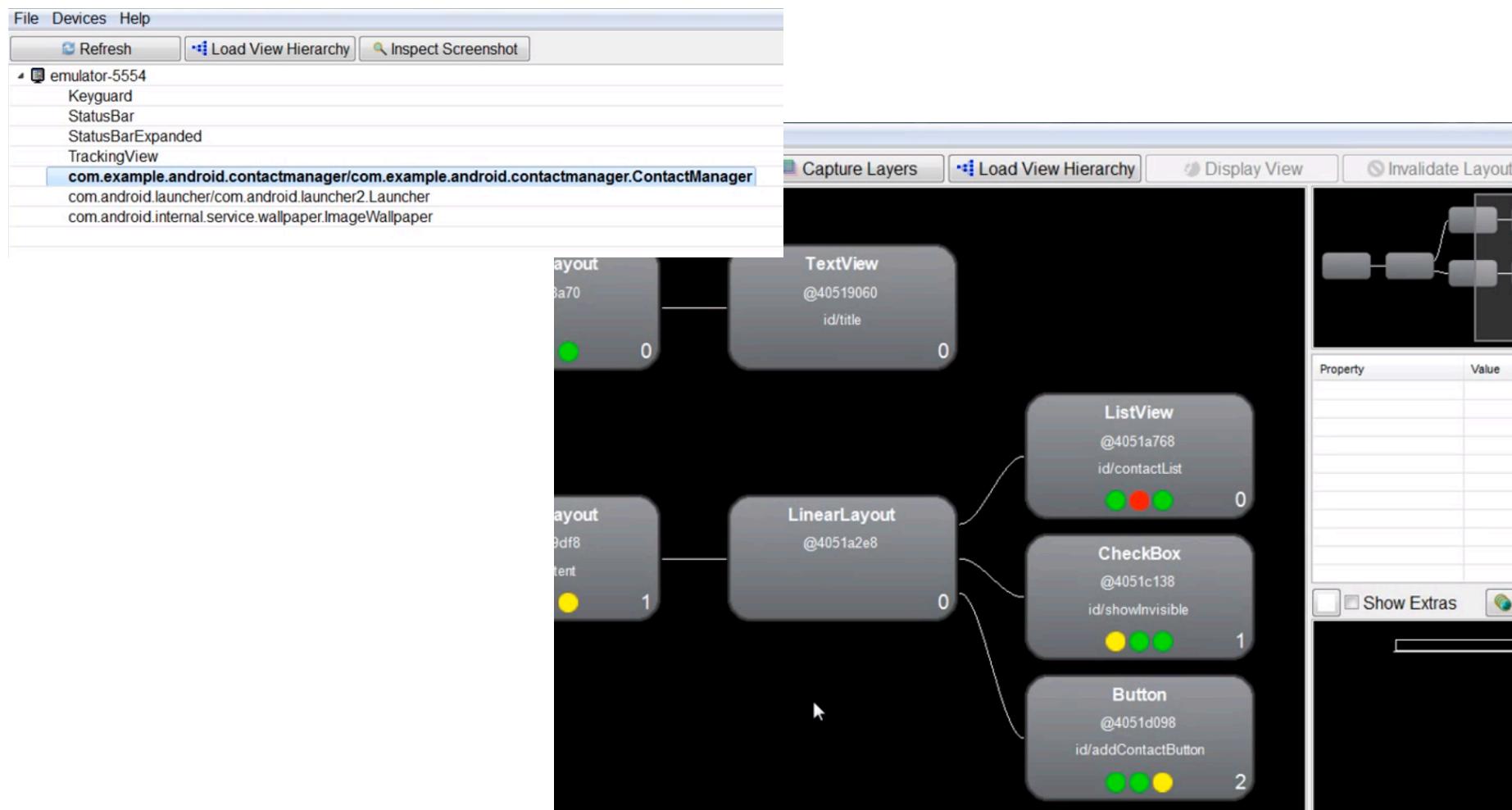
Emulador.

Emulated Device Key	Keyboard Key
Home	HOME
Menu (left softkey)	F2 or Page-up button
Star (right softkey)	Shift-F2 or Page Down
Back	ESC
Call/dial button	F3
Hangup/end call button	F4
Search	F5
Power button	F7
Audio volume up button	KEYPAD_PLUS, Ctrl-F5
Audio volume down button	KEYPAD_MINUS, Ctrl-F6
Camera button	Ctrl-KEYPAD_5, Ctrl-F3
Switch to previous layout orientation (for example, portrait, landscape)	KEYPAD_7, Ctrl-F11
Switch to next layout orientation (for example, portrait, landscape)	KEYPAD_9, Ctrl-F12
Toggle cell networking on/off	F8
Toggle code profiling	F9 (only with <code>-trace</code> startup option)
Toggle fullscreen mode	Alt-Enter
Toggle trackball mode	F6
Enter trackball mode temporarily (while key is pressed)	Delete
DPad left/up/right/down	KEYPAD_4/8/6/2
DPad center click	KEYPAD_5
Onion alpha increase/decrease	KEYPAD_MULTIPLY(*) / KEYPAD_DIVIDE(/)

# Introducción

Hierarchy Viewer.

Permite inspeccionar la jerarquía visual de una actividad.



# Introducción

Lint.

Es una herramienta de análisis de código estático que comprueba los archivos del proyecto buscando posibles errores y mejoras de optimización para la corrección, la seguridad, el rendimiento, la usabilidad, la accesibilidad y la internacionalización.

Esta integrado dentro del ADT.

The screenshot shows the 'Lint Warnings' tab in the Android Studio interface. It displays a table of issues found in the project. A tooltip is open over the second item in the list, providing detailed information about an unused resource warning.

Description	Category	Location	
<uses-sdk> tag appears after <application>	Correctness	AndroidManifest.xml:23 (Earthquake 3)	
The resource R.layout.preferences appears to	Performance	preferences.xml in layout (Earthquake 3)	The resource R.layout.preferences appears to be unused
Missing density variation folders in res: drawa	Usability:Icons	Earthquake 3	Issue: Looks for unused resources Id: UnusedResources
android:enabled is deprecated: Use state_ena	Correctness	earthquake_map.xml:11 in layout (Earthquake 3)	Unused resources make applications larger and slow down builds.
<uses-sdk> tag should specify a target API le	Correctness	AndroidManifest.xml:23 (Earthquake 3)	



# Introducción

## Logcat.

Proporciona un mecanismo para recopilar y ver la salida de depuración del sistema.

Los registros de las aplicaciones se recogen en un fichero y se pueden ver y ser filtrados por el comando Logcat.

Podemos utilizar Logcat dentro de ADB para ver los mensajes de registro.

For example:

```
Log.i("MyActivity", "MyClass.getView() - get item number " + position);
```

The LogCat will then output something like:

```
I/MyActivity( 1557): MyClass.getView() - get item number 1
```

# Introducción

Monkey.

Es un programa que se ejecuta en el emulador o dispositivo y genera ejecuciones pseudo-aleatorias de eventos de usuario, como clics, toques o gestos, así como una serie de eventos a nivel de sistema.

Ideal para pruebas de estrés sobre la aplicación que se está desarrollando, de forma aleatoria pero repetible.

Sintaxis:

```
$ adb shell monkey [options] <event-count>
```

Ejemplo:

```
$ adb shell monkey -p your.package.name -v 500
```



# Introducción

Monkeyrunner.

La herramienta monkeyrunner proporciona una API para controlar un dispositivo o emulador Android.

Nos permite escribir un script Python que instala una aplicación, la ejecuta, envía las pulsaciones del teclado a la misma, toma capturas de pantalla de su interfaz de usuario, y almacena imágenes en el ordenador.

La herramienta monkeyrunner está diseñada principalmente para probar aplicaciones y para ejecutar conjuntos de pruebas unitarias.

```
monkeyrunner -plugin <plugin_jar> <program_filename> <program_options>
```



# Introducción

UIAutomator.

La herramienta de pruebas uiautomator permite probar la interfaz de usuario (UI) de manera eficiente mediante la creación de casos de prueba de interfaz de usuario funcionales automatizadas que se pueden ejecutar en una aplicación en uno o más dispositivos.

Sintaxis:

```
adb shell uiautomator runtest <jar> -c <test_class_or_method> [options]
```

Ejemplo:

```
adb shell uiautomator runtest LaunchSettings.jar -c com.uia.example.my.LaunchSettings
```



```
// Simulate a short press on the HOME button.  
getUiDevice().pressHome();  
  
// We're now in the home screen. Next, we want to simulate  
// a user bringing up the All Apps screen.  
// If you use the uiautomatorviewer tool to capture a snapshot  
// of the Home screen, notice that the All Apps button's  
// content-description property has the value "Apps". We can  
// use this property to create a UiSelector to find the button.  
UiObject allAppsButton = new UiObject(new UiSelector()  
    .description("Apps"));  
  
// Simulate a click to bring up the All Apps screen.  
allAppsButton.clickAndWaitForNewWindow();  
  
// In the All Apps screen, the Settings app is located in  
// the Apps tab. To simulate the user bringing up the Apps tab,  
// we create a UiSelector to find a tab with the text  
// label "Apps".  
UiObject appsTab = new UiObject(new UiSelector()  
    .text("Apps"));  
  
// Simulate a click to enter the Apps tab.  
appsTab.click();
```

# Entorno de desarrollo ECLIPSE

**Descargar el framework.**

<http://developer.android.com/sdk/index.html>

Platform	Package	Size	MD5 Checksum
Windows 32 & 64-bit	<a href="#">android-sdk_r22-windows.zip</a>	107505668 bytes	71722fe052ae6380444a21bce8ee87c2
	<a href="#">installer_r22-windows.exe</a> (Recommended)	87498295 bytes	e0cc167733bf8b51dbc7e0ad0a8c8d4b
Mac OS X 32 & 64-bit	<a href="#">android-sdk_r22-macosx.zip</a>	71244523 bytes	fa5193ad41edecac6960023f55569ba3
Linux 32 & 64-bit	<a href="#">android-sdk_r22-linux.tgz</a>	99643077 bytes	30fb75bad918c5c3d79f8ec3cc44b3cf

**Descargar eclipse.**

<http://eclipse.org/downloads/>

Eclipse Downloads

Packages Developer Builds Projects

Compare Packages Older Versions Eclipse Helios (3.6.2) Packages for Windows

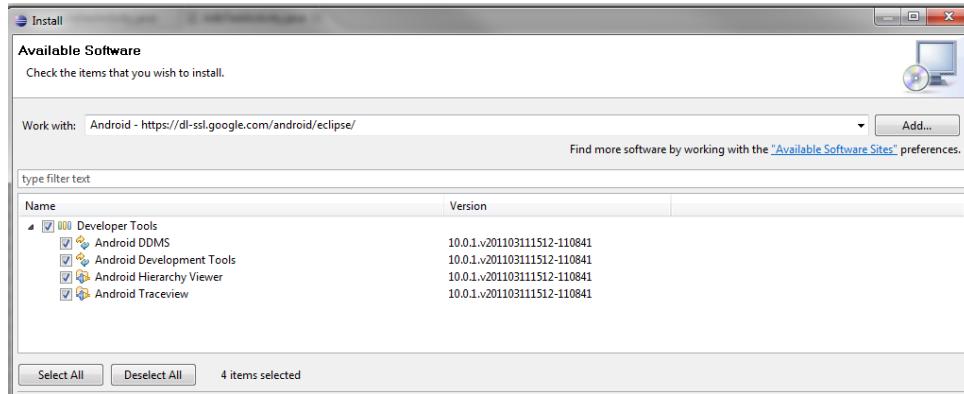
 <a href="#">Eclipse IDE for Java Developers</a> , 99 MB Downloaded 1,278,597 Times <a href="#">Details</a>	 <a href="#">Windows 32 Bit</a> <a href="#">Windows 64 Bit</a>
 <a href="#">Eclipse IDE for Java EE Developers</a> , 206 MB Downloaded 934,609 Times <a href="#">Details</a>	 <a href="#">Windows 32 Bit</a> <a href="#">Windows 64 Bit</a>
 <a href="#">Eclipse Classic 3.6.2</a> , 171 MB Downloaded 822,124 Times <a href="#">Details</a> Other Downloads	 <a href="#">Windows 32 Bit</a> <a href="#">Windows 64 Bit</a>



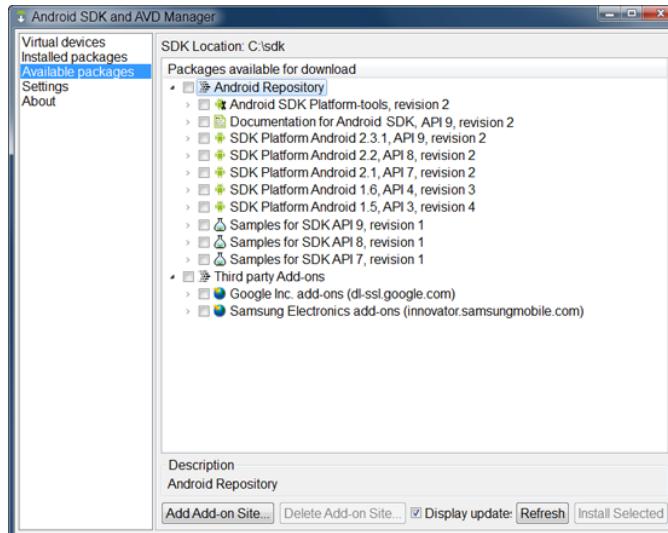
# Entorno de desarrollo

## Instalar ADT para eclipse.

<https://dl-ssl.google.com/android/eclipse/>

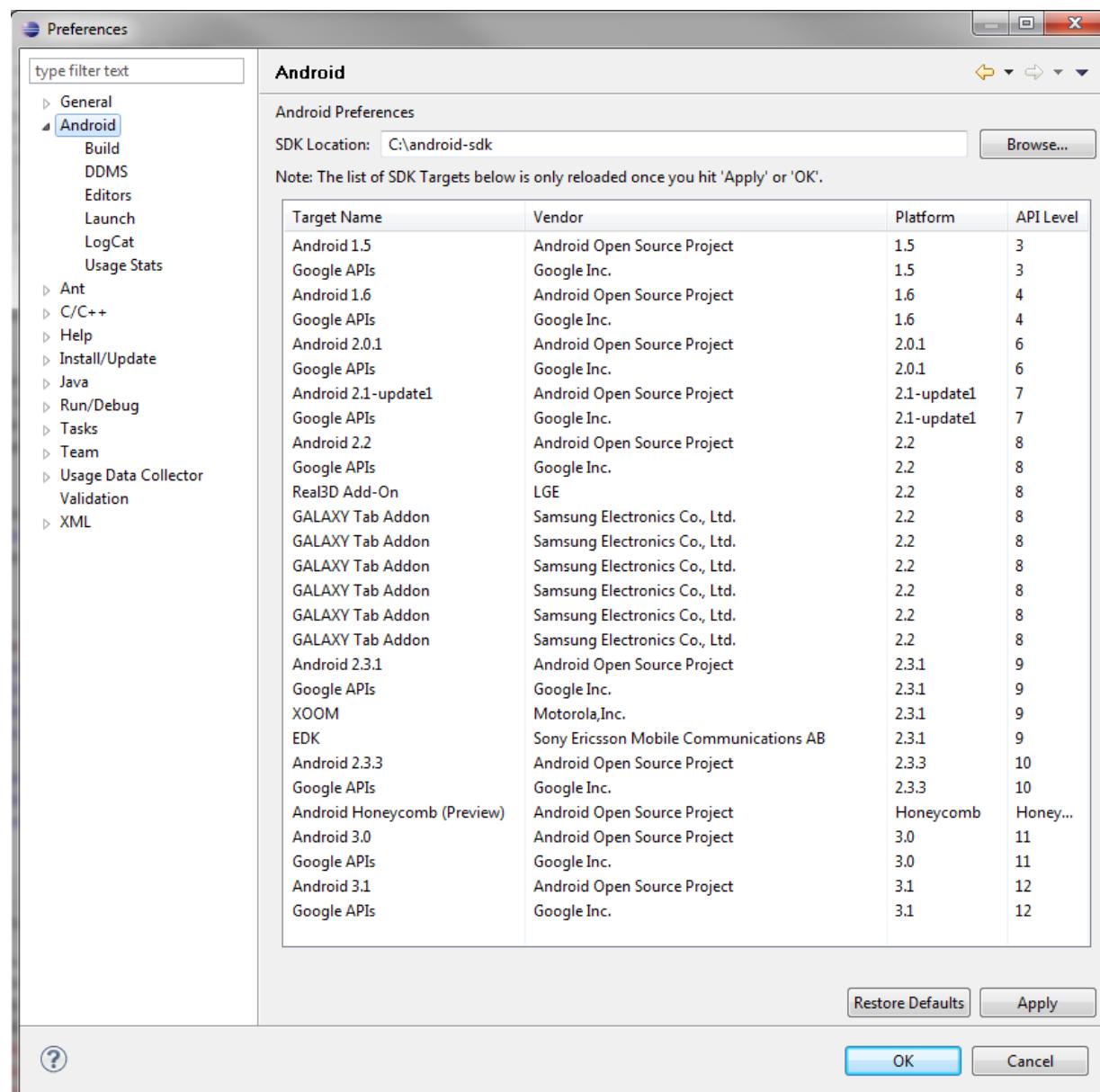


## Añadir plataformas de desarrollo y crear AVDs



# Entorno de desarrollo

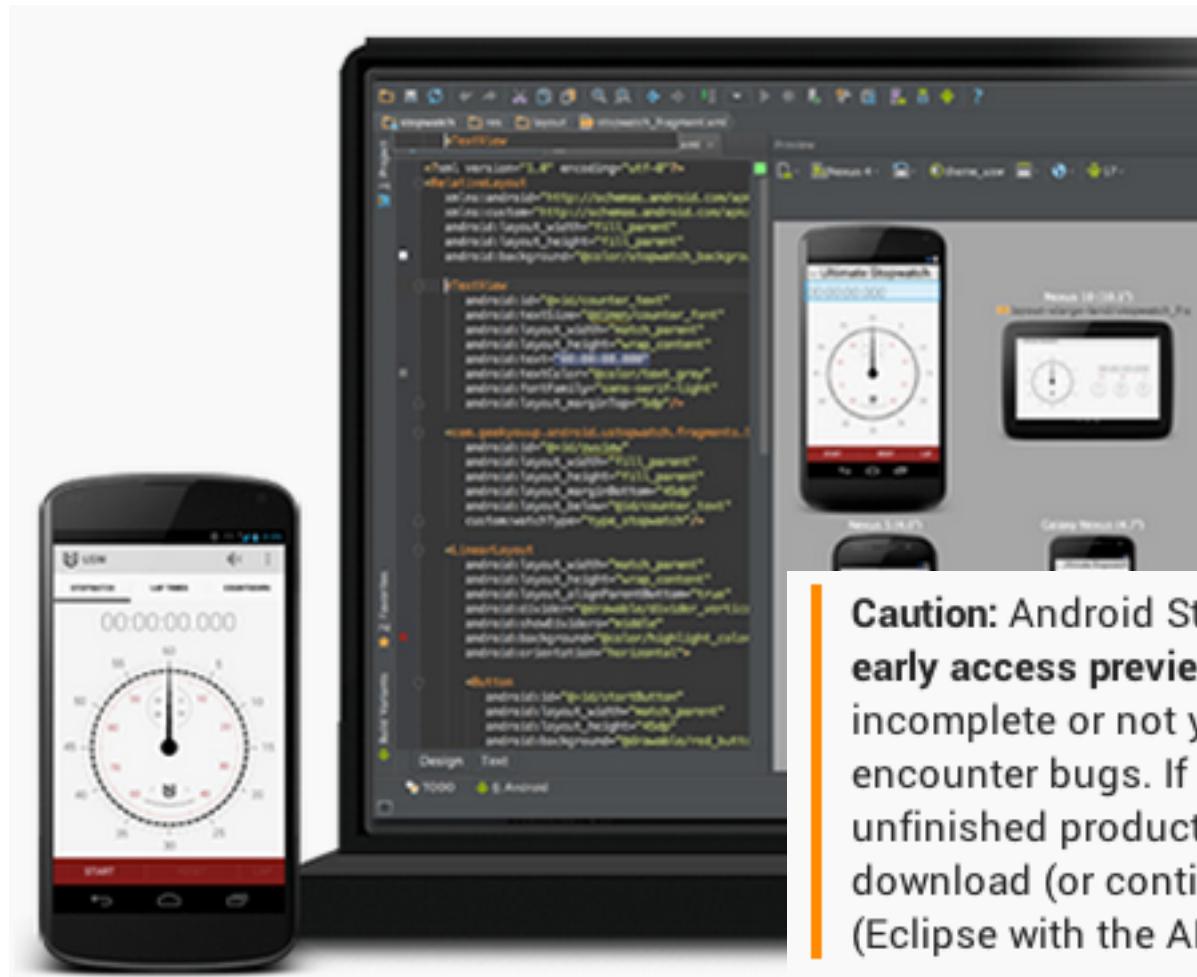
## Configurar ADT.



# Entorno de desarrollo Android Studio

## Descargar Android Studio.

<http://developer.android.com/intl/es/sdk/installing/mac-studio>



**Caution:** Android Studio is currently available as an **early access preview**. Several features are either incomplete or not yet implemented and you may encounter bugs. If you are not comfortable using an unfinished product, you may want to instead download (or continue to use) the **ADT Bundle** (Eclipse with the ADT Plugin).

# Entorno de desarrollo Android Studio

Basado en IntelliJ IDEA – Es un Port! – No es una Extensión

Soporte para Código.

Soporte para Compilación.

Soporte para Refactorización.

Soporte para Depuración

Soporte para UI. (Mejoras apreciables!!!)

Open Source – AOSP –

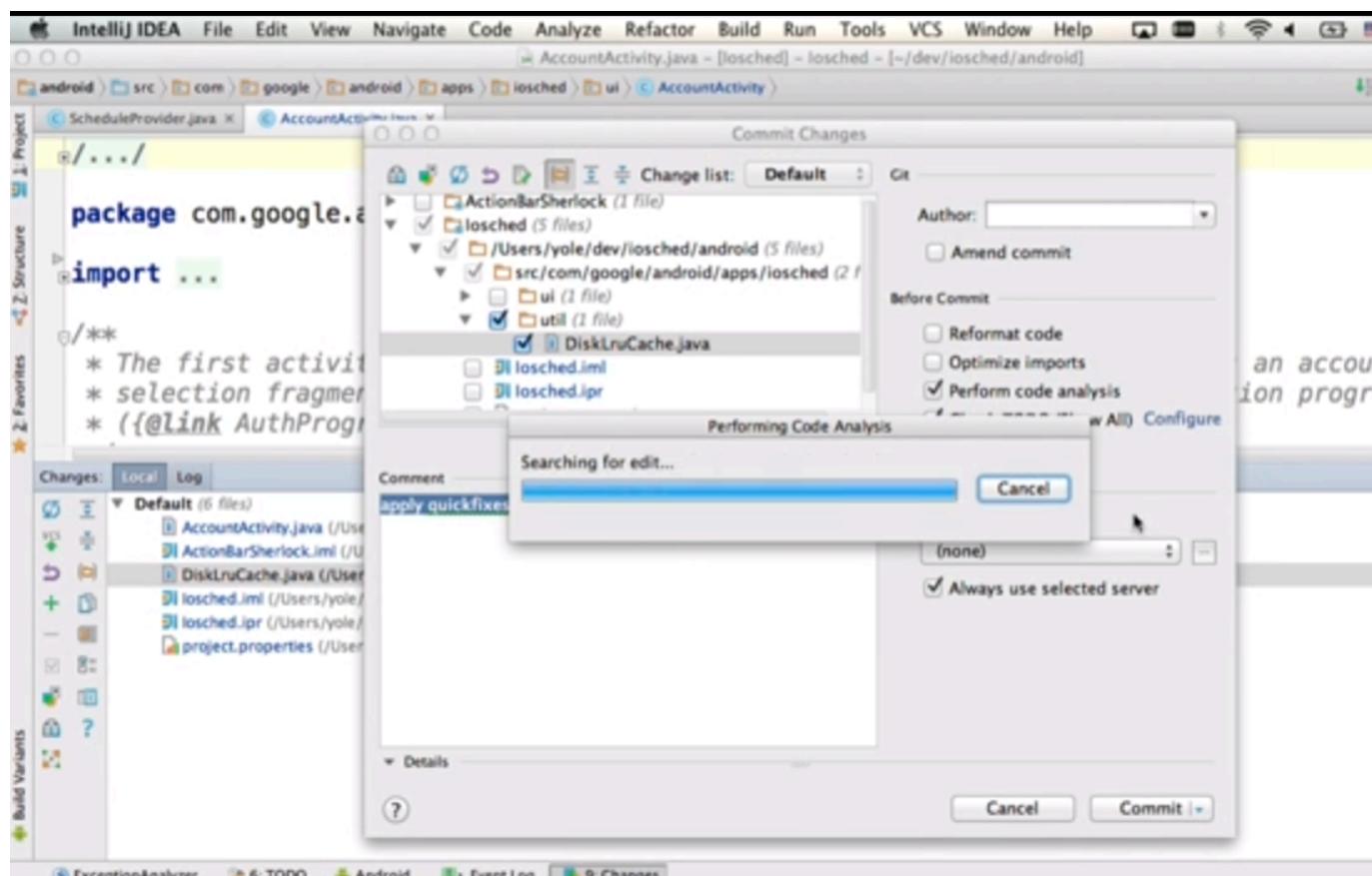
Tor Norbye: IntelliJ es totalmente estable.... Estamos trabajando en Android Preview 0.1.1



# Entorno de desarrollo Android Studio

## Integración con Sistemas de Control de Versiones.

- SVN
- Mercurial
- GIT



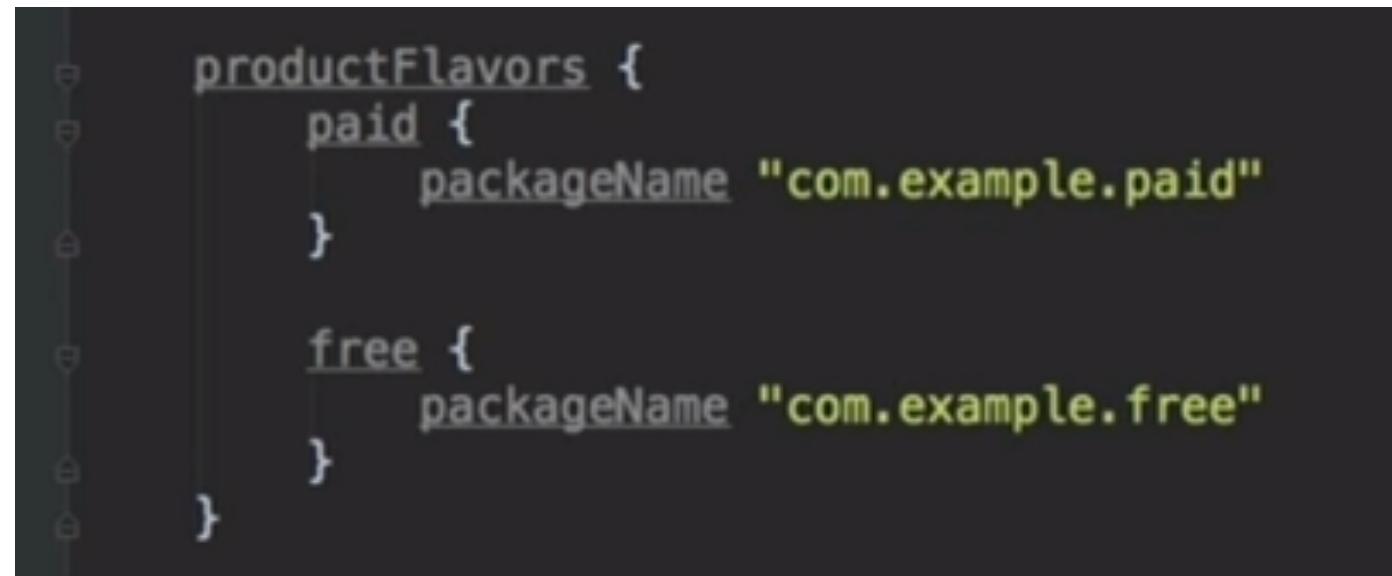
# Entorno de desarrollo Android Studio

Estructura de proyecto basada en Gradle.

Permite mantener la consistencia de un proyecto entre diferentes IDEs/Plataformas.

Permite mantener distintas construcciones de un APK.

- Debug
- Lite
- Premium
- Tablet



```
productFlavors {  
    paid {  
        packageName "com.example.paid"  
    }  
  
    free {  
        packageName "com.example.free"  
    }  
}
```

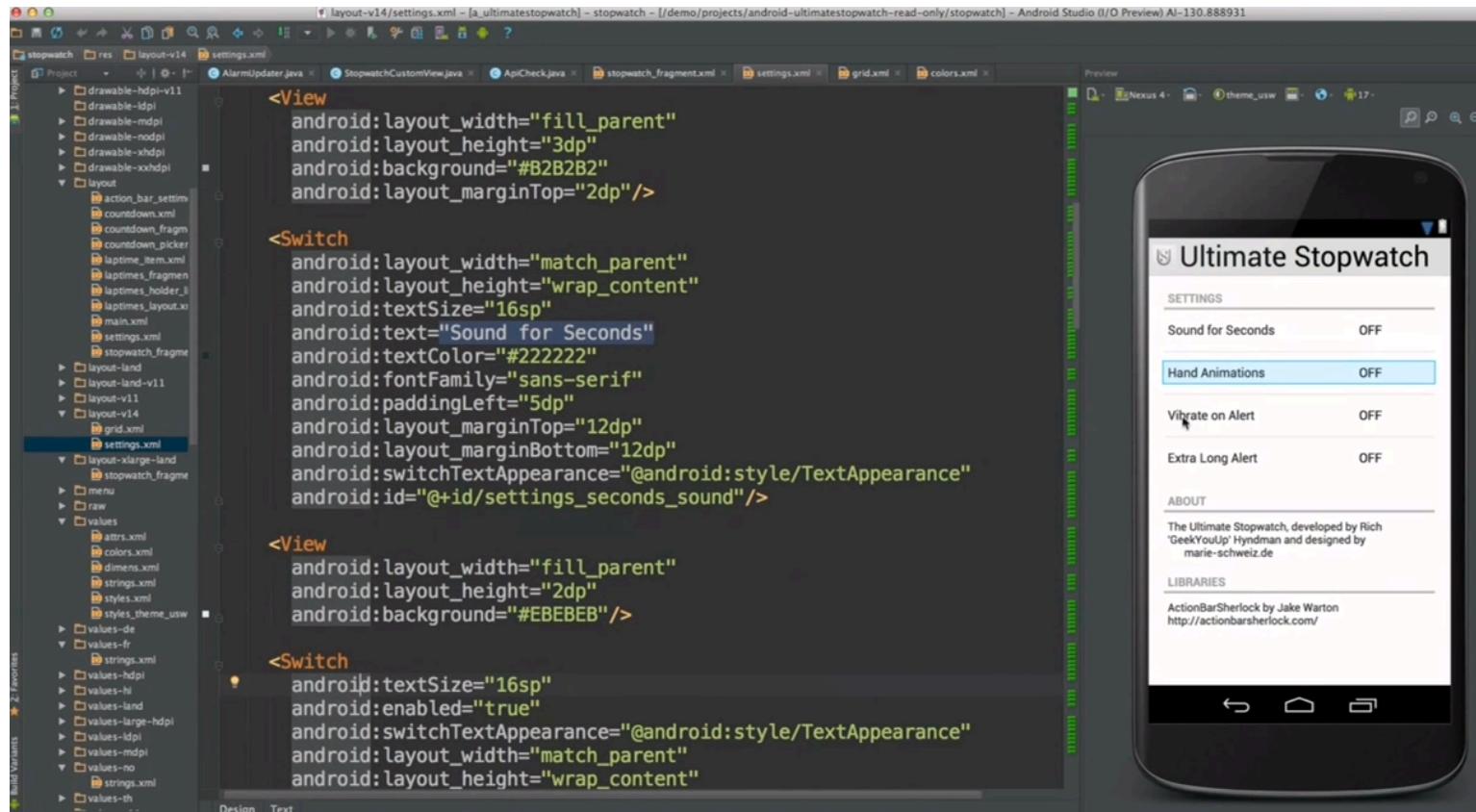
A screenshot of the Android Studio code editor showing the build.gradle file. The file defines two product flavors: 'paid' and 'free'. Each flavor has its own package name assigned. The code is written in Groovy-like syntax with indentation and curly braces.

# Entorno de desarrollo

## Android Studio

Mejoras en el Editor Gráfico.

Navegación entre XML y su representación.



# Entorno de desarrollo Android Studio

Mejoras en el Editor  
Gráfico.

Representación en  
distintos dispositivos.

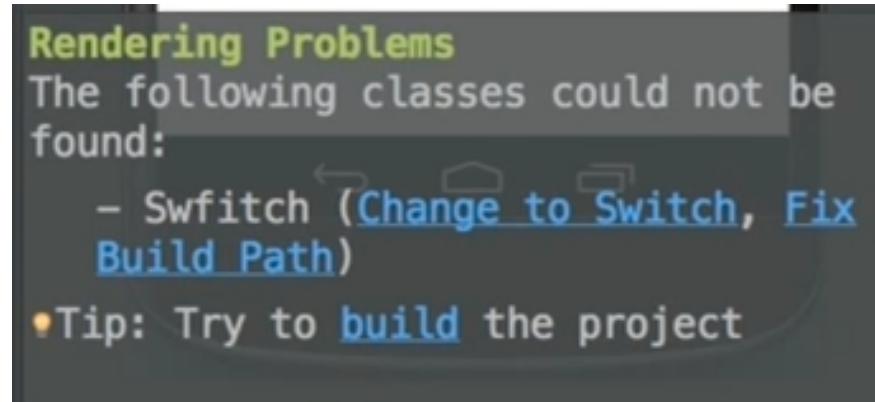


# Entorno de desarrollo Android Studio

Mejoras en el Editor Gráfico.

Panel de Errores. (“Typos” y Rendering fails)

```
<Swfitch
    android:textSize="16sp"
    android:enabled="true"
    android:switchTextAppearance="@android:s
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Hand Animations"
    android:textColor="#222222"
    android:fontFamily="sans-serif"
    android:paddingLeft="5dp"
    android:layout_marginTop="12dp"
    android:layout_marginBottom="12dp"
    android:id="@+id/settings_animations"/>
```



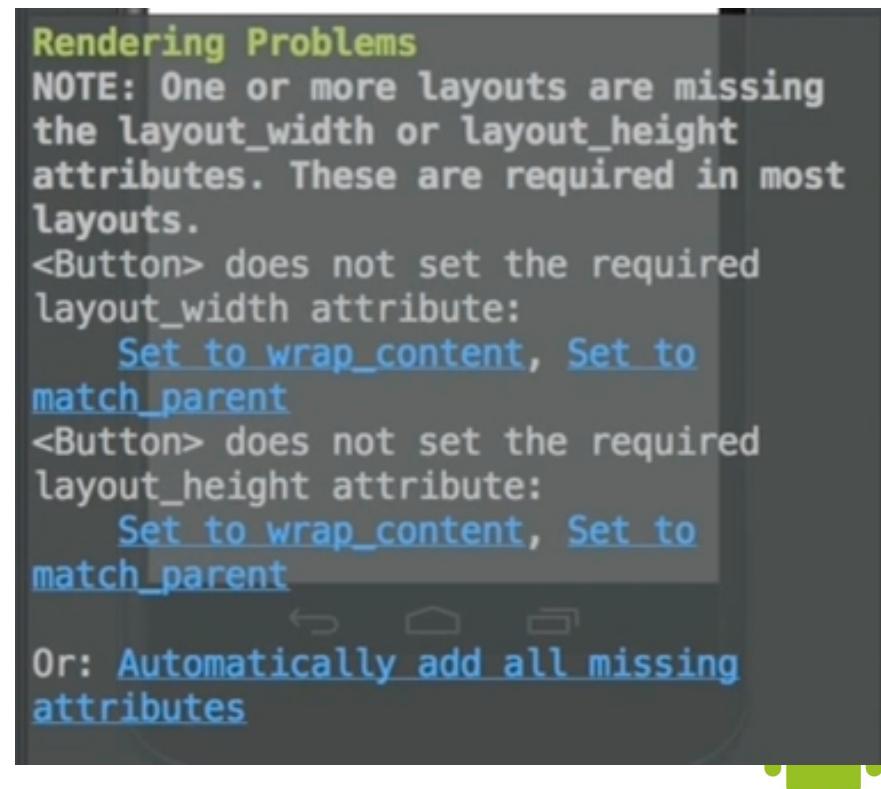
# Entorno de desarrollo Android Studio

Mejoras en el Editor Gráfico.

Panel de Errores. (“Typos” y Rendering fails)

A screenshot of the Android Studio XML layout editor. On the left, there's a tree view showing a RelativeLayout containing a TextView and a fragment. The TextView has attributes android:layout\_width="wrap\_content" and android:layout\_height="wrap\_content". The fragment tag is highlighted in red with the error message "Typos" above it. The main area shows the XML code:

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Hello world!" />  
  
<fragment  
/>  
  
</RelativeLayout>
```



# Entorno de desarrollo Android Studio

Mejoras en el Editor Gráfico.

Mejor soporte para los Layouts.

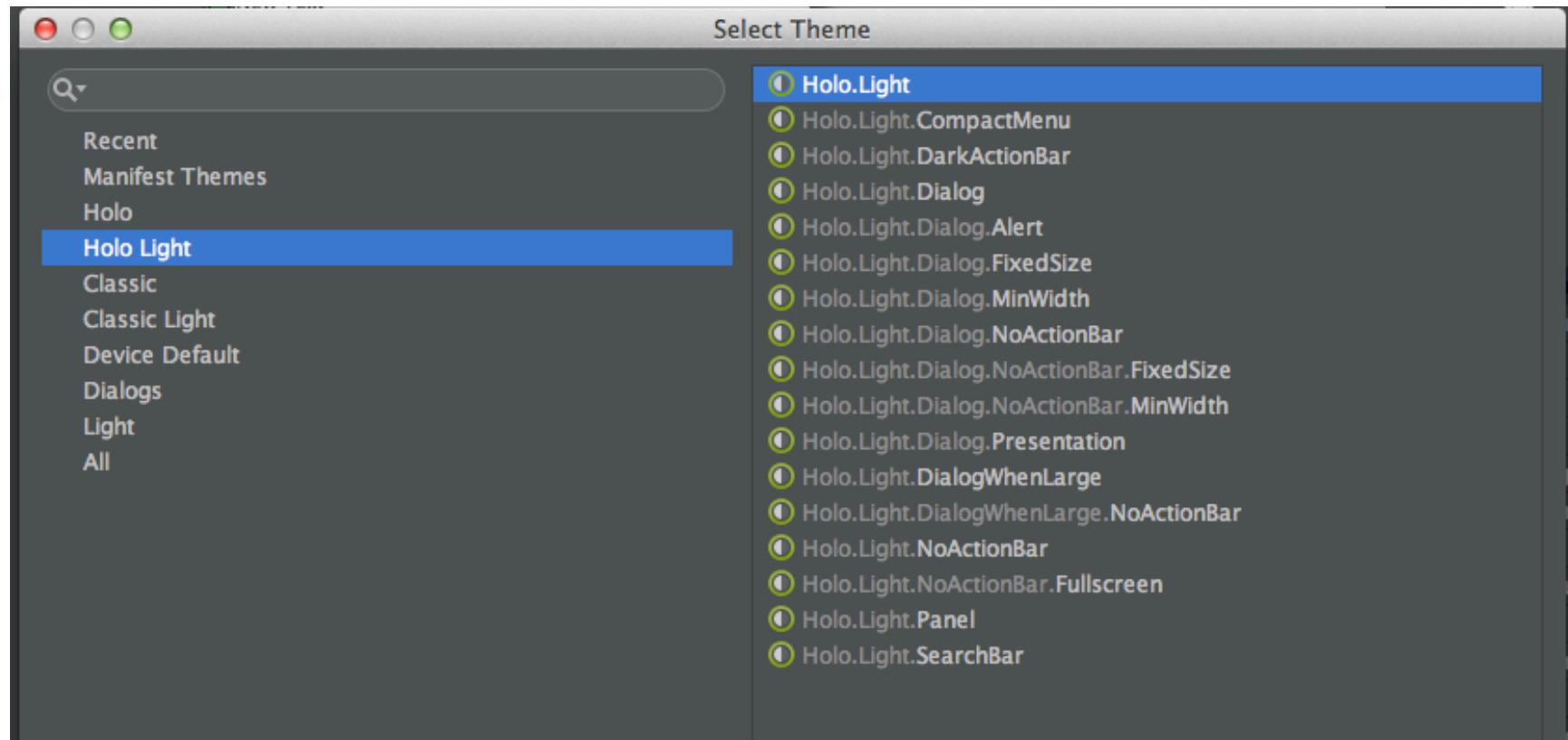


# Entorno de desarrollo

## Android Studio

Mejoras en el Editor Gráfico.

Render para los Themes.



# Tipos de Proyecto

## Estructura.

- res/anim/  
Contiene las animaciones del proyecto
- res/color/  
Contiene los colores usados en la aplicación
- res/drawable/  
Contiene las imágenes usadas en el proyecto
- res/layout  
Contiene las interfaces graficas de usuario
- res/menu/  
Contiene la estructura de los menús
- res/raw/  
Cualquier fichero que no encaje en el resto de directorios y que queramos referenciar desde «R.java»



# Tipos de Proyecto

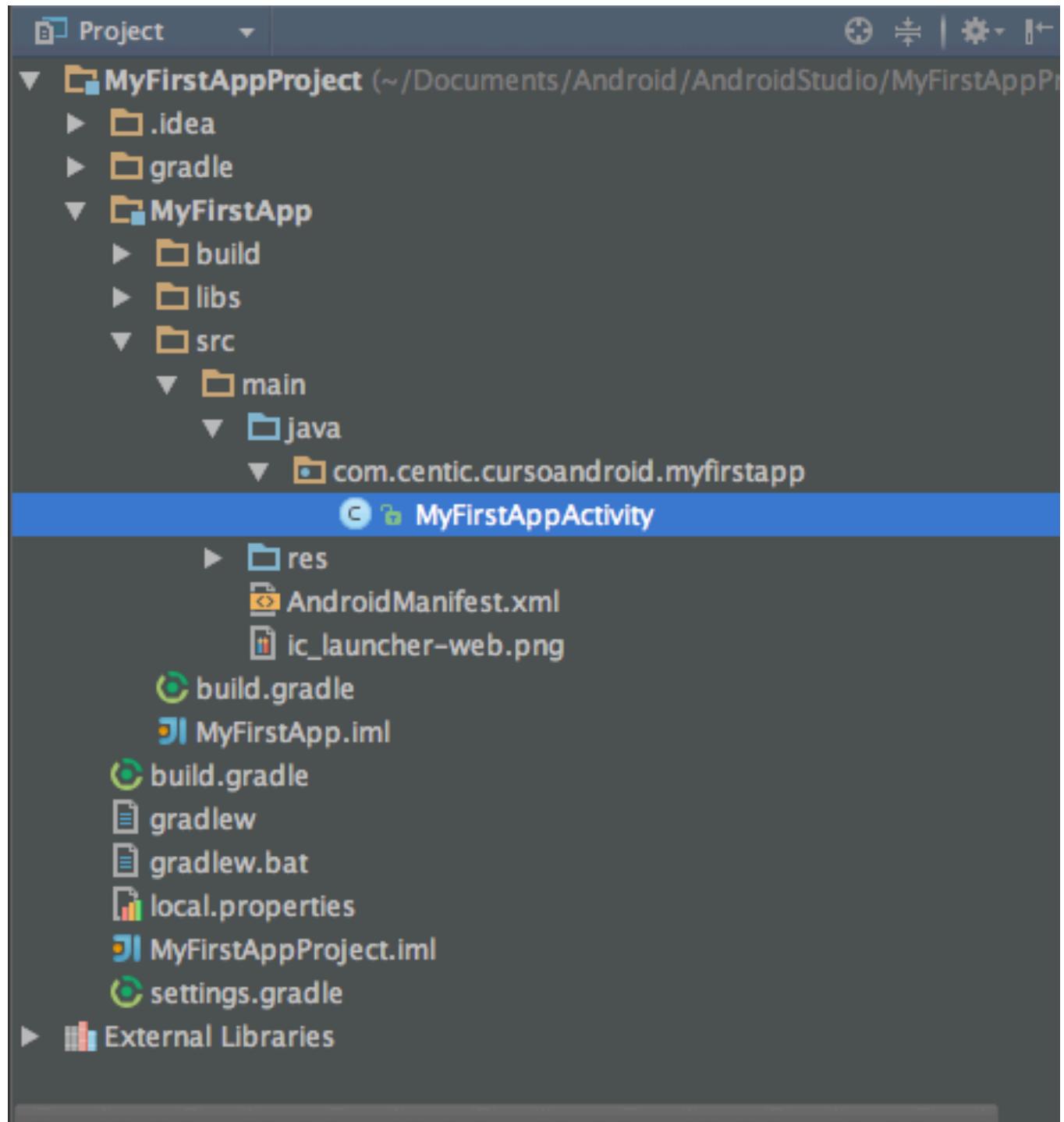
## Estructura.

- res/values  
Contiene los valores como cadenas de texto que queramos referenciar desde «R.java»
- res/xml  
Contiene los fichero de configuración de nuestra aplicación tales como preferencias, widget o metadatos.
- libs/  
Contiene librerias necesitadas por el proyecto. «.jar»
- AndroidManifest.xml  
Es el fichero que describe nuestra aplicación. Detalla los componentes de esta, sus permisos y librerias externas.



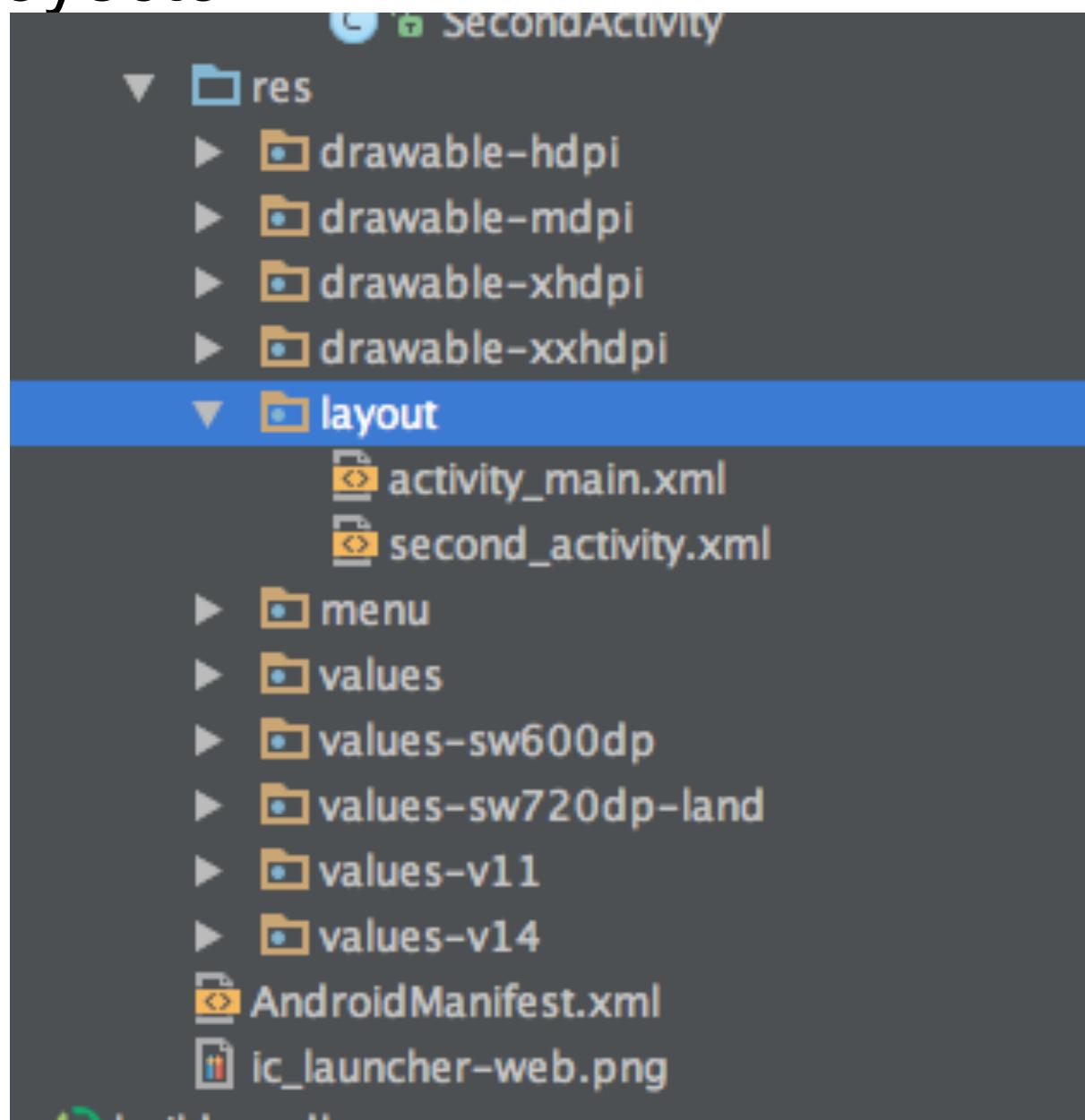
# Tipos de Proyecto

Estructura.



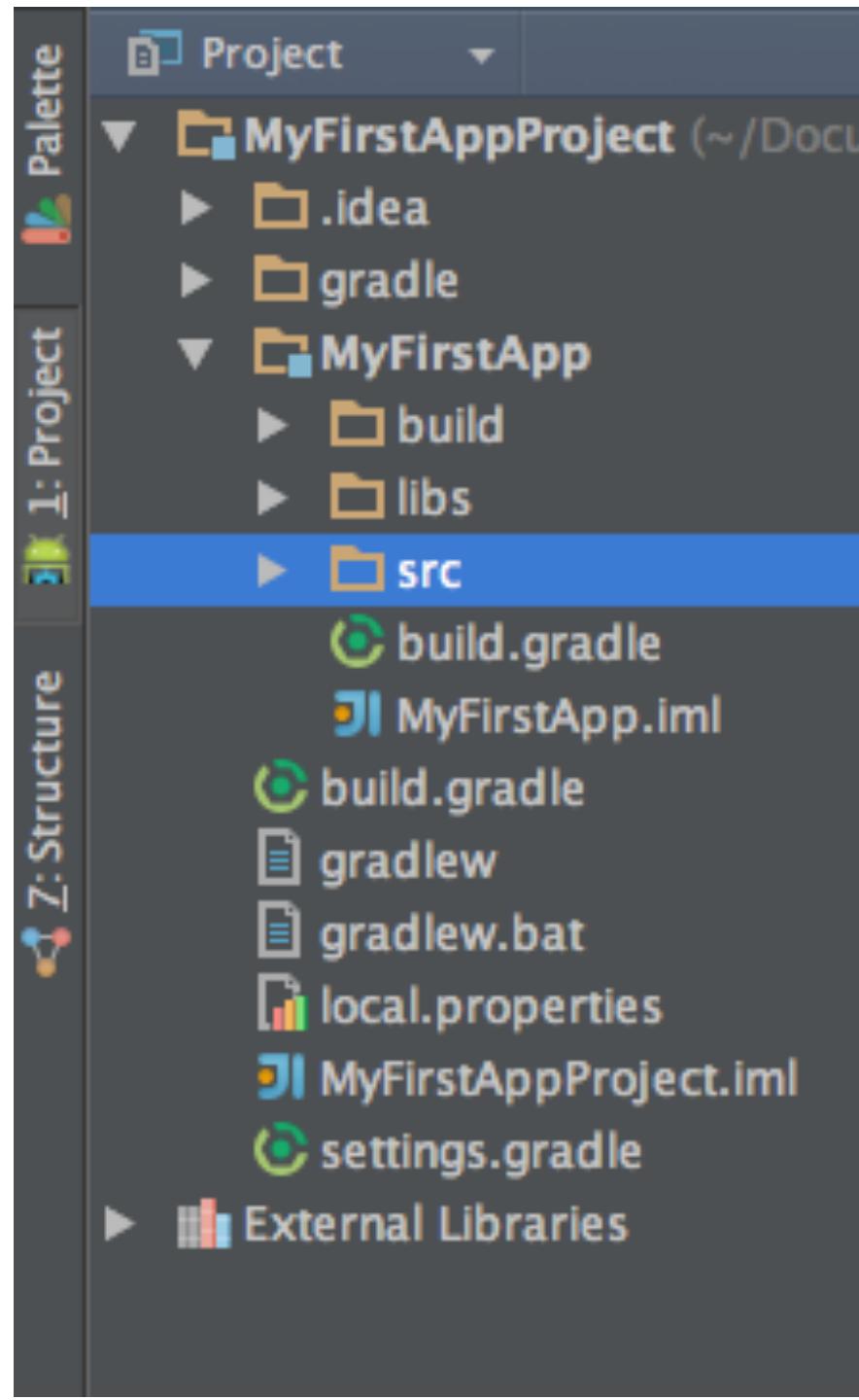
# Tipos de Proyecto

Estructura.



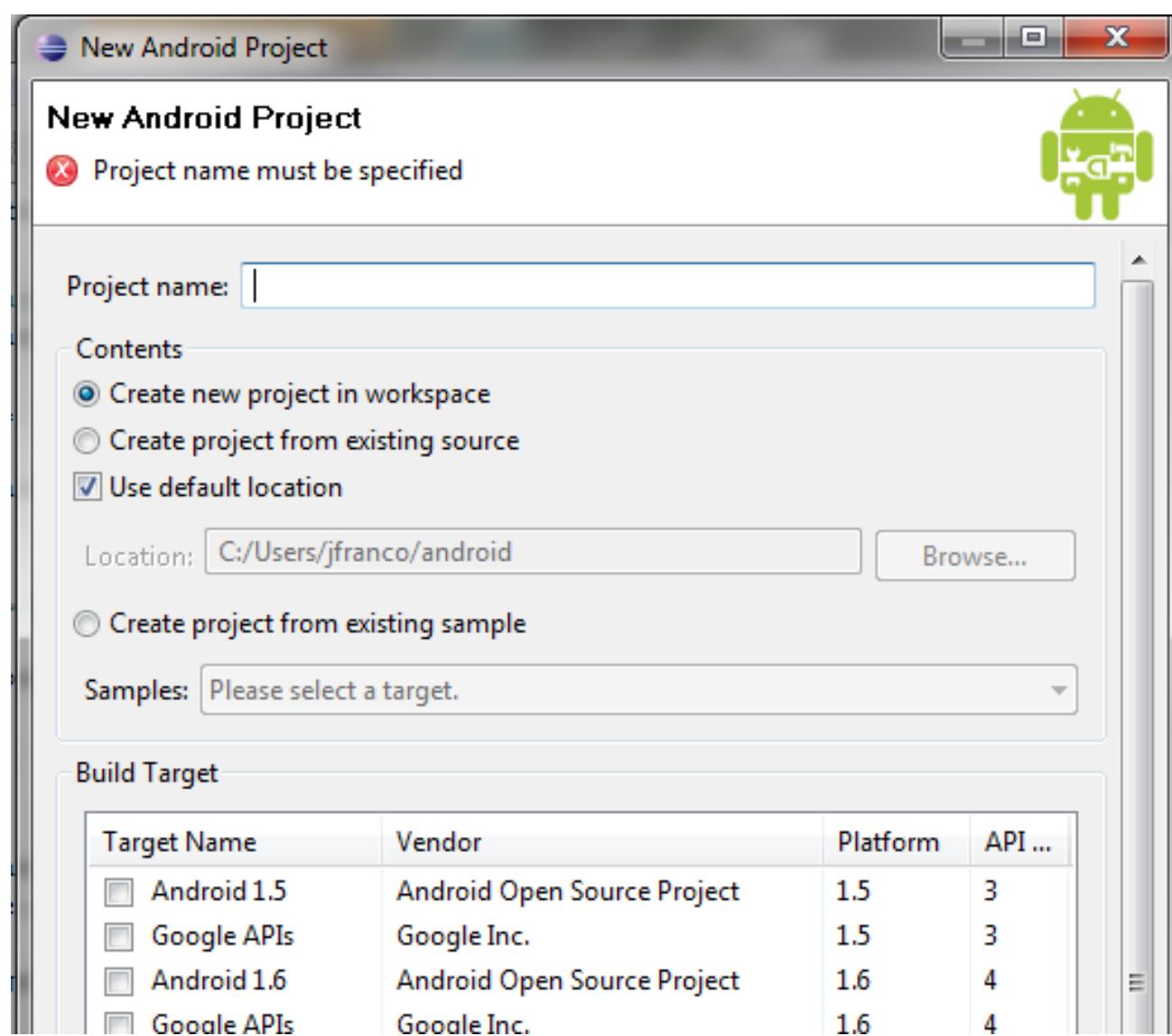
# Tipos de Proyecto

Estructura.



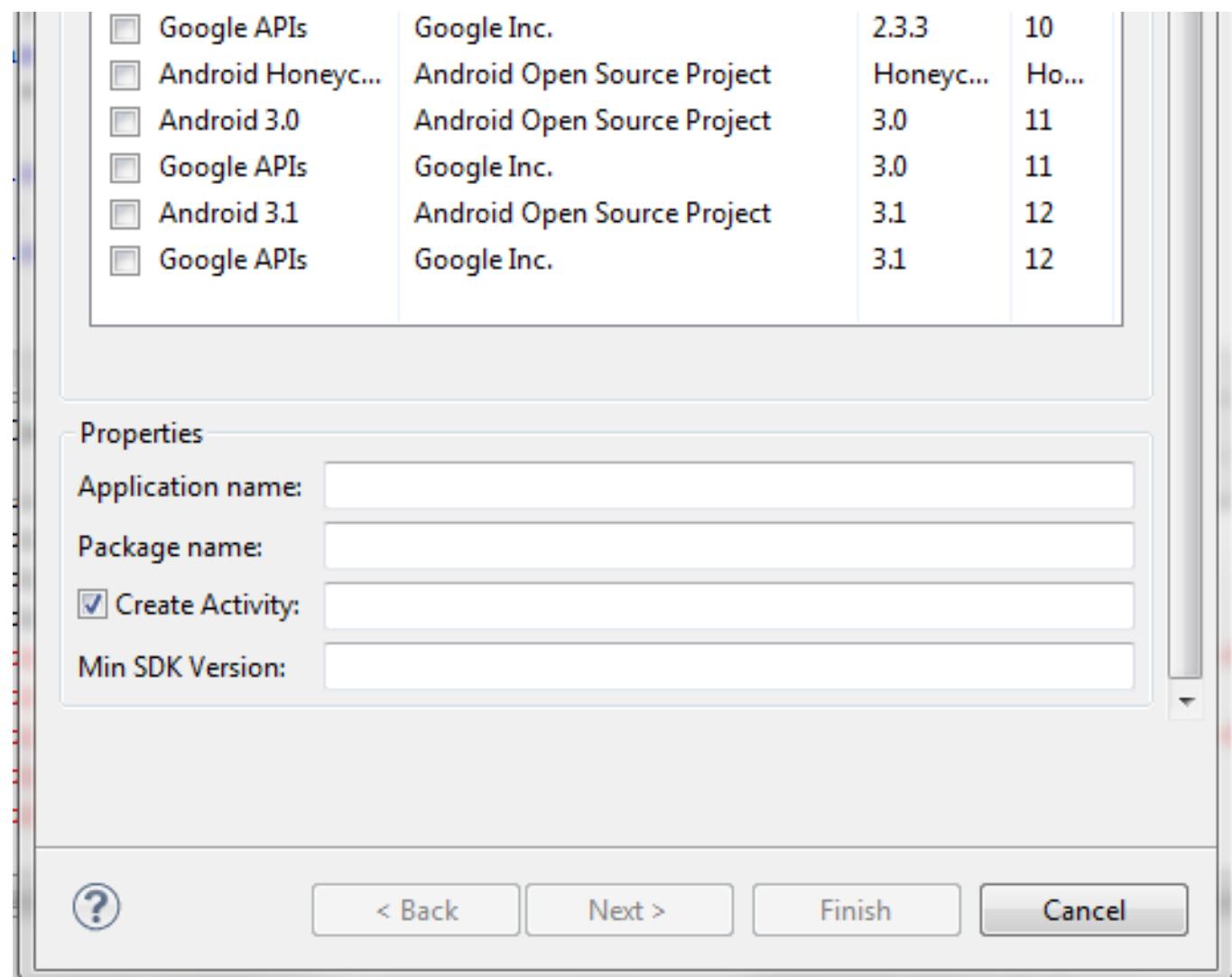
# Tipos de Proyecto

Crear el proyecto.



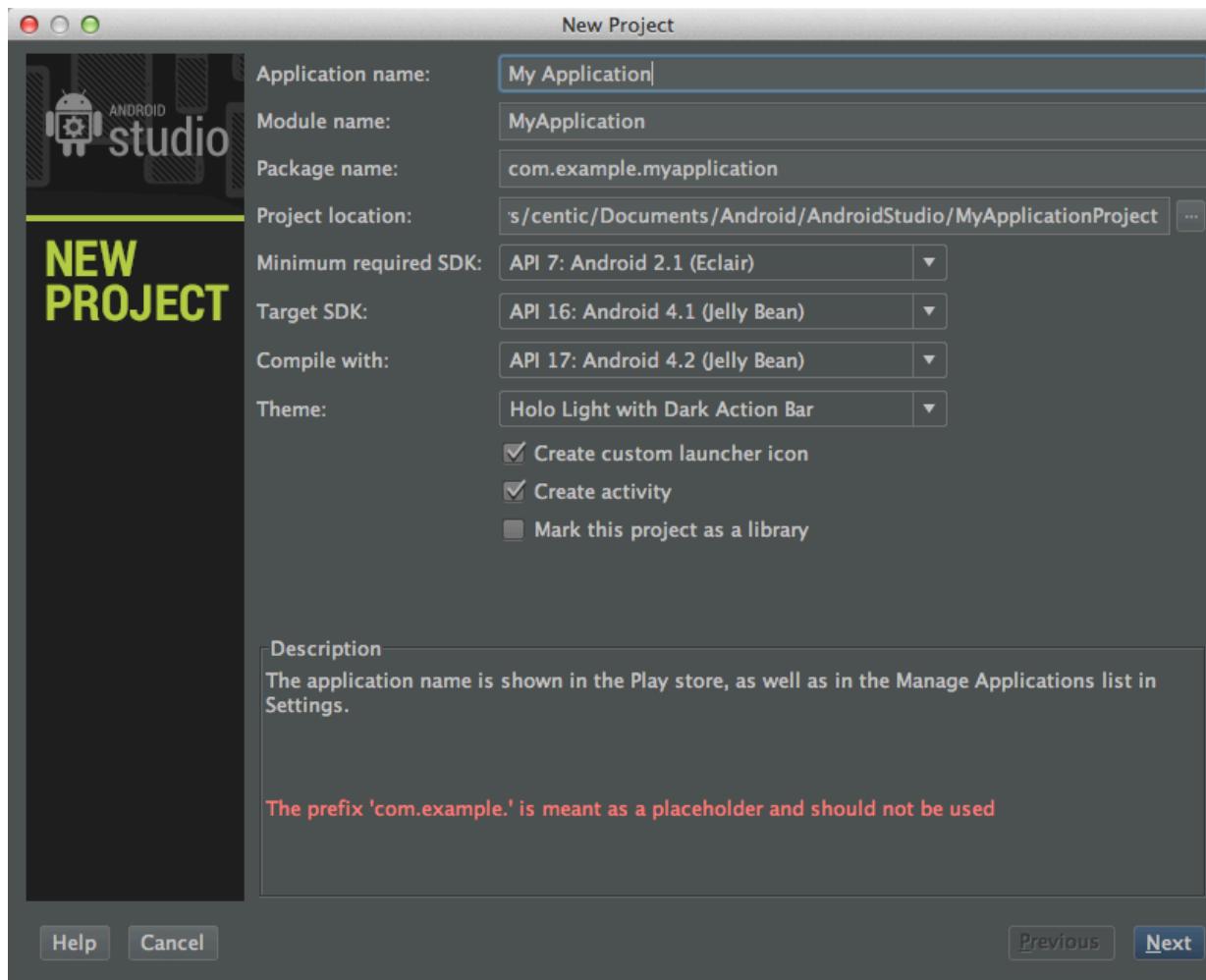
# Tipos de Proyecto

Crear el proyecto.



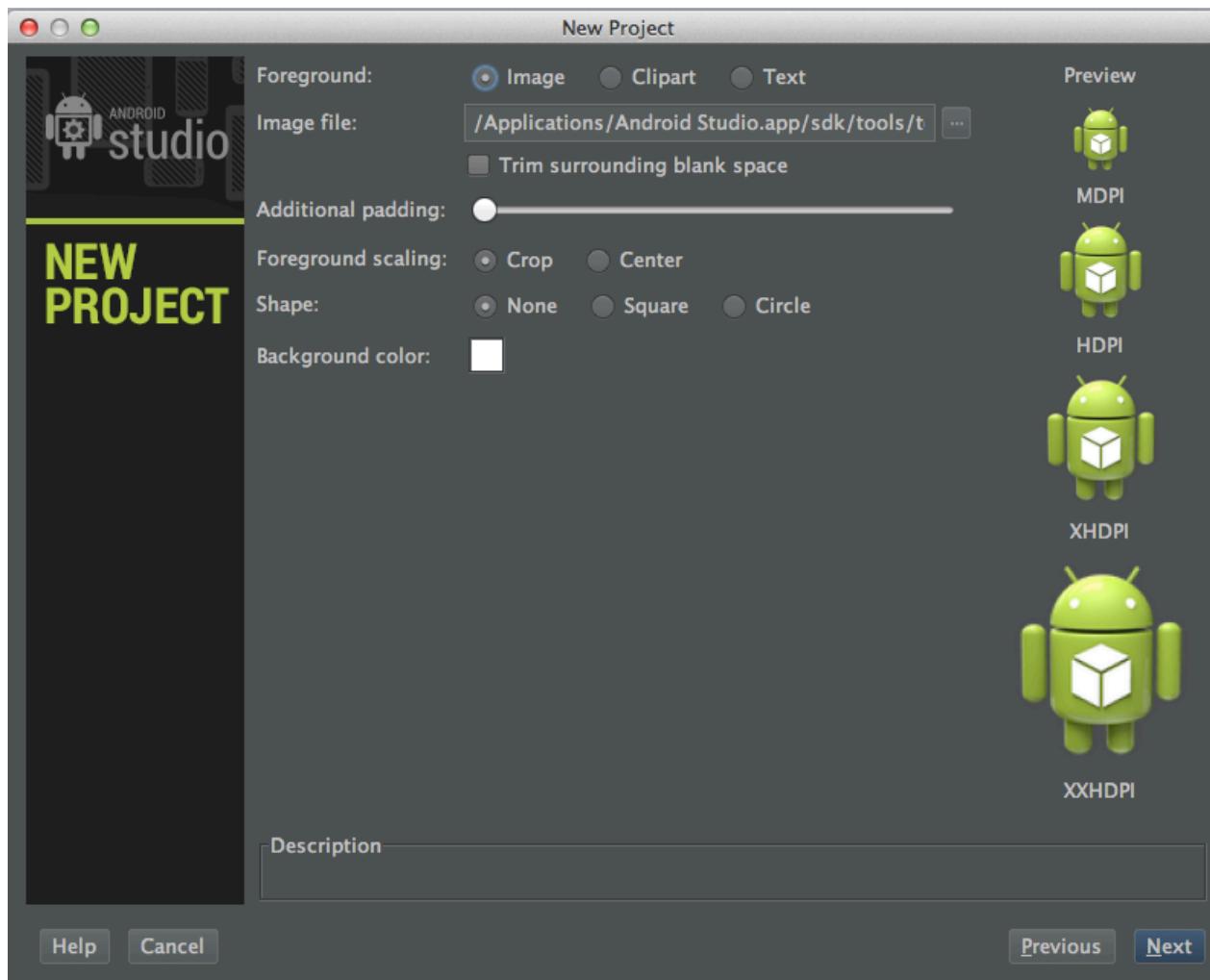
# Tipos de Proyecto

Crear el proyecto.



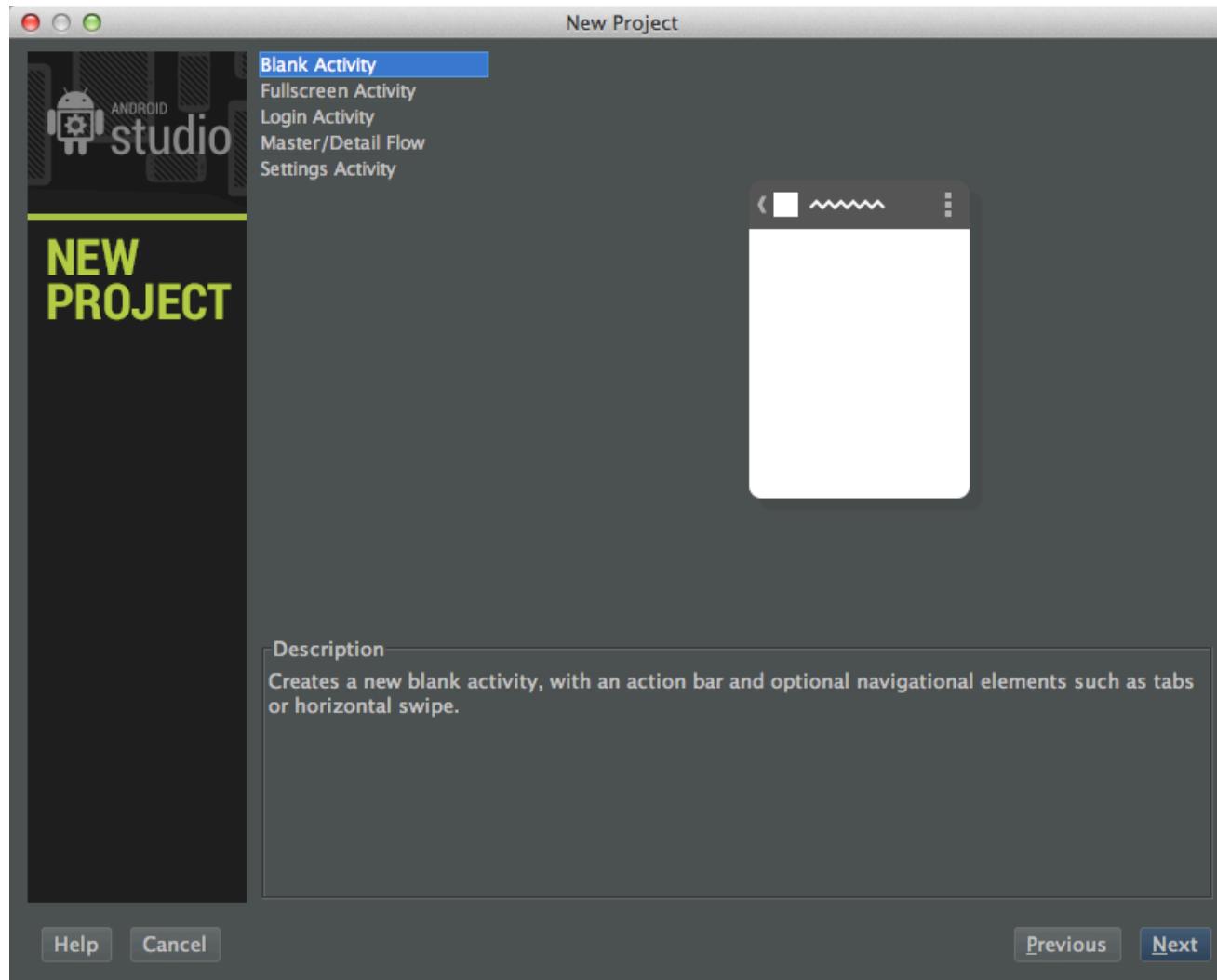
# Tipos de Proyecto

Crear el proyecto.



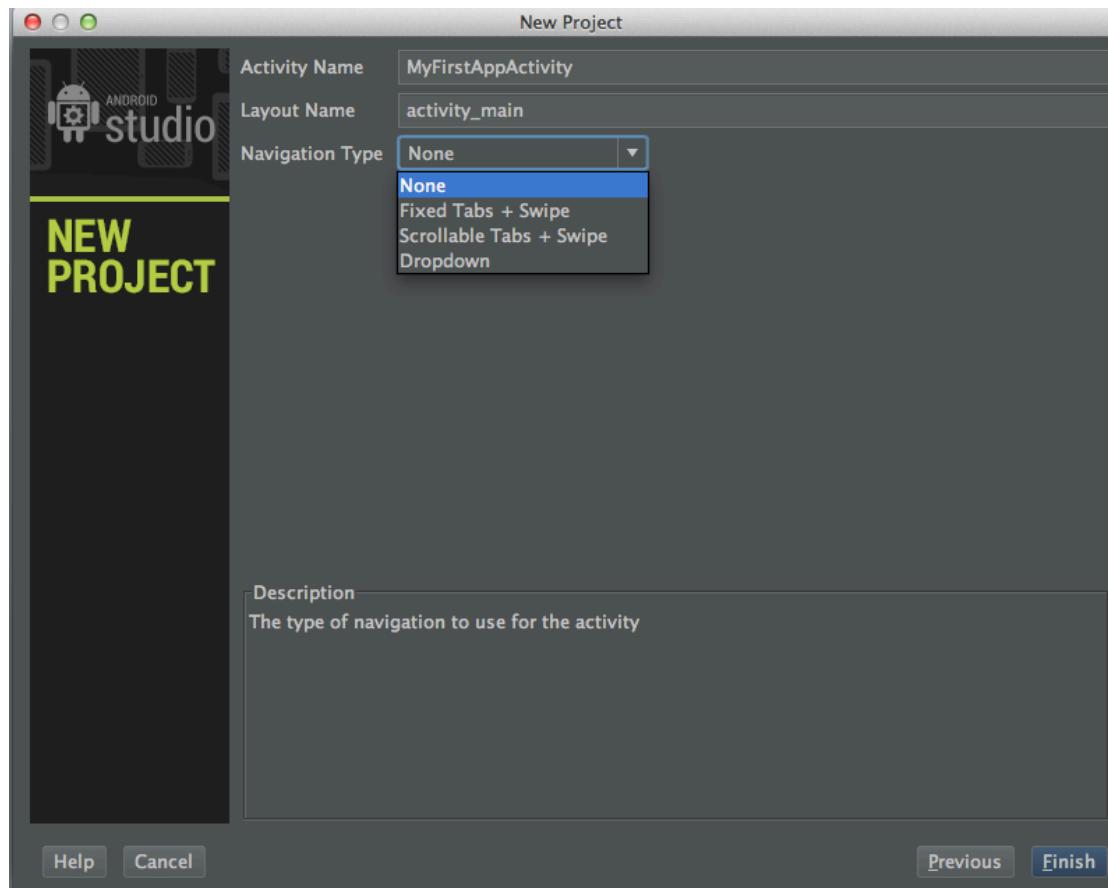
# Tipos de Proyecto

Crear el proyecto.



# Tipos de Proyecto

Crear el proyecto.



# Tipos de Proyecto

## Proyecto Biblioteca.

- No Genera un APK / Genera AAR
- Comparte la estructura de un proyecto Android
- No tiene un punto de lanzamiento
- Sirve para contener partes comunes a diferentes proyectos
- Muy útil cuando mantenemos distintas versiones de un mismo proyecto. Tipo Lite/ Premium
- Requiere ADT 9+

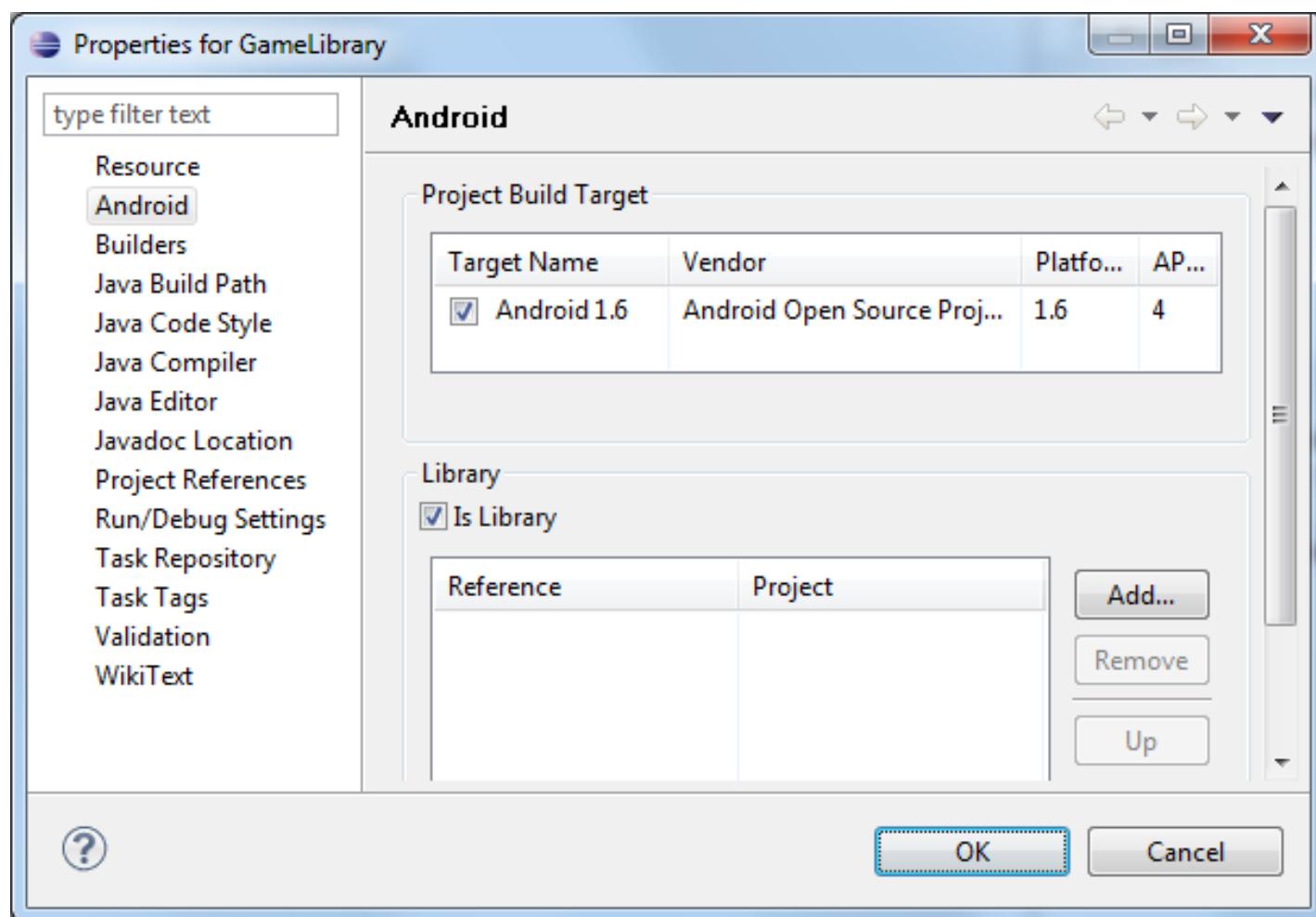
## Consideraciones.

- Conflictos entre recursos (Naming)
- Uso de recursos
- No puedes exportar a un JAR (Próximamente AAR)
- Puedes incluir JAR
- La aplicación que use la biblioteca debe declarar las librerías externas de la biblioteca. «<uses-library>»
- No puedes incluir el directorio assets/
- Debe estar dirigido a un SDK igual o inferior que el de las aplicaciones que lo contengan.
- Cada biblioteca crea su R.java



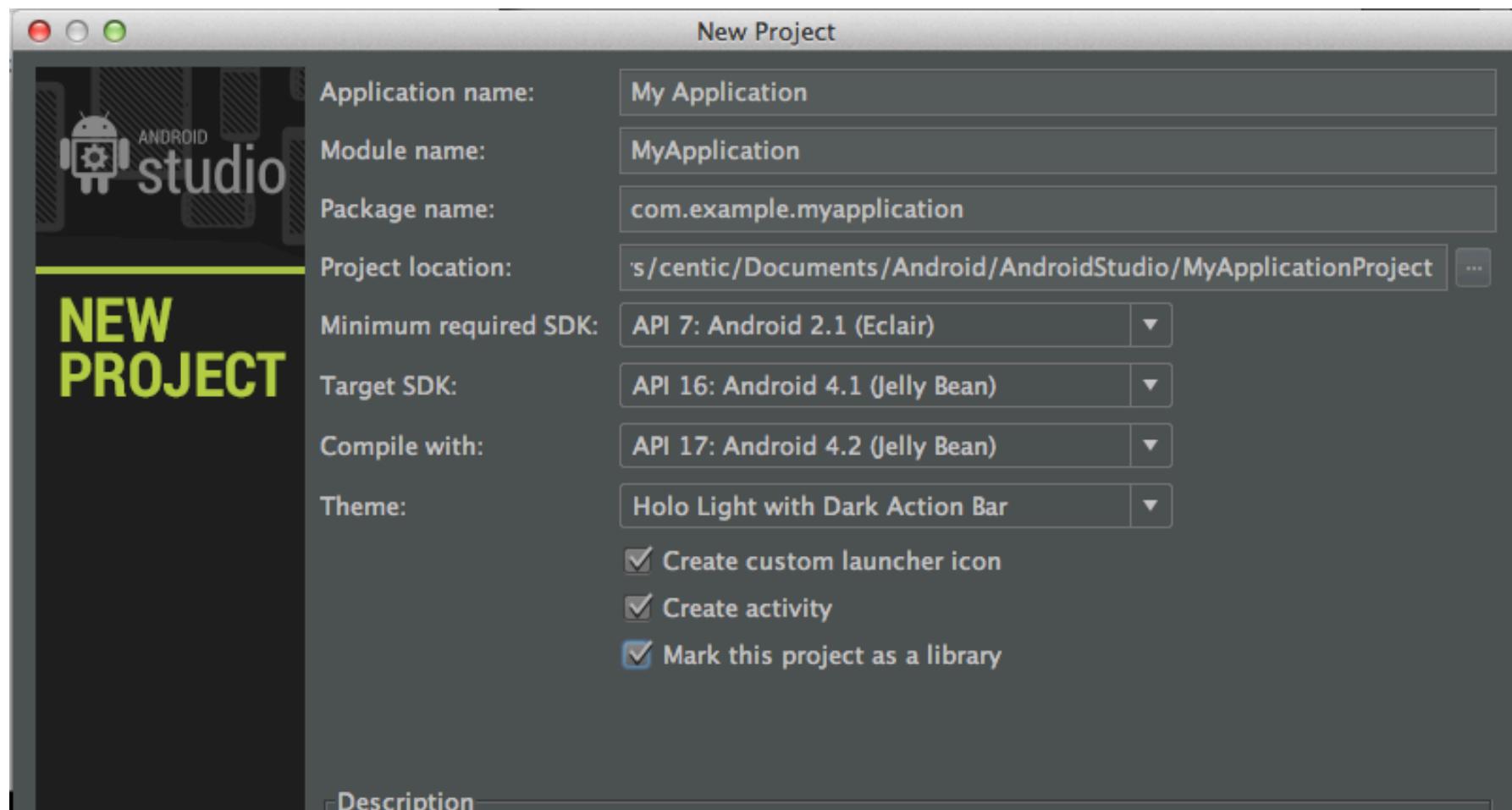
# Tipos de Proyecto

Crear una Biblioteca.



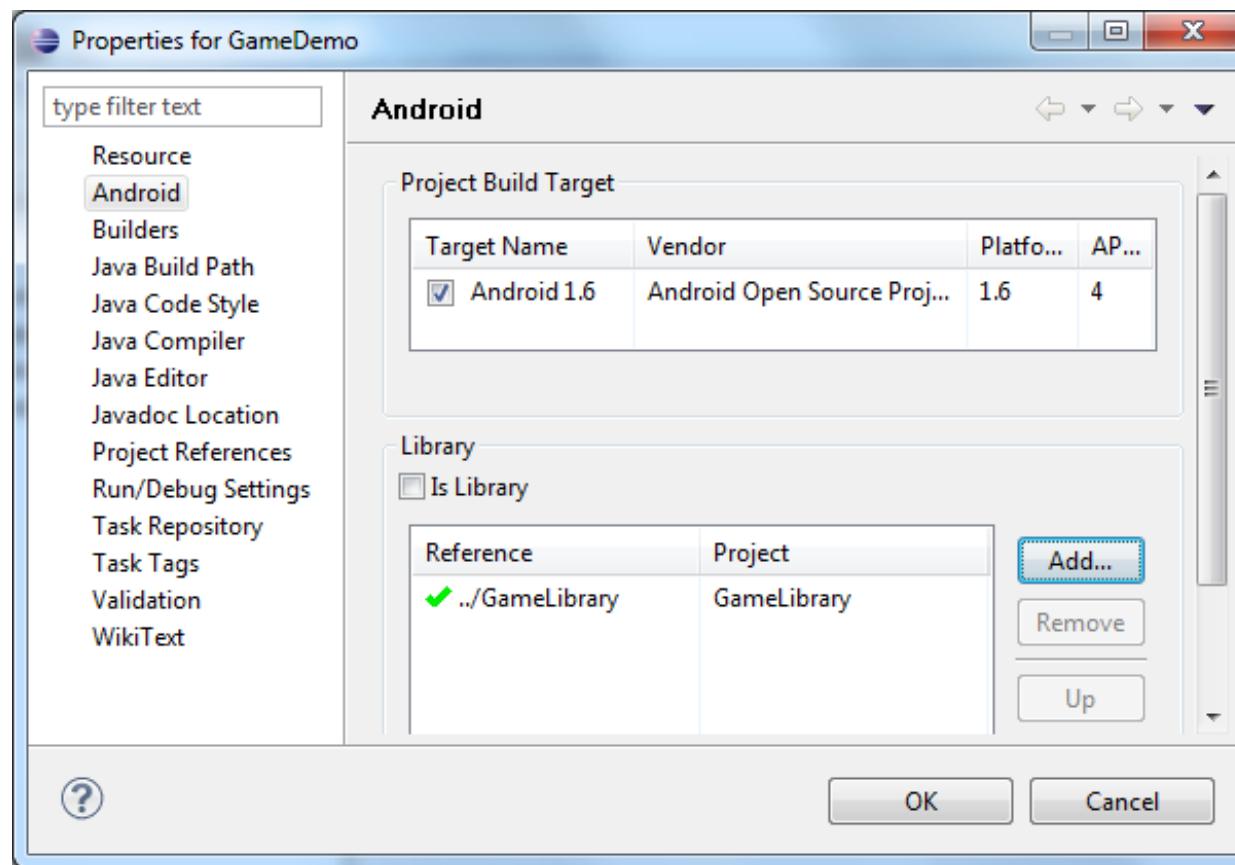
# Tipos de Proyecto

Crear una Biblioteca.



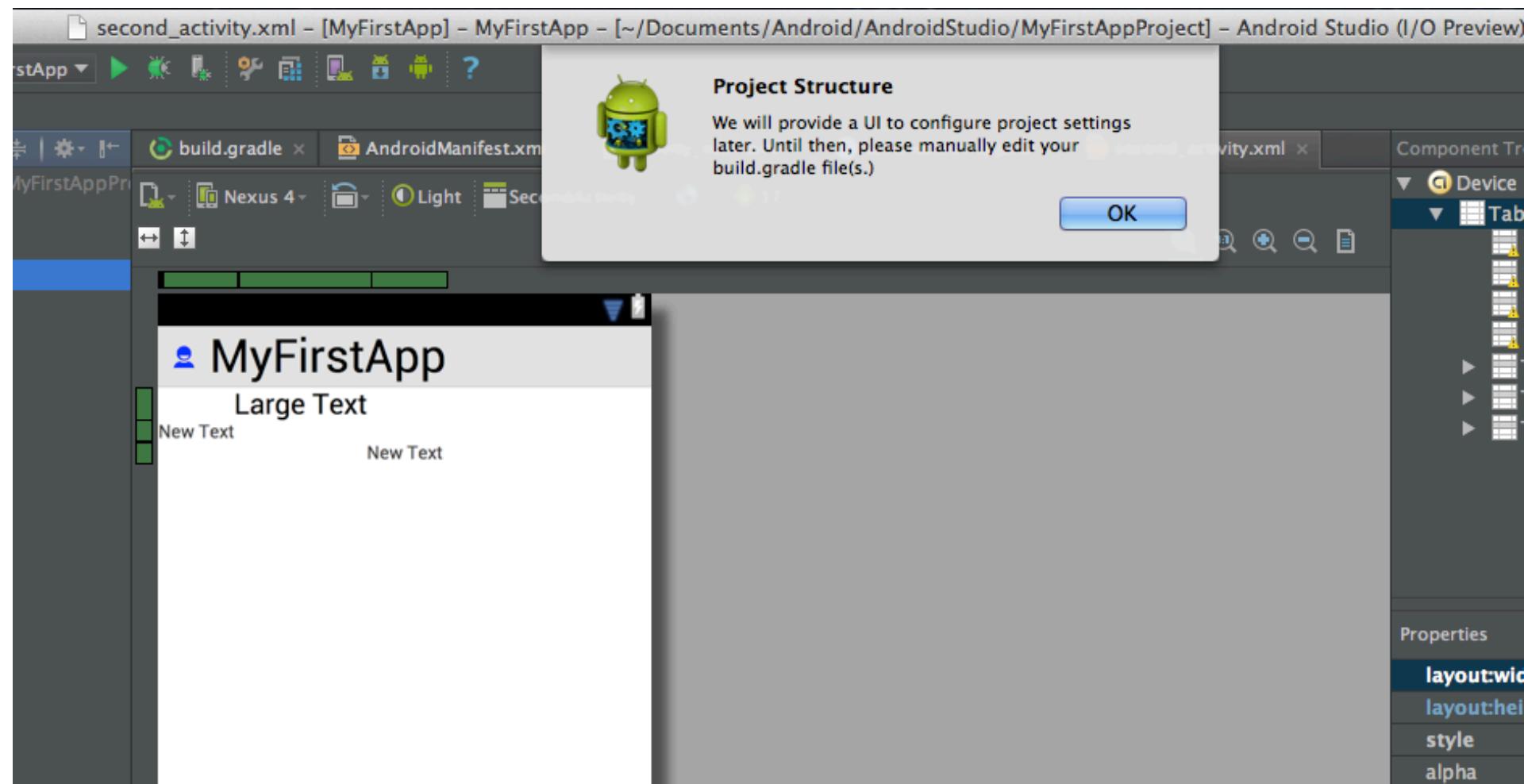
# Tipos de Proyecto

Referenciar una Biblioteca.



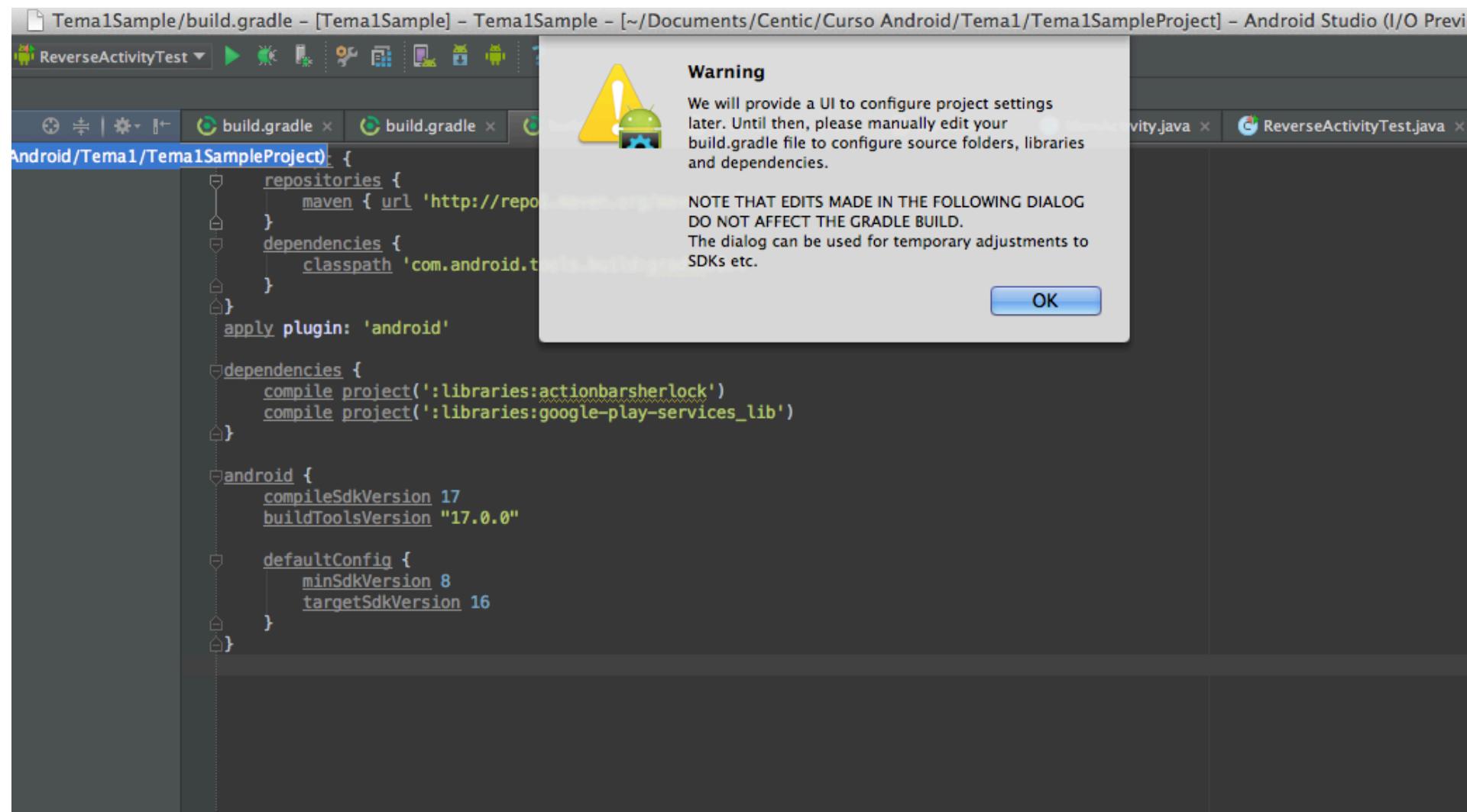
# Tipos de Proyecto

Referenciar una Biblioteca. Android Studio 0.1.1



# Tipos de Proyecto

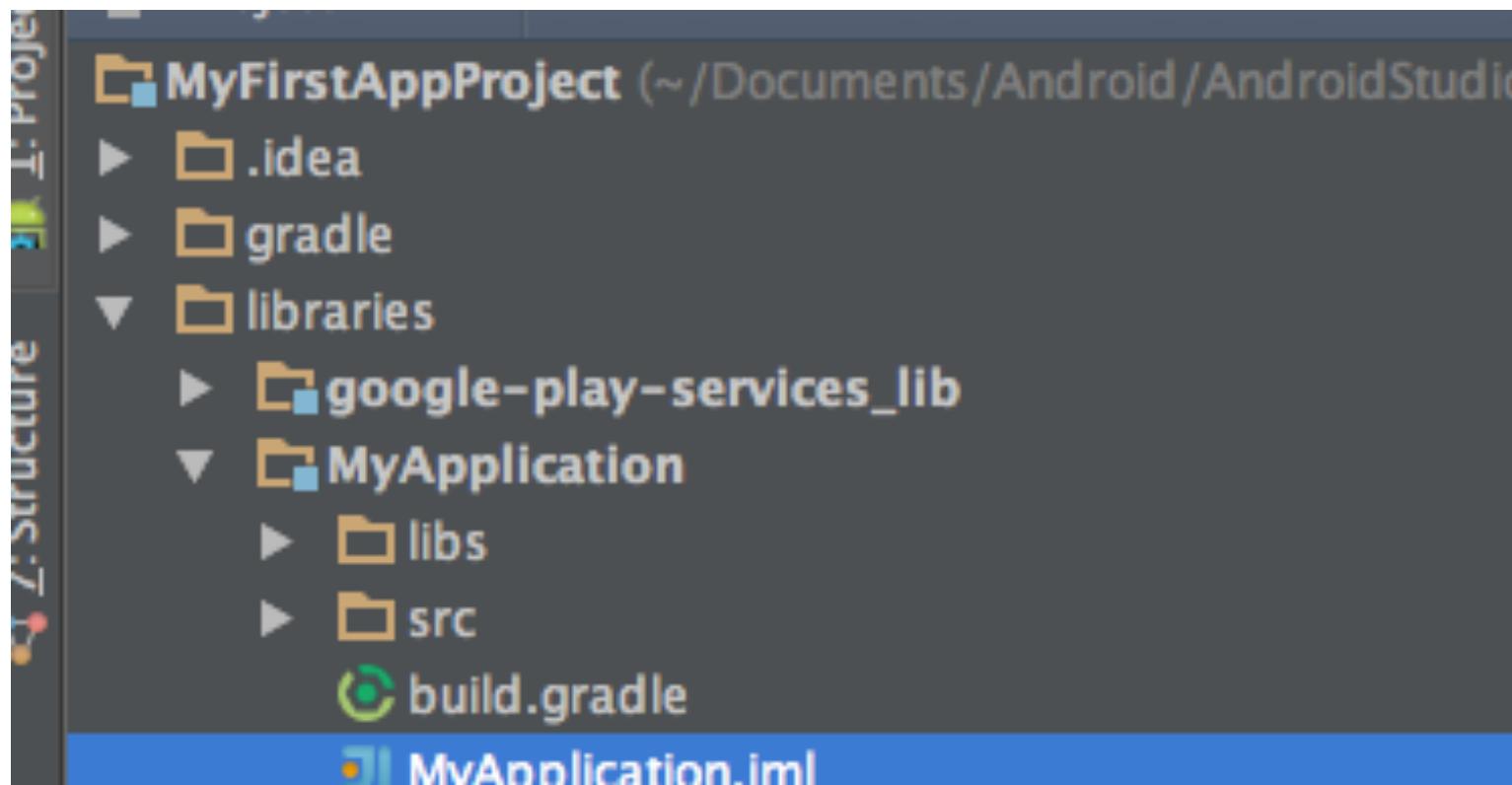
## Referenciar una Biblioteca. Android Studio 0.1.2



# Tipos de Proyecto

Referenciar una Biblioteca.

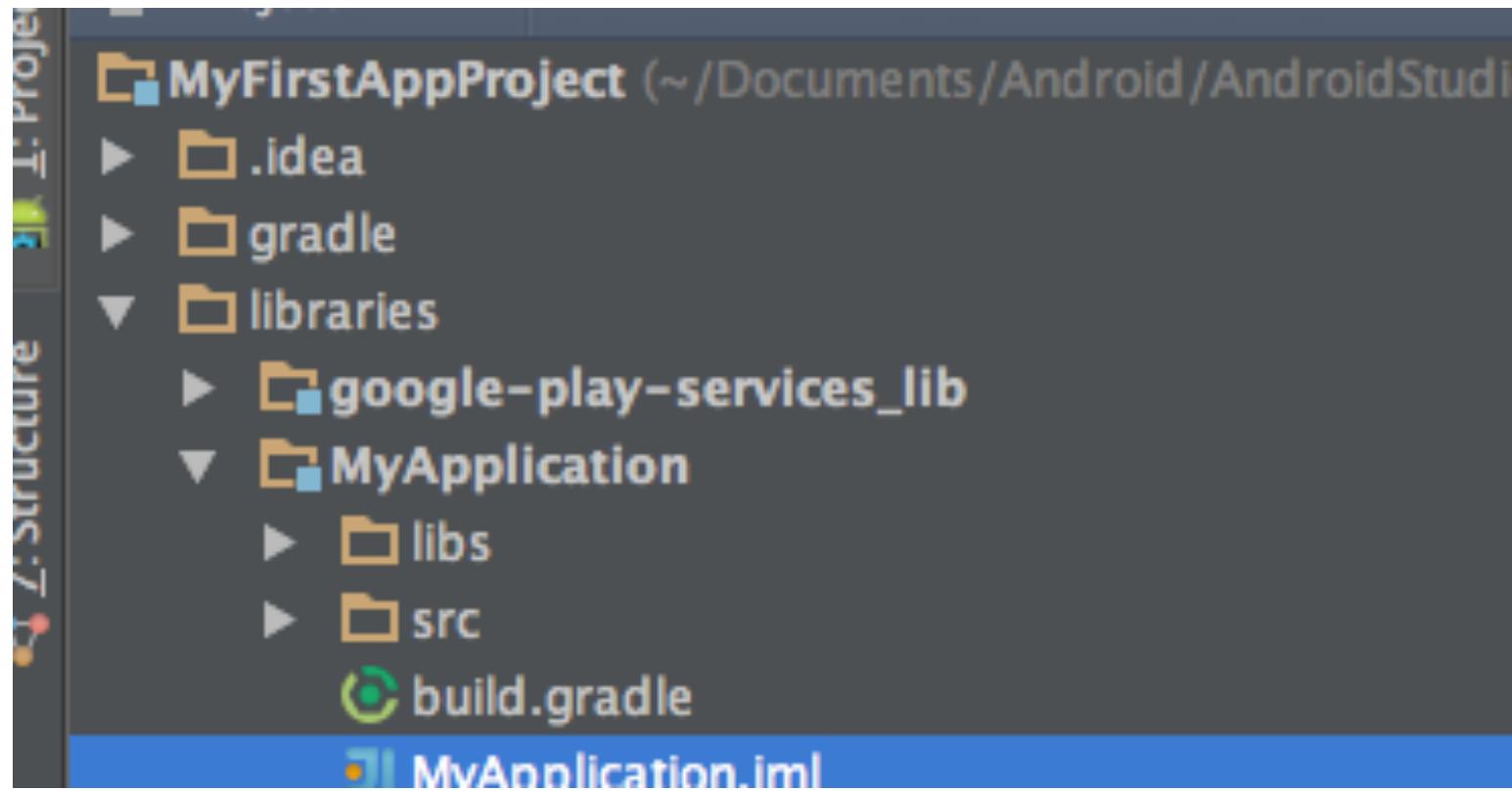
Crear el directorio “libraries” en la raíz del proyecto.



# Tipos de Proyecto

Referenciar una Biblioteca.

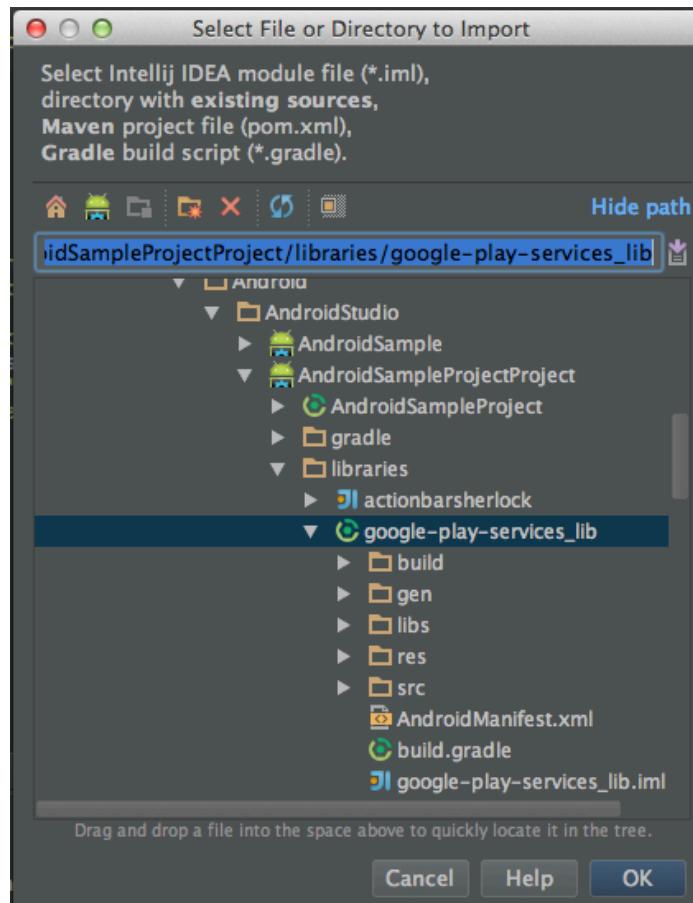
Copiar el proyecto “library” en dicha carpeta.



# Tipos de Proyecto

Referenciar una Biblioteca.

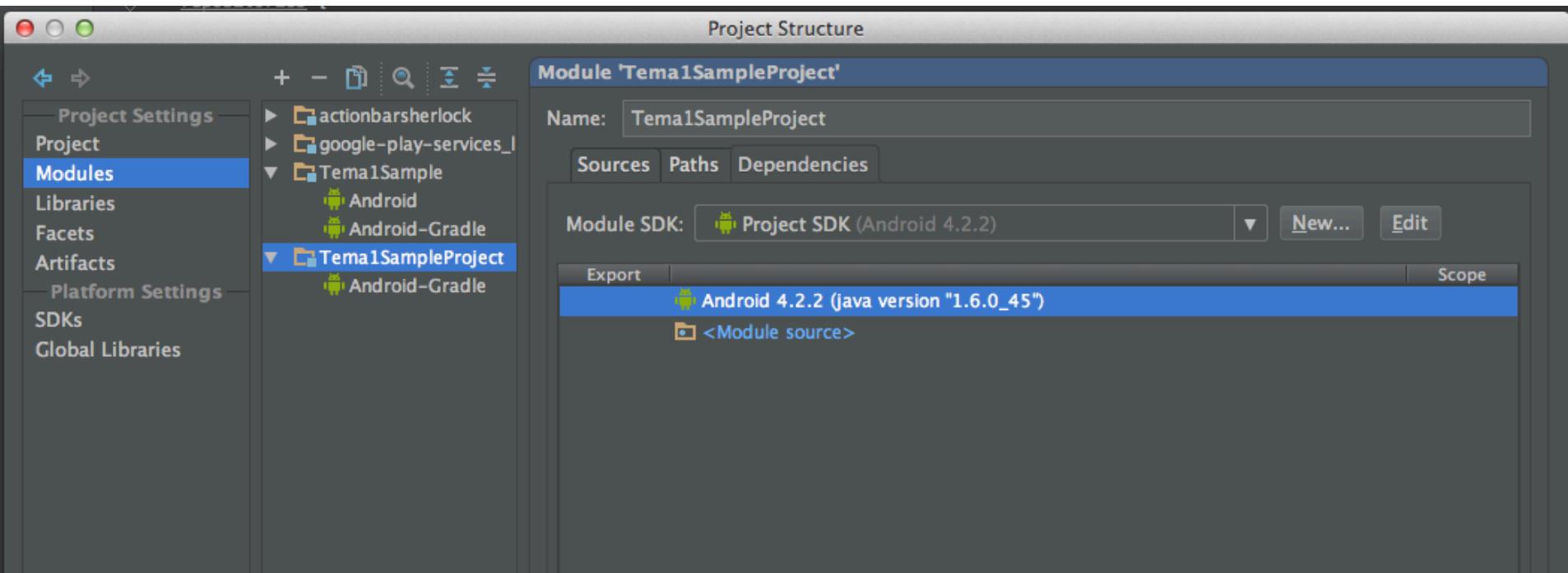
Importar como módulo File>Import Module. (Android Studio 0.1)



# Tipos de Proyecto

Referenciar una Biblioteca.

Importar como módulo File > Estructure (F4). (Android Studio 0.1.2)



# Tipos de Proyecto

Referenciar una Biblioteca.

Crear build.gradle para la biblioteca.

The screenshot shows the Android Studio interface with three tabs at the top: 'build.gradle', 'settings.gradle', and 'google-play-services\_lib.iml'. The 'build.gradle' tab is active and displays the following Groovy code:

```
apply plugin: 'android-library'

buildscript {
    repositories {
        mavenCentral()
    }

    dependencies {
        classpath 'com.android.tools.build:gradle:0.4'
    }
}

dependencies {
    compile files('libs/android-support-v4.jar')
    compile files('libs/google-play-services.jar')
}

android {
    compileSdkVersion 17
    buildToolsVersion '17.0.0'

    sourceSets {
        main {
            manifest.srcFile 'AndroidManifest.xml'
            java.srcDirs = ['src']
            resources.srcDirs = ['src']
            aidl.srcDirs = ['src']
            renderscript.srcDirs = ['src']
            res.srcDirs = ['res']
            assets.srcDirs = ['assets']
        }
    }
}
```

# Tipos de Proyecto

Referenciar una Biblioteca.

Modificar build.gradle para añadir la dependencia.

```
buildscript {
    repositories {
        maven { url 'http://repo1.maven.org/maven2' }
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:0.4'
    }
}
apply plugin: 'android'

dependencies {
    compile project(':libraries:actionbarsherlock')
    compile project(':libraries:google-play-services_lib')
}

android {
    compileSdkVersion 17
    buildToolsVersion "17.0.0"

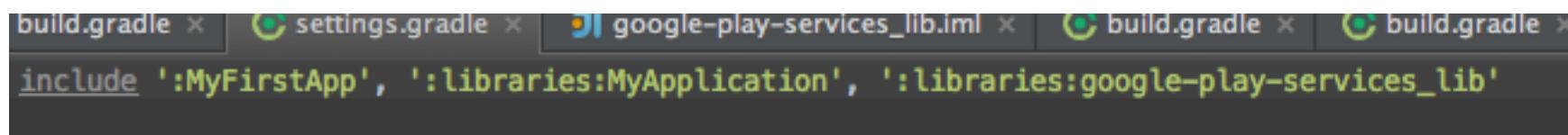
    defaultConfig {
        minSdkVersion 8
        targetSdkVersion 16
    }
}
```

# Tipos de Proyecto

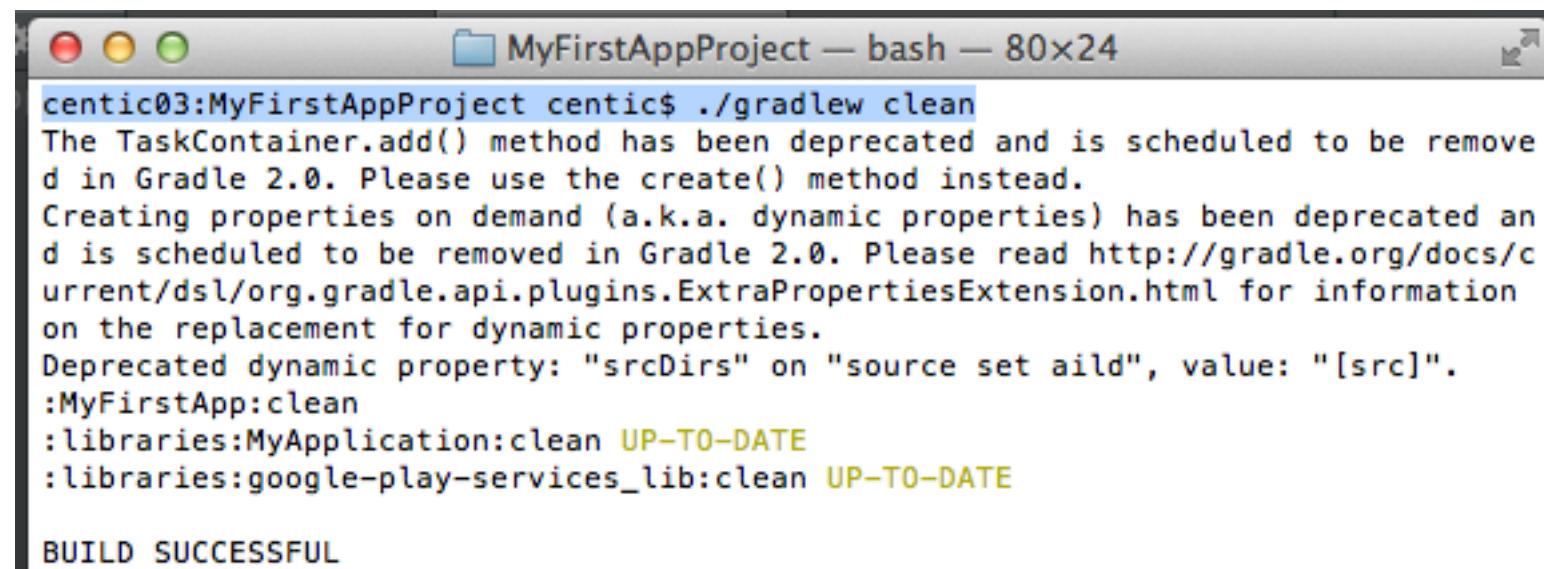
Referenciar una Biblioteca.

Añadir biblioteca en Settings.gradle.

Limpiar gradle y reiniciar IDE ?¿?¿ (Android Studio 0.1.2)



```
build.gradle x settings.gradle x google-play-services_lib.iml x build.gradle x build.gradle x
include ':MyFirstApp', ':libraries:MyApplication', ':libraries:google-play-services_lib'
```



```
centic03:MyFirstAppProject centic$ ./gradlew clean
The TaskContainer.add() method has been deprecated and is scheduled to be removed in Gradle 2.0. Please use the create() method instead.
Creating properties on demand (a.k.a. dynamic properties) has been deprecated and is scheduled to be removed in Gradle 2.0. Please read http://gradle.org/docs/current/dsl/org.gradle.api.plugins.ExtraPropertiesExtension.html for information on the replacement for dynamic properties.
Deprecated dynamic property: "srcDirs" on "source set aild", value: "[src]".
:MyFirstApp:clean
:libraries:MyApplication:clean UP-TO-DATE
:libraries:google-play-services_lib:clean UP-TO-DATE

BUILD SUCCESSFUL
```

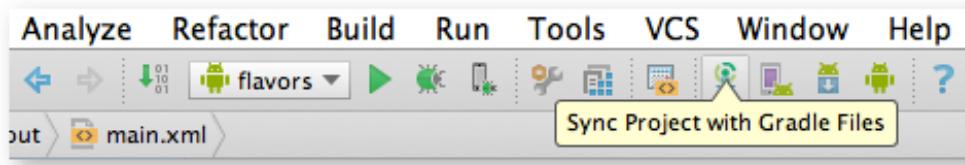
# Tipos de Proyecto

Referenciar una Biblioteca.

Añadir biblioteca en Settings.gradle.

Limpiar gradle y reiniciar IDE ?¿?¿ (Android Studio 0.1.3)

- There were a lot of fixes in the Gradle project import and build areas. There is now a "sync" button in the toolbar which will reimport the Gradle project state into your Android Studio project. Use this after editing your Gradle files, for example to add a library. In the future we will more automatically handle state syncing, but for now this is the simplest way to keep the IDE up to date with project structure changes made to the Gradle files.



# Tipos de Proyecto

## **Proyecto de Testing**

- No genera APK

### **Estructura.**

- La misma que un proyecto Android

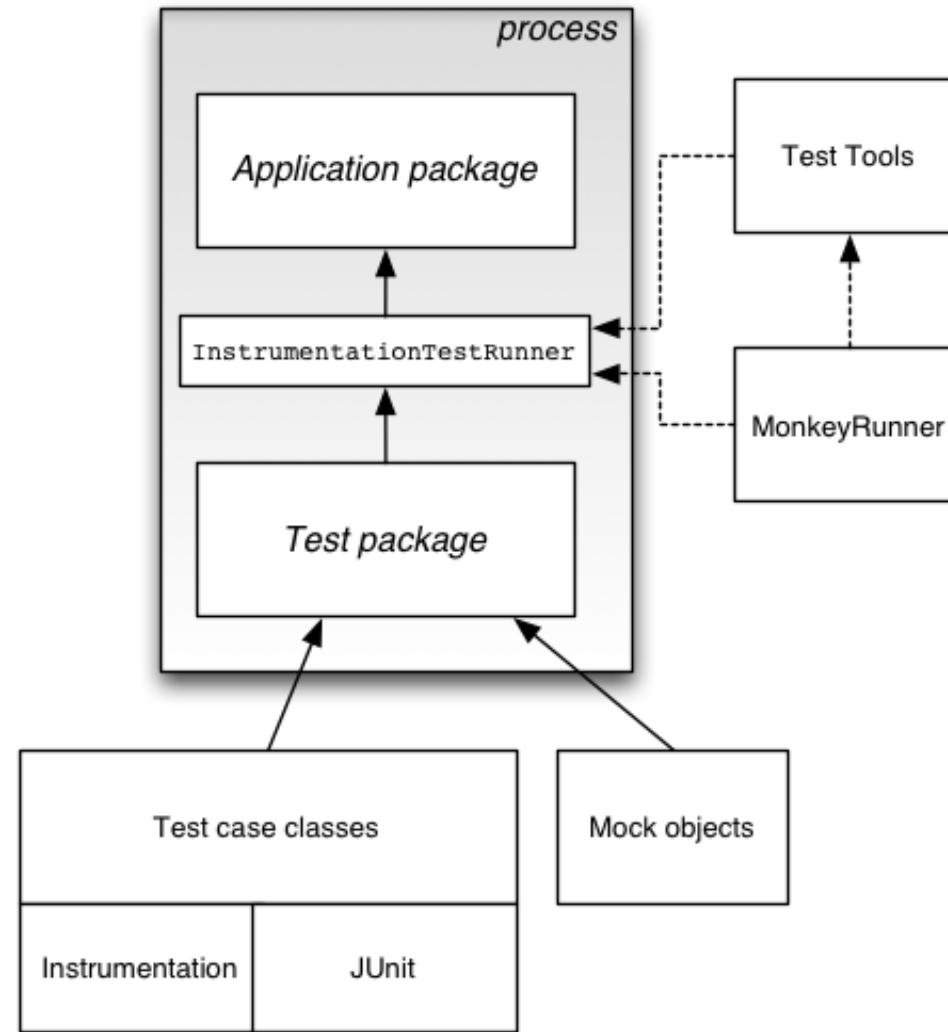
### **Consideraciones.**

- Está basado en Junit
- Proporciona clases de test específicas como `AndroidTestCase`
- Genera su propio espacio de nombres, basado en el de la aplicación que se quiere testear.
- El SDK trae la herramienta «monkeyrunner» para realizar los test de la GUI



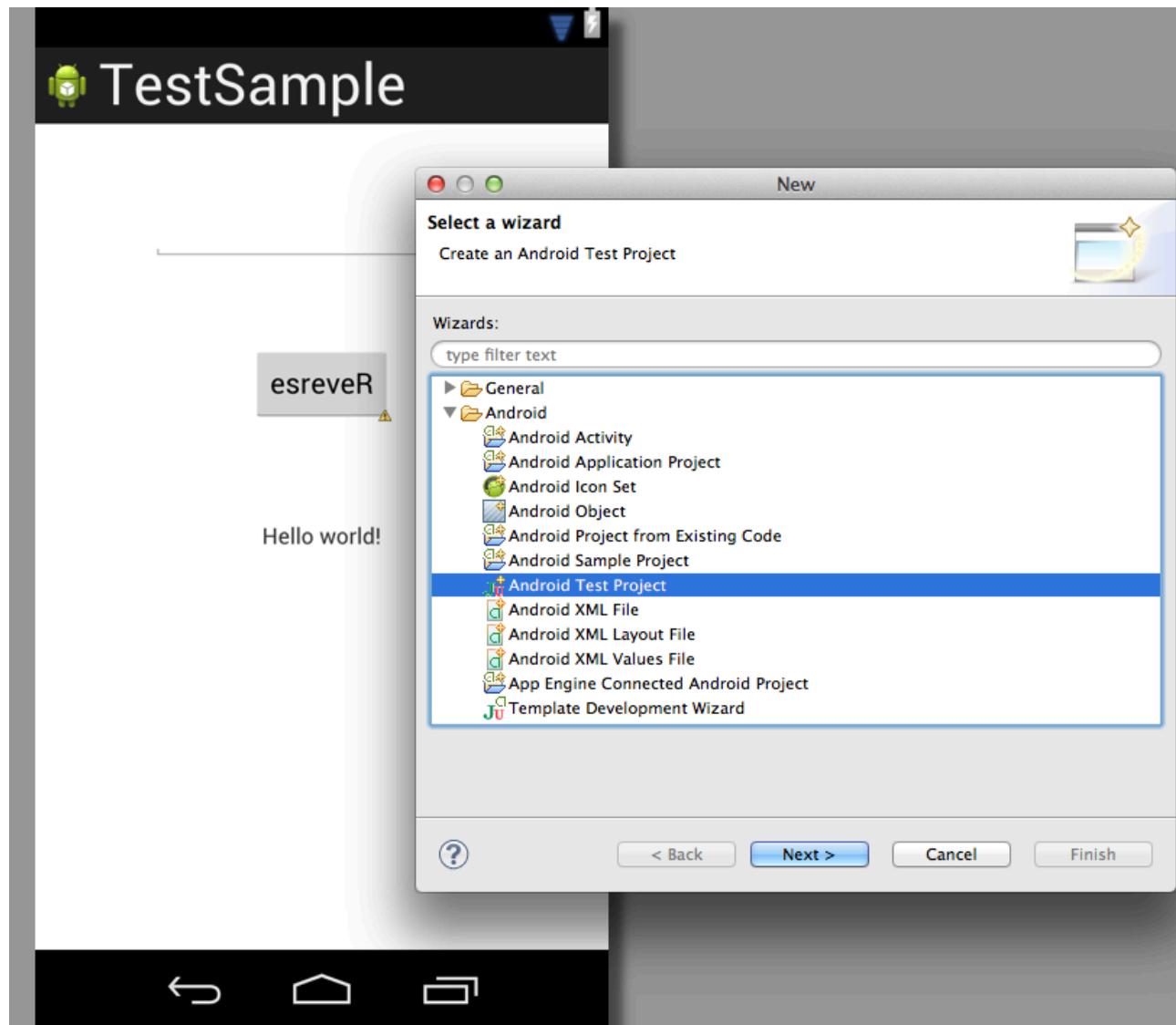
# Tipos de Proyecto

Proyecto para Test



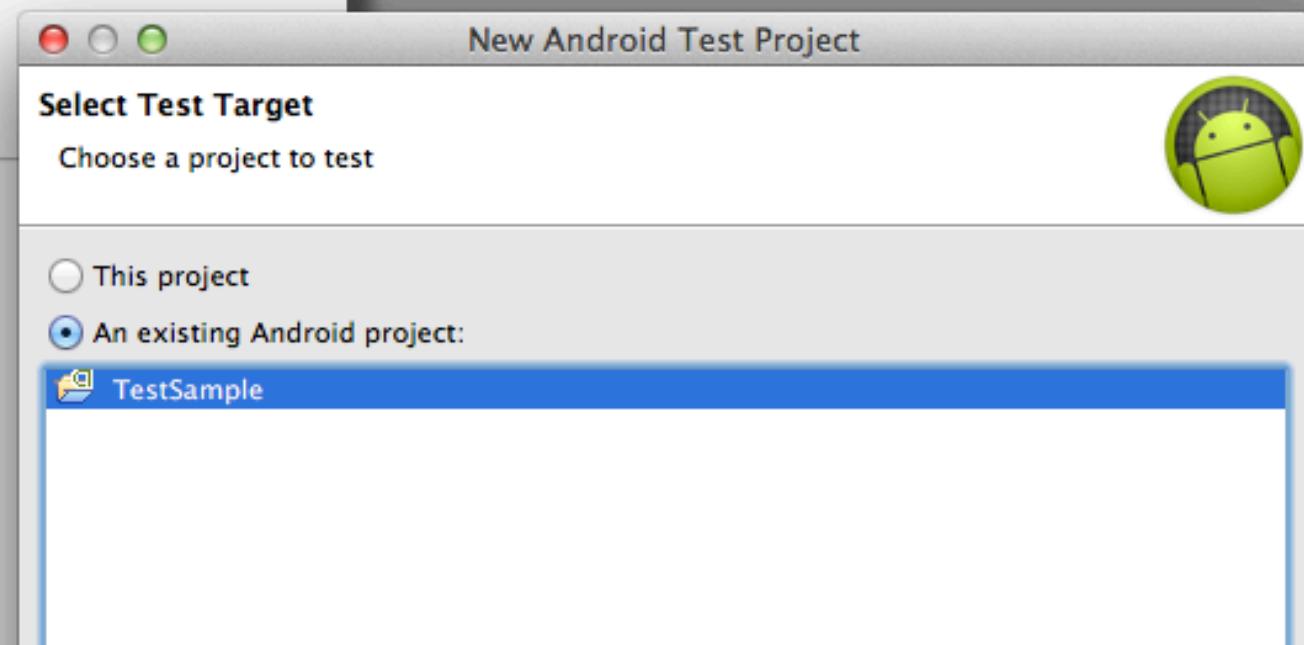
# Tipos de Proyecto

Crear un proyecto de Testing



# Tipos de Proyecto

Crear un proyecto de Testing



# Tipos de Proyecto

Crear un proyecto de Testing

```
<instrumentation  
    android:name="android.test.InstrumentationTestRunner"  
    android:targetPackage="com.centic.testsample" />  
  
<application  
    android:icon="@drawable/ic_launcher"  
    android:label="@string/app_name" >  
    <uses-library android:name="android.test.runner" />  
</application>  
  
</manifest>
```



# Tipos de Proyecto

## Crear un proyecto de Testing

```
// Start the main activity of the application under test
mActivity = getActivity();

// Get a handle to the Activity object's main UI widget, a Spinner
mSpinner = (Spinner)mActivity.findViewById(com.android.example.spinner.R.id.Spinner01);

// Set the Spinner to a known position
mActivity.setSpinnerPosition(TEST_STATE_DESTROY_POSITION);

// Stop the activity - The onDestroy() method should save the state of the Spinner
mActivity.finish();

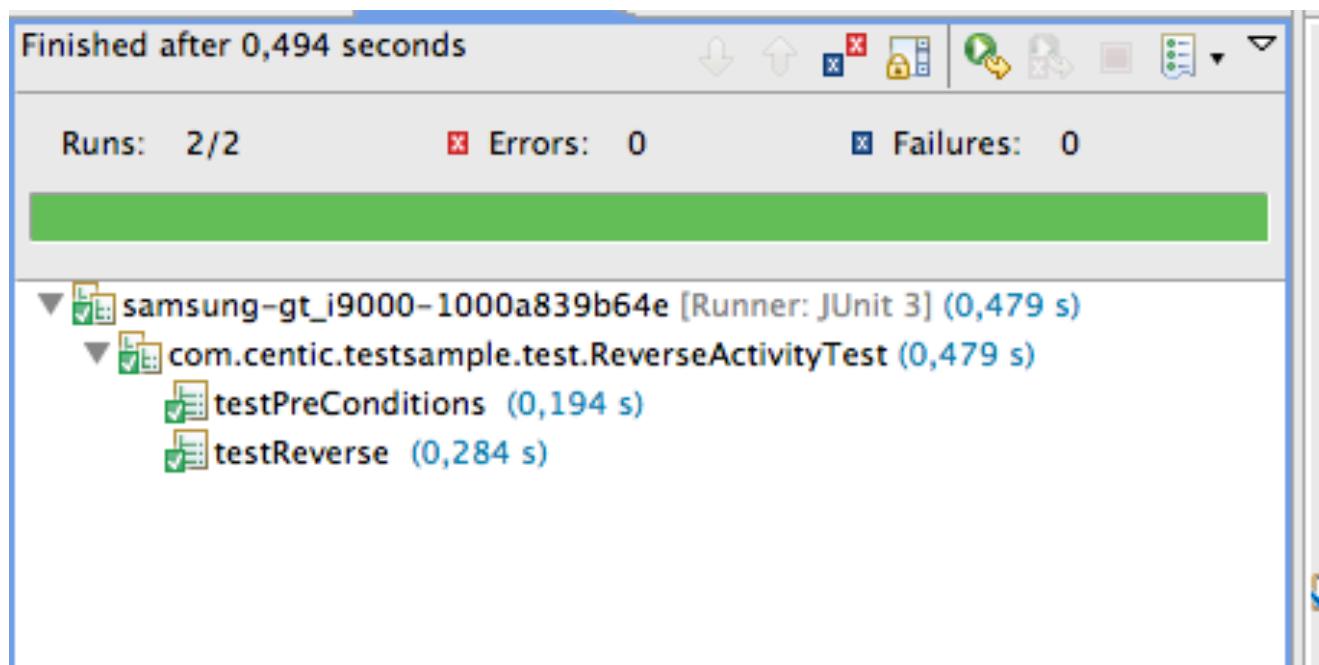
// Re-start the Activity - the onResume() method should restore the state of the Spinner
mActivity = getActivity();

// Get the Spinner's current position
int currentPosition = mActivity.getSpinnerPosition();

// Assert that the current position is the same as the starting position
assertEquals(TEST_STATE_DESTROY_POSITION, currentPosition);
```

# Tipos de Proyecto

Crear un proyecto de Testing



# Tipos de Proyecto

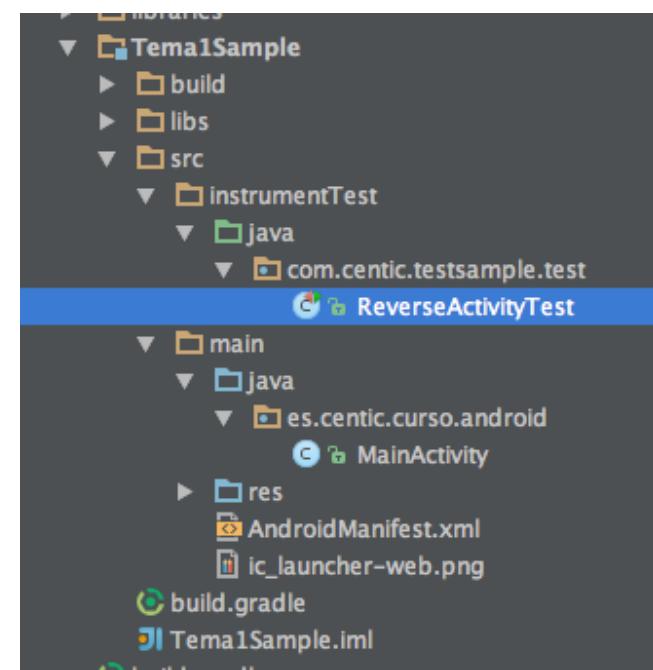
Android-Studio.

Crear el directorio instrumentTest

Crear el AndroidTestCase

Crear Test

Ejecutar Test



```
centic03:Tema1SampleProject centic$ ./gradlew connectedInstrumentTest
```

```
The TaskContainer.add() method has been deprecated and is scheduled to be removed in Gradle 2.0. Please use the create() method instead.
```

```
Creating properties on demand (a.k.a. dynamic properties) has been deprecated and is scheduled to be removed in Gradle 2.0. Please read http://gradle.org/docs/current/dsl/org.gradle.api.plugins.ExtraPropertiesExtension.html for information on the replacement for dynamic properties.
```

```
Deprecated dynamic property: "srcDirs" on "source set aild", value: "[src]".
```

```
Deprecated dynamic property "srcDirs" created in multiple locations.
```

# Tipos de Proyecto

Android-Studio.

## Test Summary

Resultados.

2  
tests

1  
failures

1.045s  
duration

50%  
successful

Failed tests

Packages

Classes

Class	Test
<u>ReverseActivityTest</u>	<u>testReverse</u>

# Tipos de Proyecto

Android-Studio.

Xavier Ducrohet:

[http://www.youtube.com/watch?  
v=LCJAgPkpmR0&list=PLWz5rJ2EKKc9WGUwq2gQ-  
coU3fSyexgOx](http://www.youtube.com/watch?v=LCJAgPkpmR0&list=PLWz5rJ2EKKc9WGUwq2gQ-coU3fSyexgOx)



Xavier Ducrohet

May 20

★ To be honest, I don't think we've check this part after the Gradle integration. We'll look into it.

In the meantime you can run your tests from the command line. `./gradlew connectedCheck` will run you tests on all connected devices.



# Tipos de Proyecto

## **UIAutomator.**

Automatizar pruebas sobre la UI de una aplicación en un dispositivo.

Ahorrar tiempo de ejecutar de forma manual.

Elimina el error humano.

Es un API Java simular la interactuación entre la aplicación y un usuario real.

Solo disponible para SDK  $\geq 16$  y ADT  $\geq 21$



# Tipos de Proyecto

## **UIAutomator.**

Instalar la aplicación a testar en el dispositivo.

Simular las acciones del usuario. (API Java)

Compilar Test en un JAR.

Ejecutar test

Comprobar salida del test, podemos redireccionar su salida a un fichero de logs.



# Tipos de Proyecto

**UIAutomator.** Proceso para crear un test

Analizar la UI de nuestra aplicación. Uiatumatorviewer.

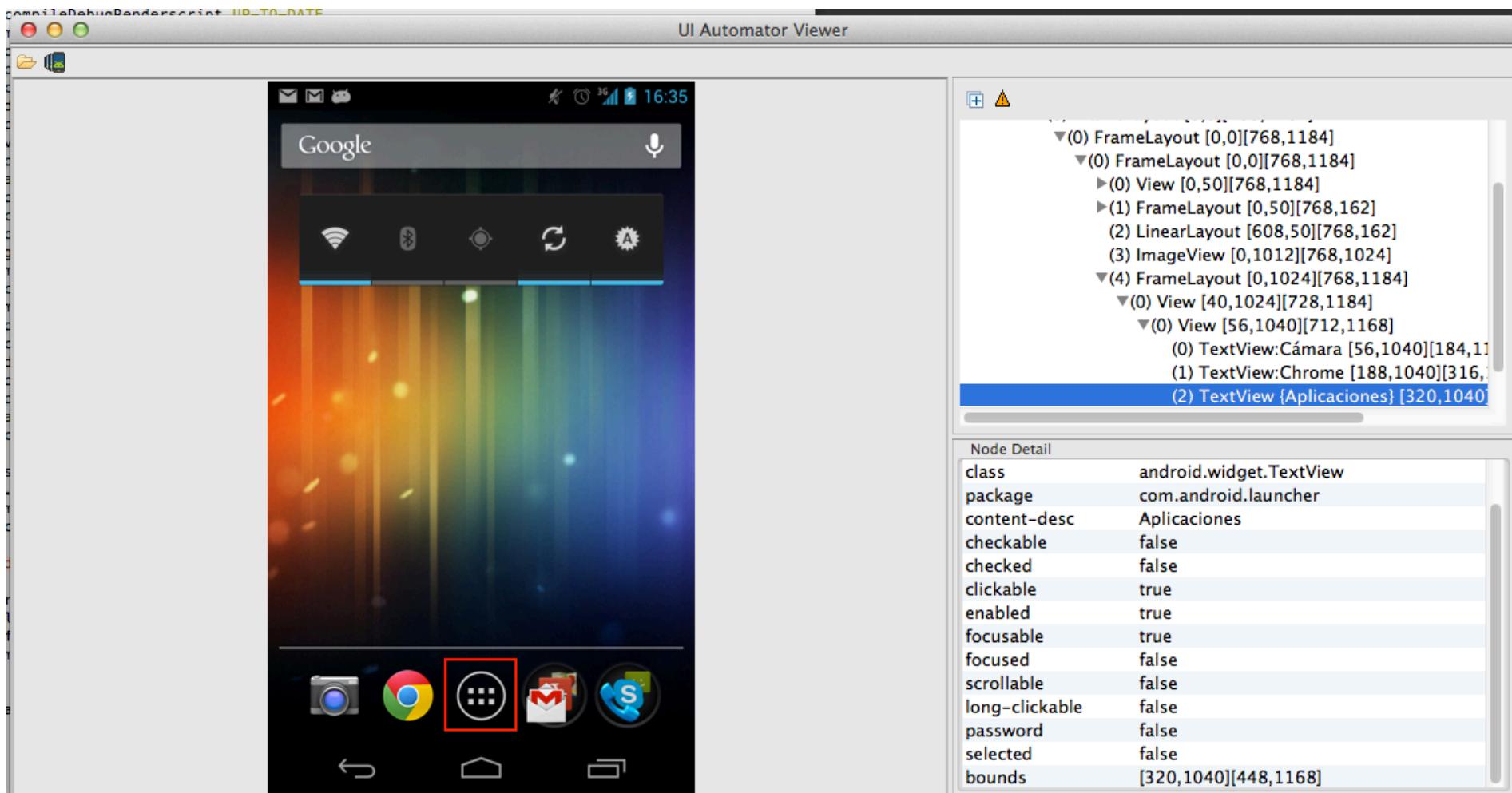
- Cargar apk en el dispositivo
- Conectar dispositivo al ADB
- Ejecutar uiautomatorviewer
- Anotar los elementos graficos que queremos utilizar.
- Asegurar que la aplicación es accesible
  - contentDescription
  - Hit
  - text

```
$ uiautomatorviewer
```



# Tipos de Proyecto

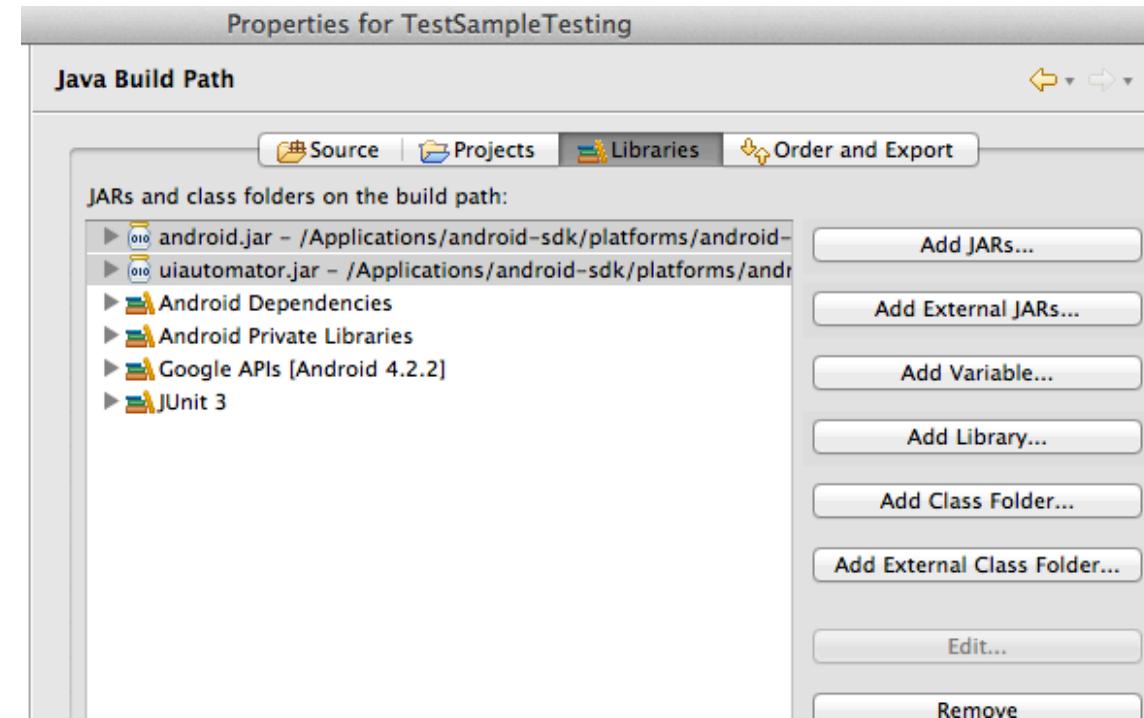
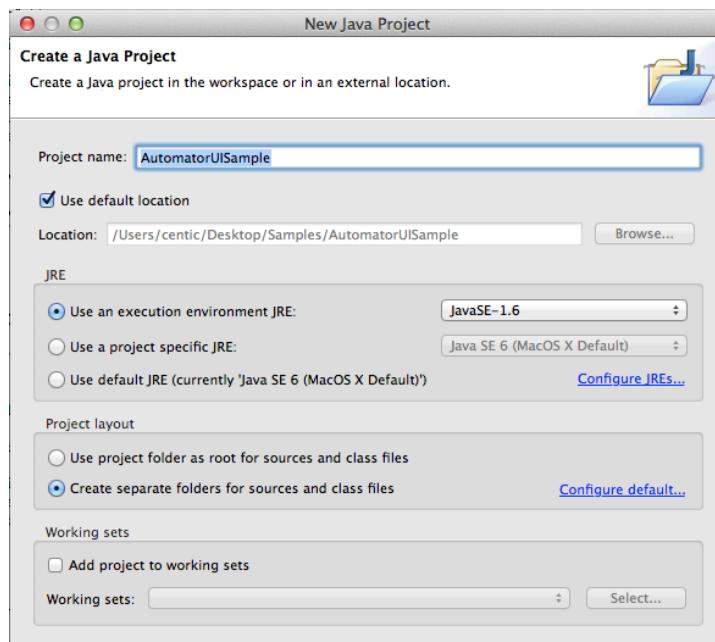
**UIAutomator.** Proceso para crear un test



# Tipos de Proyecto

**UIAutomator.** Configurar el entorno de desarrollo.

- Crear un proyecto Java.
- Añadir JUnit3
- Añadir uiautomator.jar y android.jar(>=16)



# Tipos de Proyecto

## UIAutomator. Codificar Test.

```
// Simulate a short press on the HOME button.  
getUiDevice().pressHome();  
  
// We're now in the home screen. Next, we want to simulate  
// a user bringing up the All Apps screen.  
// If you use the uiautomatorviewer tool to capture a snapshot  
// of the Home screen, notice that the All Apps button's  
// content-description property has the value "Apps". We can  
// use this property to create a UiSelector to find the button.  
UiObject allAppsButton = new UiObject(new UiSelector()  
    .description("Aplicaciones"));  
  
// Simulate a click to bring up the All Apps screen.  
allAppsButton.clickAndWaitForNewWindow();  
  
// In the All Apps screen, the Settings app is located in  
// the Apps tab. To simulate the user bringing up the Apps tab,  
// we create a UiSelector to find a tab with the text  
// label "Apps".  
UiObject appsTab = new UiObject(new UiSelector()  
    .text("Aplicaciones"));  
  
// Simulate a click to enter the Apps tab.  
appsTab.click();
```

# Tipos de Proyecto

## UIAutomator. Codificar Test.

```
// Next, in the apps tabs, we can simulate a user swiping until  
// they come to the Settings app icon. Since the container view  
// is scrollable, we can use a UiScrollable object.  
UiScrollable appViews = new UiScrollable(new UiSelector()  
    .scrollable(true));  
  
// Set the swiping mode to horizontal (the default is vertical)  
//appViews.setAsHorizontalList();  
  
// Create a UiSelector to find the Settings app and simulate  
// a user click to launch the app.  
UiObject settingsApp = appViews.getChildByText(new UiSelector()  
    .className(android.widget.TextView.class.getName()),  
    "Ajustes");  
settingsApp.clickAndwaitForNewWindow();  
  
// Validate that the package name is the expected one  
UiObject settingsValidation = new UiObject(new UiSelector()  
    .packageName("com.android.settings"));  
assertTrue("Unable to detect Settings",  
    settingsValidation.exists());
```



# Tipos de Proyecto

## UIAutomator. Compilar Test.

```
android create uitest-project -n <name> -t X -p <path>
```

X => android list targets

ant build

```
centic03:AutomatorUISample centic$  
centic03:AutomatorUISample centic$  
centic03:AutomatorUISample centic$ ant build  
Buildfile: /Users/centic/Desktop/Samples/AutomatorUISample/build.xml  
  
-check-env:  
  
centic03:AutomatorUISample centic$ android create uitest-project -n AutomatorUISample -t 26 -p .  
Updated file ./build.xml  
centic03:AutomatorUISample centic$
```

# Tipos de Proyecto

UIAutomator. Ejecutar Test.

```
BUILD SUCCESSFUL
centic03:AutomatorUISample centic$ adb push bin/AutomatorUISample.jar /data/local/tmp
60 KB/s (1542 bytes in 0.024s)
centic03:AutomatorUISample centic$ adb shell uiautomator runtest AutomatorUISample.jar -c es.
INSTRUMENTATION_STATUS: current=1
INSTRUMENTATION_STATUS: id=UiAutomatorTestRunner
INSTRUMENTATION_STATUS: class=es.centic.curso.android.uiautomator.UITestSample
```

```
INSTRUMENTATION_STATUS: test=testDemo
INSTRUMENTATION_STATUS_CODE: -1
INSTRUMENTATION_STATUS: stream=
Test results for WatcherResultPrinter=.E
Time: 12.371
```

```
FAILURES!!!
Tests run: 1, Failures: 0, Errors: 1
```

```
INSTRUMENTATION_STATUS: test=testDemo
INSTRUMENTATION_STATUS_CODE: 0
INSTRUMENTATION_STATUS: stream=
Test results for WatcherResultPrinter=.
Time: 5.676

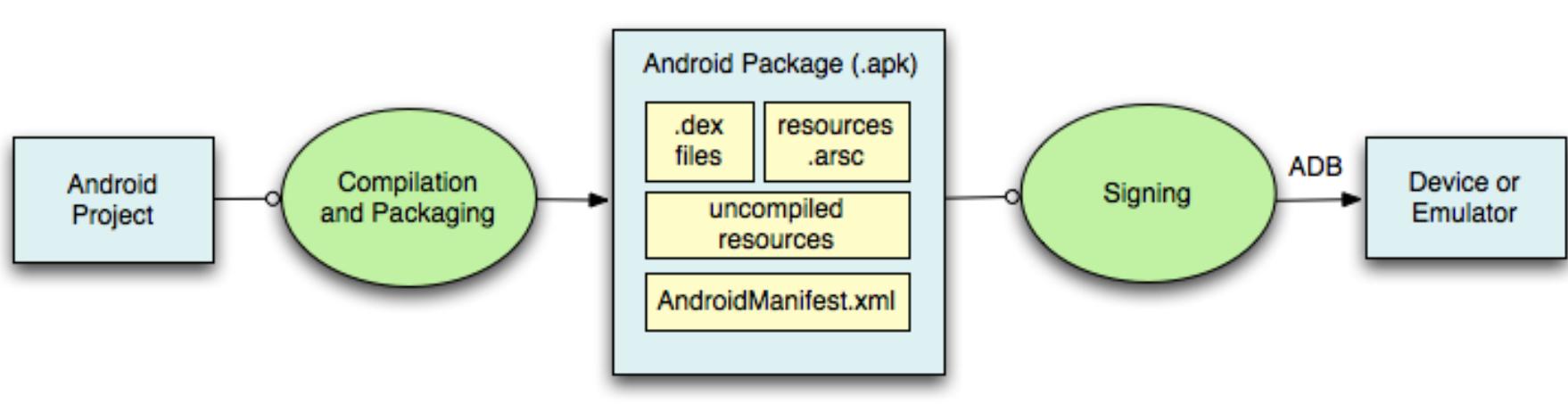
OK (1 test)
```

```
INSTRUMENTATION_STATUS_CODE: -1
```

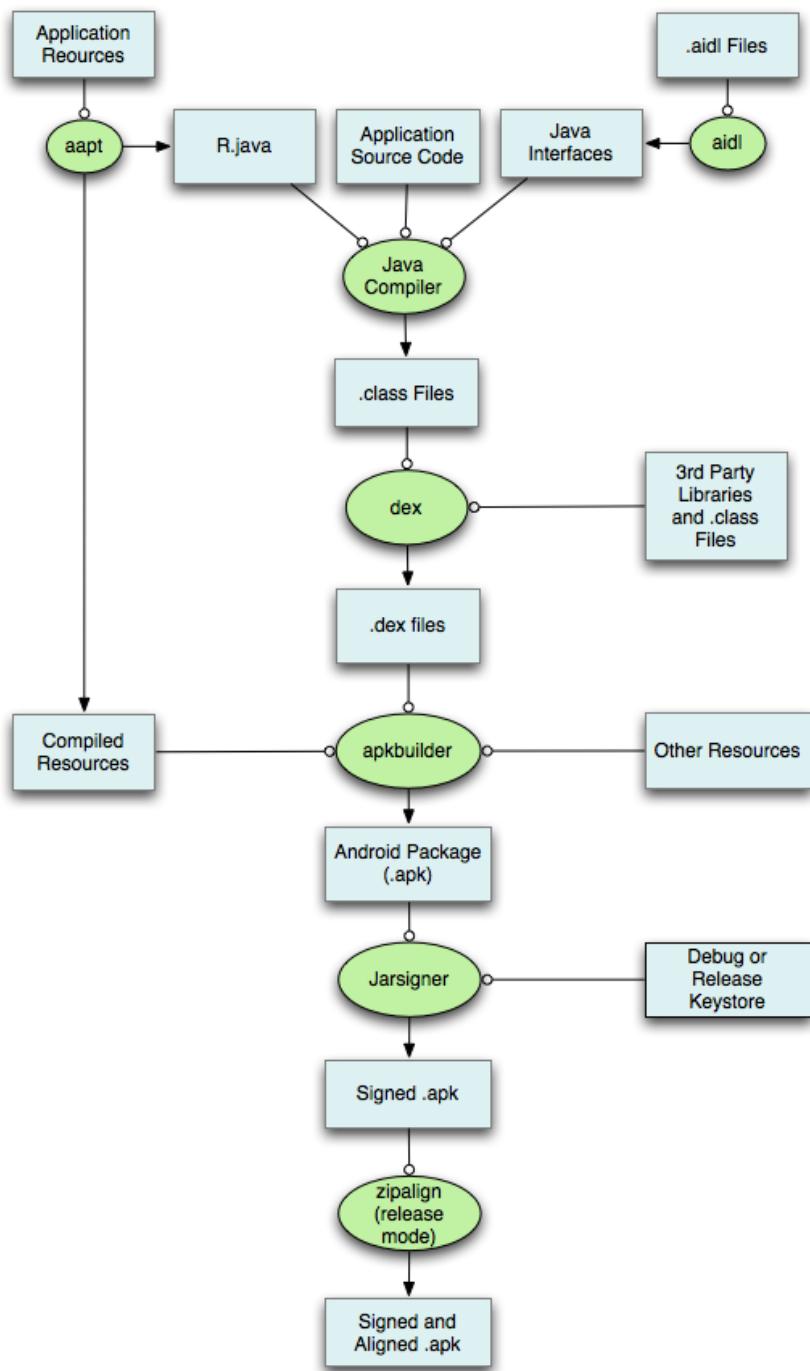


# Compilar y Ejecutar

## Procesos



# Compilar y Ejecutar.



# Depurar un Proyecto

## **ADB.**

- Sistema de comunicación entre un dispositivo/emulador y nuestro entorno de desarrollo

## **DDMS.**

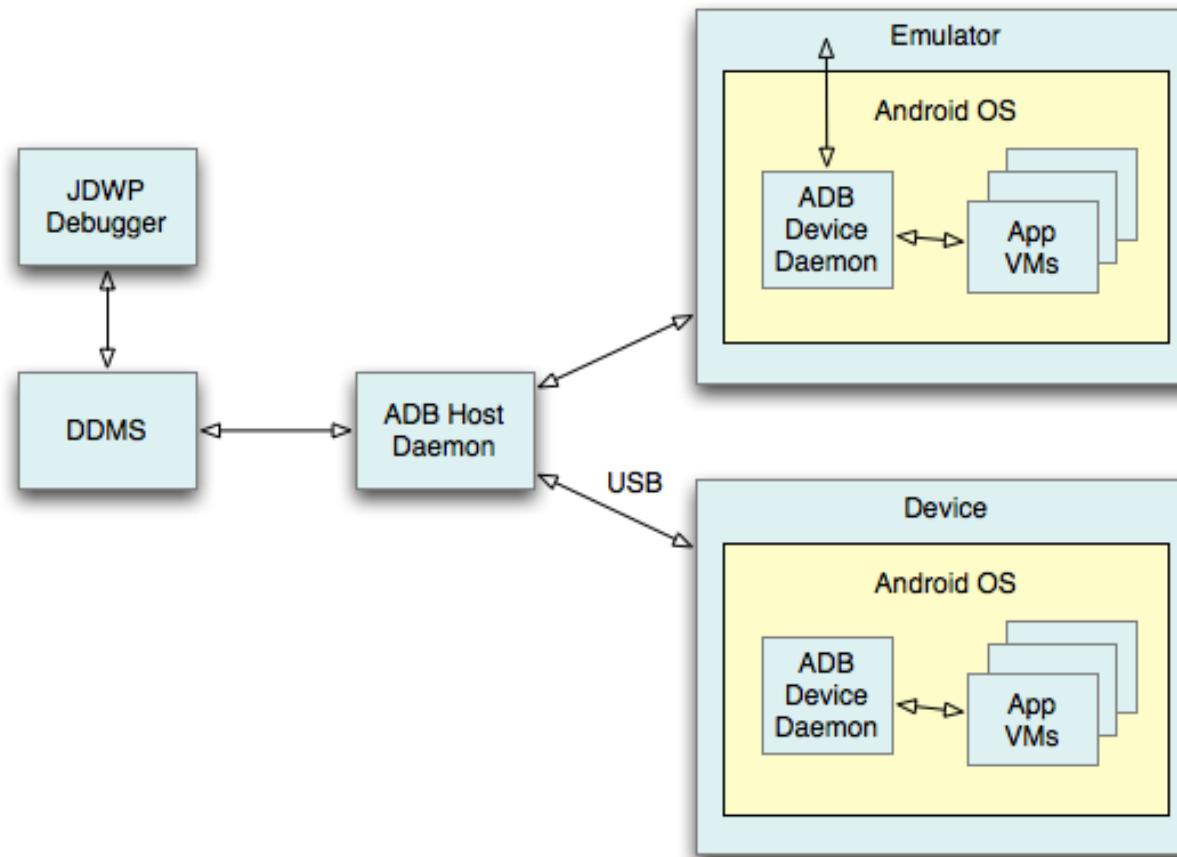
- «Dalvik Debug Monitor Server», Interfaz gráfica para ADB

## **JDWP debugger.**

- La máquina virtual Dalvik soporta el protocolo JDWP que permite unir un depurador a una máquina virtual. El DDMS se encarga de gestionar este protocolo.



# Depurar un Proyecto



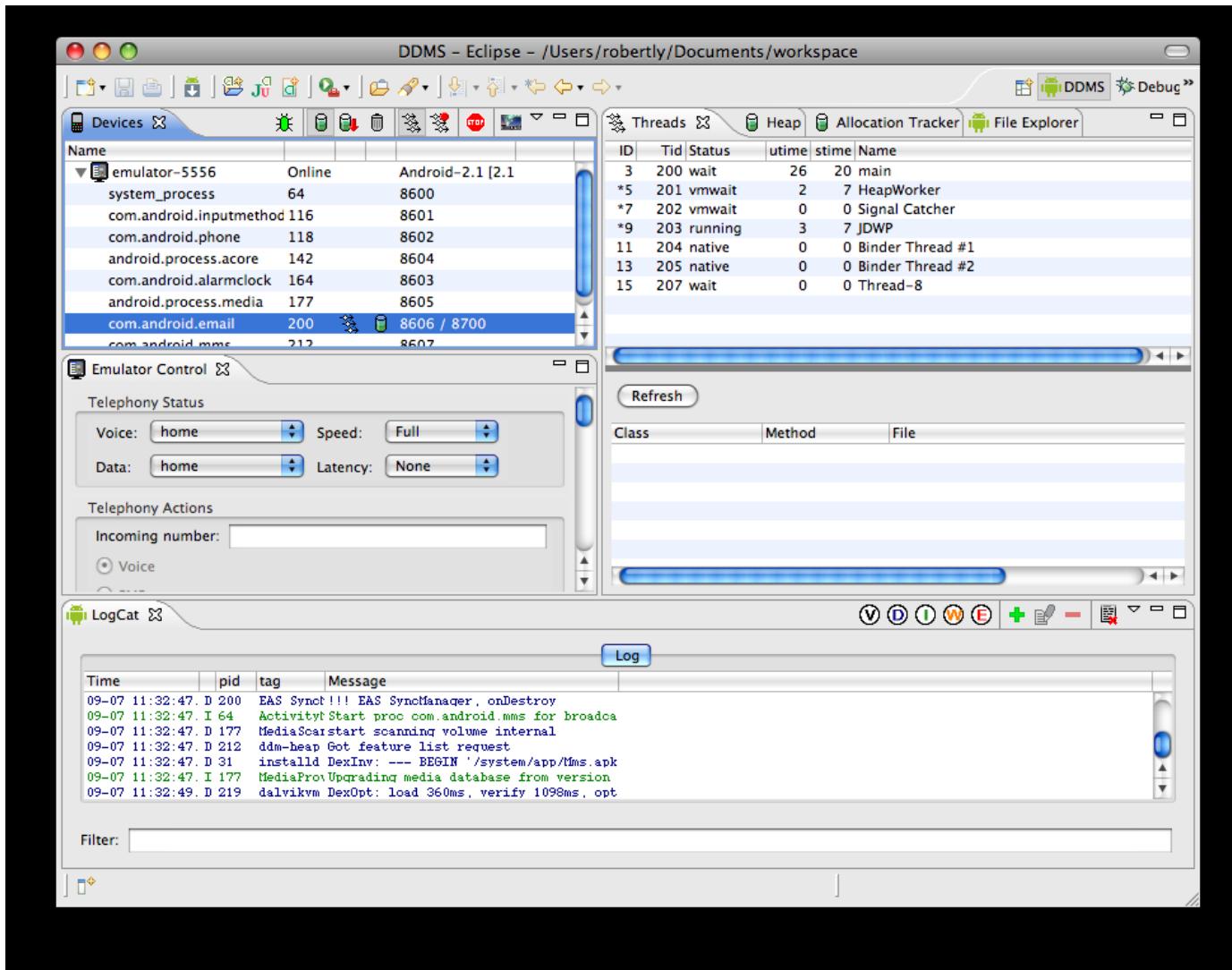
# Depurar un Proyecto

## DDMS.

- Devices.
  - Muestra el listado de dispositivos/emuladores conectados al ADB
- Emulator Control.
  - Permite ejecutar algunas funciones del emulador
- LogCat.
  - Muestra los mensajes del dispositivo en tiempo real.
- Threads
  - Muestra los hilos en ejecución dentro de la MV
- Heap
  - Muestra el uso de la pila de la MV
- Allocation Tracker
  - Muestra las direcciones de memoria de los objetos dentro de la MV
- File Explorer
  - Permite explorar el sistema de ficheros del dispositivo conectado



# Depurar un Proyecto



# Depurar un Proyecto

## DEBUG.

- Debug.
  - Muestra la aplicación que está siendo depurada
- Variables.
  - Valor de las variables cargadas en memoria
- LogCat.
  - Muestra los mensajes del dispositivo en tiempo real.
- Breakpoints
  - Muestra los puntos de ruptura insertados en el código de la aplicación.



# Depurar un Proyecto

Screenshot of the Android Studio IDE interface during debugging:

- Top Bar:** Shows the title "Debug" and various debug icons.
- Left Panel (Stacked Threads):** Displays the current stack trace for the main thread, showing calls from VisualtisAR.onCreate() up to DalvikVM's main loop.
- Middle Left Panel (Code Editor):** Shows the code for `MissileLauncherActivity.java`. The line `requestWindowFeature(Window.FEATURE_NO_TITLE);` is highlighted in green, indicating it is currently executing.
- Middle Right Panels:**
  - Variables:** Shows local variables `this` (VisualtisAR) and `savedInstanceState` (null).
  - Outline:** Shows the class structure of `es.centic.tramas.visualtis.VisualtisAR`, including fields like `RESPONSE_ZONA_DETALLES` and `RESPONSE_SETTINGS`.
- Bottom Left Panel (Console):** Displays log output from the Android system, including messages about failed re-installation and launch cancellation.
- Bottom Right Panel (LogCat):** Displays log output from the application itself, including messages from `ActivityManager` and `dalvikvm`.

```

77    @Override
78    public void onCreate(Bundle savedInstanceState) {
79        super.onCreate(savedInstanceState);
80
81        //Preparamos la actividad a pantalla completa
82        getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
83        requestWindowFeature(Window.FEATURE_NO_TITLE);
84        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
85
86        //Instanciamos los componentes de RA
87        mCameraView = new CustomCameraView(this);
88        mCenticARView = new CenticARView(this);
89        mCenticRadarView = new CenticRadarView(this);

```

```

2011-05-12 12:30:43 - Visualtis] Installing Visualtis.apk...
[2011-05-12 12:30:45 - Visualtis] Re-installation failed due to different application signatures.
[2011-05-12 12:30:45 - Visualtis] You must perform a full uninstall of the application. WARNING: This will remove the application completely.
[2011-05-12 12:30:45 - Visualtis] Please execute 'adb uninstall es.centic.tramas.visualtis' in a shell.
[2011-05-12 12:30:45 - Visualtis] Launch canceled!
[2011-05-12 12:38:01 - Visualtis] -----
[2011-05-12 12:38:01 - Visualtis] Android Launch!
[2011-05-12 12:38:01 - Visualtis] adb is running normally.
[2011-05-12 12:38:01 - Visualtis] Performing es.centic.tramas.visualtis.VisualtisSplash activity launch

```

Time	pid	tag	Message
05-12 12:41...	I	5168	System.out
05-12 12:41...	I	5168	System.out
05-12 12:41...	D	5168	dalvikvm
05-12 12:41...	D	5168	dalvikvm
05-12 12:41...	I	1276	ActivityManager
05-12 12:41...	I	1276	ActivityManager
05-12 12:41...	W	1276	ActivityManager



# Publicar una aplicación

Publicar una aplicación.

## Versionar la aplicación

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.package.name"
    android:versionCode="2"
    android:versionName="1.1">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        ...
    </application>
</manifest>
```

## Firmar la aplicación

```
$ keytool -genkey -v -keystore my-release-key.keystore
-alias alias_name -keyalg RSA -keysize 2048 -validity 10000
```



# Publicar una aplicación.

Publicar una aplicación.

## Checklist

### Before you consider your application ready for release:

1. Test your application extensively on an actual device
2. Consider adding an End User License Agreement in your application
3. Consider adding licensing support
4. Specify an icon and label in the application's manifest
5. Turn off logging and debugging and clean up data/files

### Before you do the final compile of your application:

6. Version your application
7. Obtain a suitable cryptographic key
8. Register for a Maps API Key, if your application is using MapView elements

### Compile your application

### After you compile your application:

9. Sign your application
10. Test your compiled application



# Ejercicios.

1. Instalar el entorno de trabajo
  - Configurar Eclipse
  - Configurar Android SDK
  - Añadir rutas al path
  - Instalar Plugin ADT
2. Crear un proyecto de aplicación
3. Crear un proyecto de biblioteca
4. Añadir una clase de testing
5. Crear varios AVD con distintas configuraciones
6. Configurar el tablet



# Ejercicios.

7. Depurar el proyecto aplicación
8. Configurar DDMS
9. Crear APK firmado

