



Introducción a Android



- Historia
- ¿Qué es Android?
- Arquitectura del sistema
- *Herramientas de desarrollo*
- Componentes Android
- Modelos de Negocio



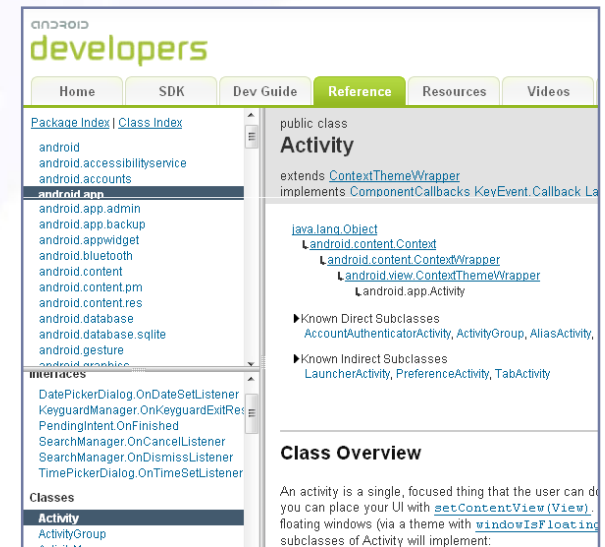
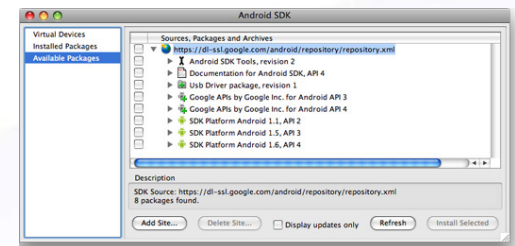
Herramientas de desarrollo

- ▶ Conocer las herramientas disponibles para desarrollar.
- ▶ Identificar que contiene un SDK de desarrollo Android.
- ▶ Configurar el IDE Eclipse.
- ▶ Análisis de las principales herramientas del SDK.
- ▶ Conocer la estructura de un proyecto Android.
- ▶ Crear una primera aplicación Android.



Entorno de desarrollo: Utilidades principales

- IDE Eclipse:
 - Menor esfuerzo y tiempo de desarrollo.
 - Presentación visual de componentes
- SDK Android
 - Kit de desarrollo software en Android.
 - Incluye herramientas (tools) & Add-ons.
 - Ejemplos & Documentación.
 - **API** para todas las versiones.
- Developer Guide: [URL](#)
- Foros Especializados
 - [Stackoverflow](#)
 - [Android10.org](#)



Entorno de desarrollo: Configuración

- Descarga e instalar eclipse: <http://www.eclipse.org/downloads>, escoger la opción de Eclipse IDE for Java Developers (unos 99 Megas). En la carpeta del curso (/software) hay un eclipse para agilizar la instalación.
- Descargar y instalar Android SDK: ([recomendado 1.6](#)). En la carpeta del curso (/software/SDK Android) se disponen de los SDKs en función del sistema (Windows, MAC, Linux).
- Descargar y instalar Android ADT plugin para Eclipse: <http://developer.android.com/sdk/eclipse-adt.html>. La forma más práctica: en Eclipse ir a HELP > Software Updates > Available software > add > <https://dl-ssl.google.com/android/eclipse/>
 - Si no funciona la URL o quitar https por http o instalar manual Available software > add > archive > .zip Name = "Android Plugin"
 - Para usuarios de Linux ayuda adicional en: <http://www.siliconkernel.com/tag/adt/>
- Variables de entorno: SDK/tools en el PATH. En eclipse: Window-preferences-poner el SDK.
- API & AVD: Windows – Android SDK – Available Packages – Instalar las dos API4. Luego crear un AVDs
- Workspace: Copiar workspace a la máquina local y importar los proyectos (File-Import-Existing projects...)

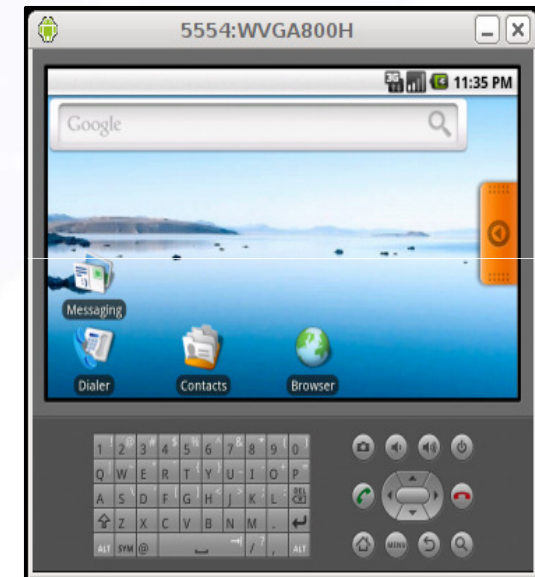
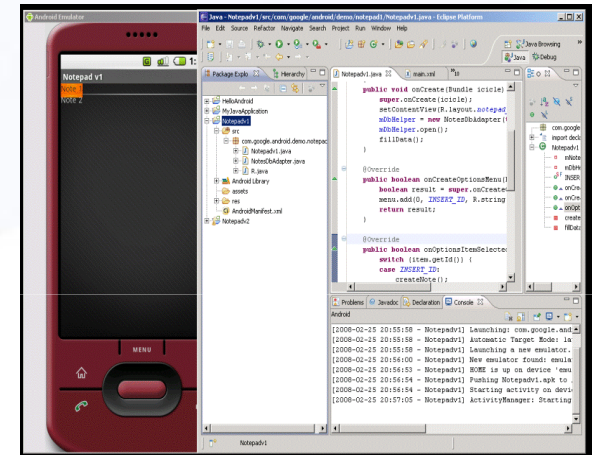


Entorno de desarrollo: SDK Android

Name	Description
<code>add-ons/</code>	Contains add-ons to the Android SDK development environment, which let you develop against external libraries that are available on some devices.
<code>docs/</code>	A full set of documentation in HTML format, including the Developer's Guide, API Reference, and other information. To read the documentation, load the file <code>offline.html</code> in a web browser.
<code>platforms/</code>	Contains a set of Android platform versions that you can develop applications against, each in a separate directory.
<code><platform>/</code>	Platform version directory, for example "android-1.6". All platform version directories contain a similar set of files and subdirectory structure.
<code>data/</code>	Storage area for default fonts and resource definitions.
<code>images/</code>	Storage area for default disk images, including the Android system image, the default userdata image, the default ramdisk image, and more. The images are used in emulator sessions.
<code>skins/</code>	A set of emulator skins available for the platform version. Each skin is designed for a specific screen resolution.
<code>templates/</code>	Storage area for file templates used by the SDK development tools.
<code>tools/</code>	Any development tools that are specific to the platform version.
<code>android.jar</code>	The Android library used when compiling applications against this platform version.
<code>samples/</code>	Sample code and apps that are specific to platform version.
<code>tools/</code>	Contains the set of development and profiling tools available to you, such as the emulator, the <code>android</code> tool, adb, ddms, and more.
<code>SDK Readme.txt</code>	A file that explains how to perform the initial setup of your SDK, including how to launch the Android SDK and AVD Manager tool on all platforms

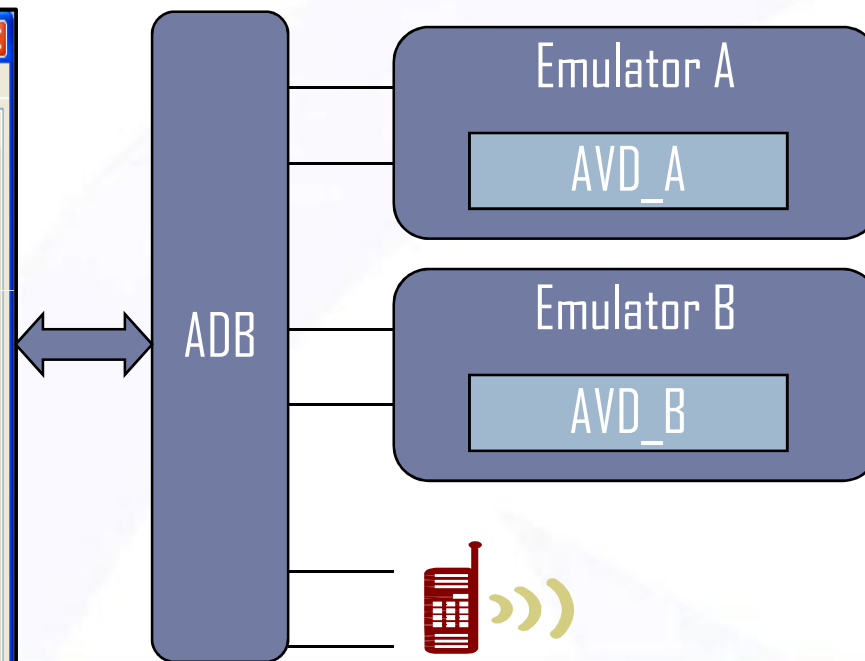
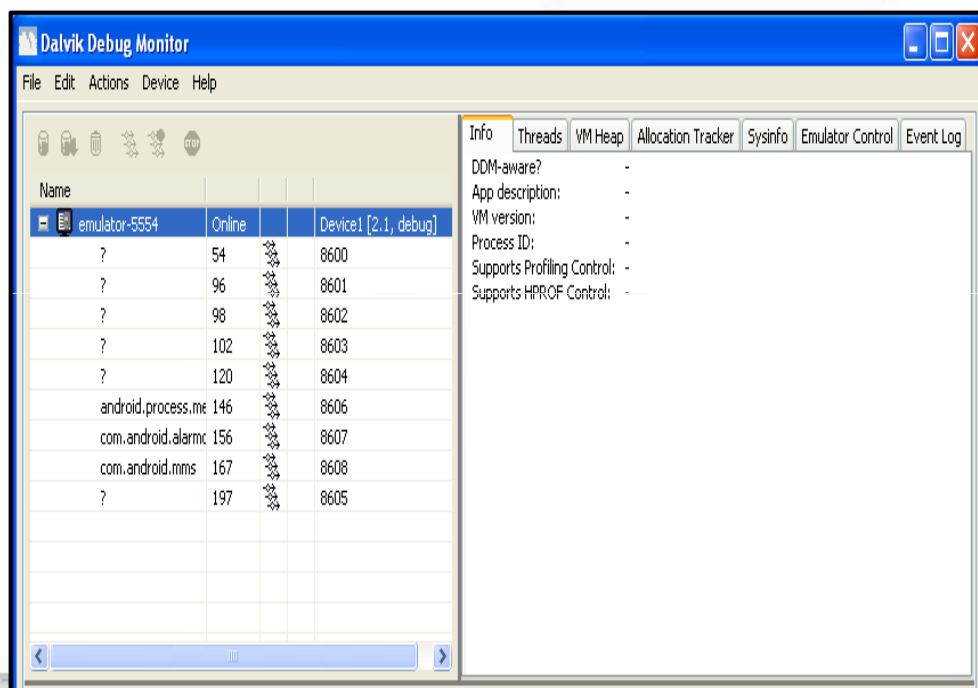
Entorno de desarrollo: %sdk_path%/tools/emulator

- El SDK contiene un emulador para testear las aplicaciones desarrolladas, emula hw y sw.
- Requiere de configuraciones AVD (Android Virtual Devices), para testear sobre distintas plataformas Android. (*android list avd | android create avd*)
- Para iniciar el emulador: *emulator -avd <avd_name>*: inicializa el emulador y carga una configuración AVD o vía Eclipse
- Para finalizarlo, basta con cerrar la ventana
- Se puede conectar vía consola textual, usando: *telnet localhost 5554* (puerto por defecto)
- Desde la consola, se pueden simular distintos eventos de hardware (ejemplo: *geo fix 41.1 12*) o software (*sms send ...*)



Herramientas: %sdk_path%/tools/DDMS

- Siglas de Dalvik Debug Monitor Server.
- Proporciona información del estado de las instancias Android ya sea dispositivo real o emulador.
 - Middleware entre IDE y device/emulador
- Monitoriza, vía el ADB, instancias de máquina virtual. (un servicio monitorización = 1 puerto = 1 device)
- En IDE Eclipse se puede acceder de forma integrada.



- Siglas de **Android Debug Bridge**.
- Gestiona el estado de una instancia del emulador o de un propio dispositivo Android.
- Programa cliente-servidor que se compone de:
 - Cliente:
 - Se ejecuta en la máquina desarrollo.
 - Se puede invocar desde un adb shell. (*ADT plugin o DDMS*).
 - Servidor:
 - Corre en un proceso background en la máquina de desarrollo.
 - Gestiona la comunicación entre cliente & daemon.
 - Daemon:
 - Proceso background en el emulador o dispositivo.



Android Debug Bridge

Plugin
ADT



ADB Client



ADB Server

Adb Shell

```
C:\WINDOWS.0\system32\cmd.exe - adb shell
config
cache
sdcard
acct
mnt
d
etc
system
sys
sbin
proc
init.rc
init.mahimahi.rc
init.goldfish.rc
init
default.prop
data
root
dev
$
```



ADB Client



Emulator A
AVD_A

Emulator B
AVD_B

DDMS

Name	Online	Version	Info	Threads	VM heap	Allocation Tracker	System	Emulator Control	Event Log
emulator-9554	Online	Android [2.1, debug]	DDM-Helper?						
?	54	8600	App description:						
?	96	8601	VM version:						
?	98	8602	Process ID:						
?	102	8603	Supports Profiling Control:						
?	120	8604	Supports HPROF Control:						
android.process.acore	146	8606							
com.android.alarm	156	8607							
com.android.mms	167	8608							
?	197	8605							

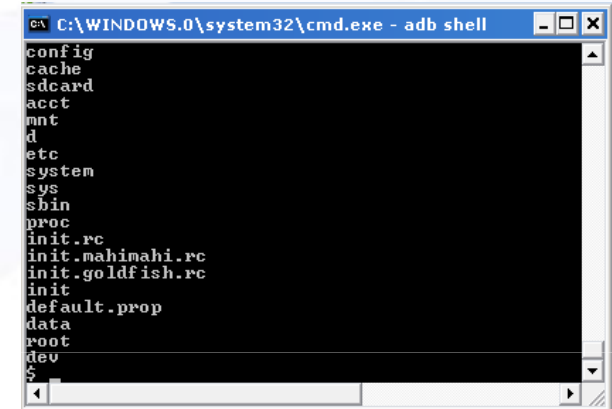


ADB Client

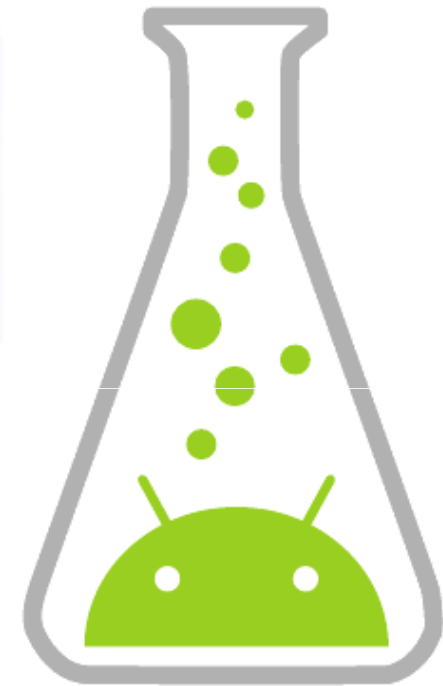


Ejemplo

- **Uso del ADB: en este ejemplo se muestra cómo utilizar la herramienta ADB para conectar con la instancia arrancada de Android**
- *adb devices*: muestra una lista de dispositivos activos accesibles.
- *adb shell*: Para conectar con la instancia, en el CMD:
- Se tiene un Linux minimalista: probar comando `pwd`, `cd`, `ls`, `top`,
- Navegar por el sistema de ficheros: `cd /system/app` (aplicaciones por defecto del sistema), `cd /sdcard` (acceso a la tarjeta SD), `cd /data/app` (aplicaciones instaladas por el usuario), `cd /data/app-private` (aplicaciones de pago), `cd /data/data` (datos asociados a las aplicaciones).
- *adb logcat*, se pueden ver las trazas que deja el sistema en la consola.



```
C:\WINDOWS.0\system32\cmd.exe - adb shell
config
cache
sdcard
acct
mnt
d
etc
system
sys
sbin
proc
init.rc
init.mahimahi.rc
init.goldfish.rc
init
default.prop
data
root
dev
$
```



➤ **android:**

- Crear, borrar & ver AVD.
- Actualizar SDKs & add-ons.
- Crear proyectos android.

➤ **dx:** Convierte .class en .dex

➤ **aapt:** (*Android Asset Packaging tool*) Empaqueta recursos en un .apk

➤ **mkcard:** Creación de sdcard

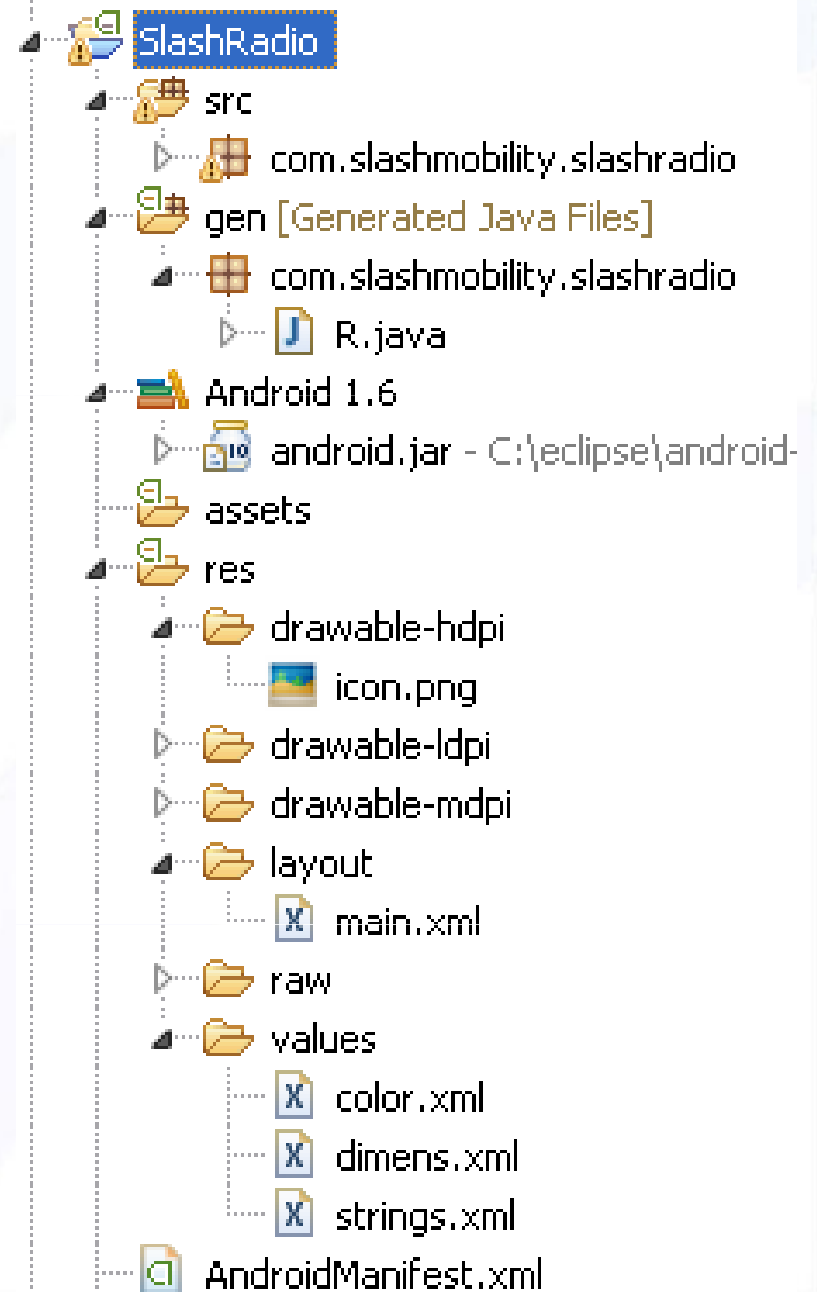
➤ **sqlite3:** Cliente sqlite.

➤ **aidl:** Android Interface Definition Language para IPC (Interprocess communication) similar a COM o Corba.



Estructura de un proyecto Android

- **src**: Código fuente
- **gen**: Binarios & R.java
- **android.jar**: Plataforma
- **res**: Recursos app
 - **drawable**: iconos & images
 - **layout**: organización UI
 - **raw**: música & video
 - **values**: Valores de la app
 - **Colors**: Colores
 - **dimens**: tamaño fuentes
 - **Strings**: literales
- **AndroidManifest.xml**



➤ Describe los componentes de la aplicación.

➤ Se declaran permisos y librerías necesarias para ejecutar la aplicación.



➤ **Icono, min SDK, Metadatos**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="prueba.Android"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".HelloWord"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-sdk android:minSdkVersion="7" />
</manifest>
```

Ejemplos y ejercicios: *Hola Mundo*

Ejemplo

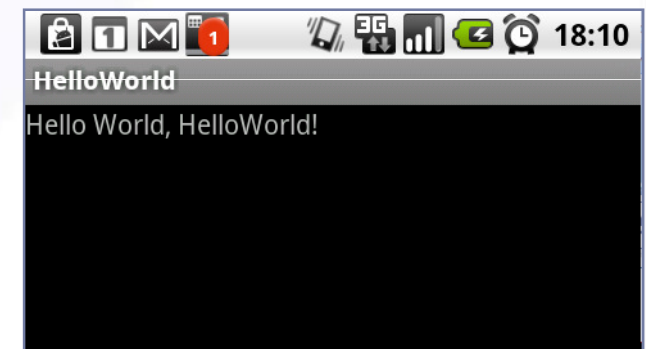
- **Hola mundo: en este ejemplo se muestra cómo crear nuestra primera aplicación Android con el eclipse**
- Eclipse: *File > New > Android Project*
- Target: *Android 1.6* Application Name: *HelloAndroid*
- Hello Android: *Run As > Android Application*



```
package com.android.helloandroid;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class HelloAndroid extends Activity {
    /** Called when the activity is first created. */
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TextView tv = new TextView(this);
        tv.setText("Hello, Android");
        setContentView(tv);
    }
}
```



Herramientas de desarrollo

- Conocer las herramientas disponibles para desarrollar.
- Identificar que contiene un SDK de desarrollo.
- Configurar el IDE Eclipse.
- Análisis de las principales herramientas del SDK.
- Conocer la estructura de un proyecto Android.
- Crear una primera aplicación Android.

