



Desarrollando Aplicaciones para Android



Tania Pérez



Índice



- **¿Qué es Android?**
- **Historia**
- **Características**
- **Competidores**
- **Dispositivos**
- **Arquitectura**
- **Modelo de Aplicaciones**
- **Herramientas de Desarrollo**



¿Qué es Android?



- **Android es un Sistema Operativo además de una plataforma de Software basada en el núcleo de Linux.**

- **Android es una plataforma de código abierto.**



Historia





- Julio 2005
 - Google compra Android Inc
- Noviembre 2007
 - Nace la Open Handset Alliance
 - Lanzamiento inicial del Android Software Development Kit
- Octubre 2008
 - Lanzamiento de Android 1.0 SDK (Release 1)
 - Se abre Android Market



Historia



- Marzo 2009
 - Google lanza la version 1.1 for the Android “dev phone”.
- Mayo 2009 
 - Google lanza la versión 1.5 de Android OS (llamada Cupcake)
- Septiembre 2009 
 - Google lanza Android 1.6 (Donut)

Historia



- Noviembre 2009
 - Motorola Droid : fue lanzado con Android 2.0 “Eclair”.
 - Consigue vender 1.05 millones de unidades en 74 días, superando el record establecido por el iPhone.
- Diciembre 2009
 - 16.000 aplicaciones en el Market



Historia



- Enero 2010
 - Google Nexus One (HTC) : Android 2.1 (algunos lo llamaron Flan, pero Google sigue considerándolo parte de Eclair)
- Febrero 2010
 - 60.000 teléfonos con Android vendidos al día
- Junio 2010
 - Lanzamiento Android – Froyo versión 2.2



Historia



- Android 2.2 "Froyo"



Historia



¿ FUTURO DE ANDROID ?

- Android Market: Tienda de aplicaciones con mayor crecimiento. Actualmente ~ 40.000 aplicaciones
- Android S.0: el que más crece, casi por superar al iphone en EEUU.



Características



- **Framework de aplicaciones:** permite el reemplazo y la reutilización de los componentes.
- **Navegador integrado:** basado en el motor open Source Webkit.
- **SQLite:** base de datos para almacenamiento estructurado que se integra directamente con las aplicaciones.
- **Multimedia:** Soporte para medios con formatos comunes de audio, video e imágenes planas (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).



Características



- **Máquina virtual Dalvik:** Base de llamadas de instancias muy similar a Java.
- **Telefonía GSM:** dependiente del terminal.
- **Bluetooth, EDGE, 3g y Wifi:** dependiente del terminal.
- **Cámara, GPS, brújula y acelerómetro:**
Dependiente del terminal
- **Pantalla Táctil.**



Competidores



Competidores



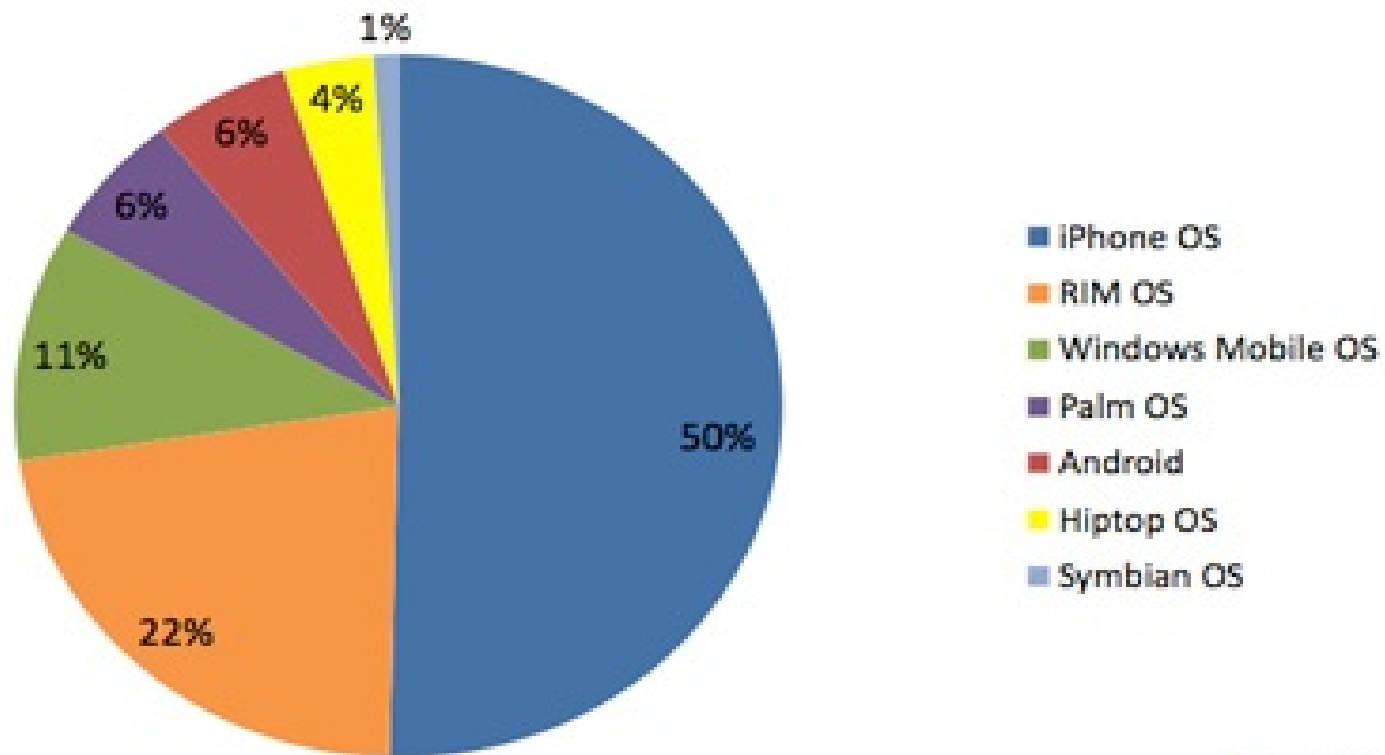
- Apple
 - iPhone OS
- RIM
 - Blackberry
- Microsoft
 - Windows Mobile 6.5 Phone 7 Series
- Symbian Foundation
 - Symbian OS
- Palm
 - WebOS



Competidores



US Operating Systems
March 2009

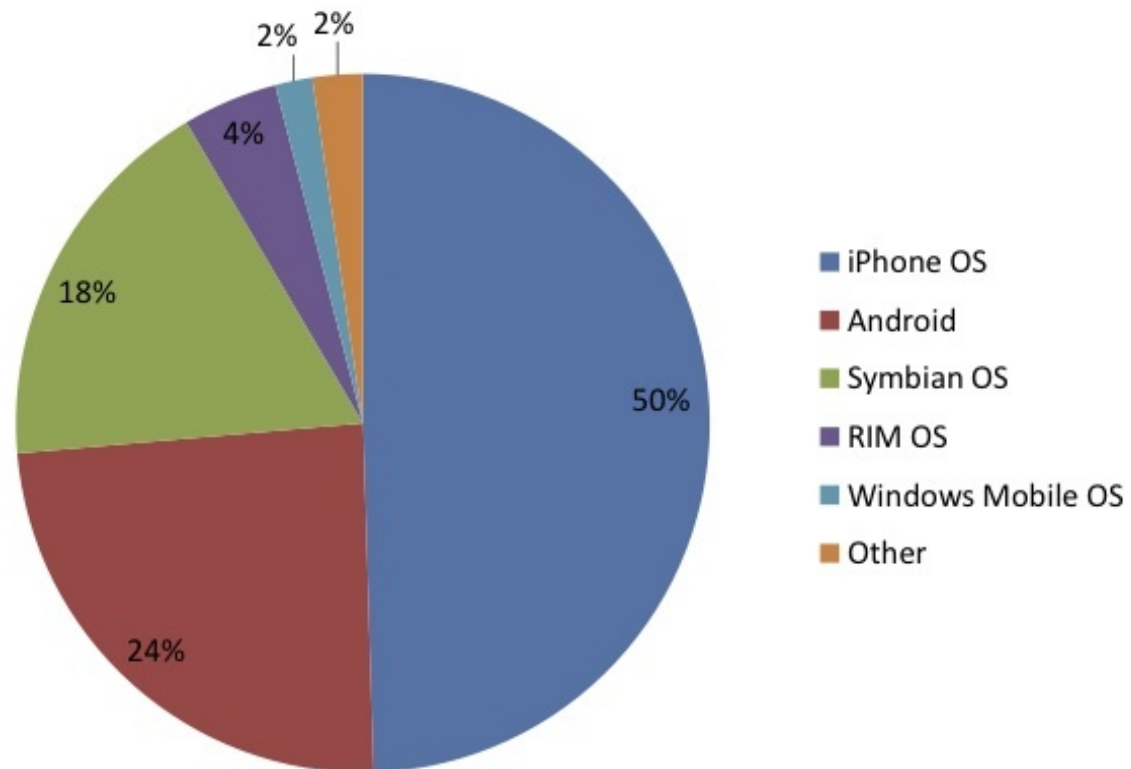


admob

Competidores



Operating System Share, Worldwide Smartphones Only, February 2010



admob^(TM)

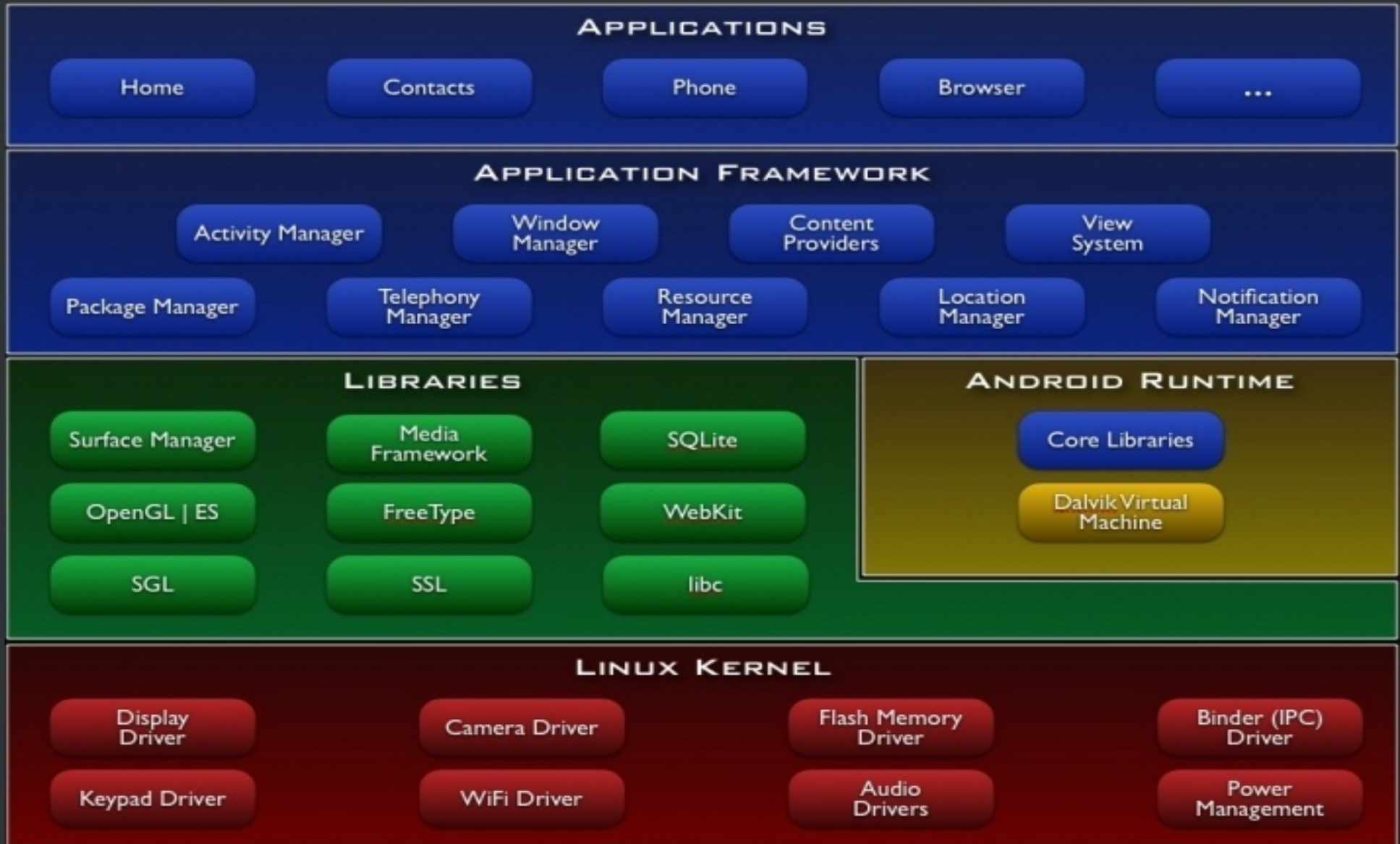
Dispositivos



- **HTC**
 - Dream, Magic (myTouch3G), Hero, Tattoo, Nexus One, Desire, Legend
- **Motorola**
 - CLIQ, Droid (Milestone), DEXT
- Acer, Archos, Dell, Huawei, Lenovo, LG, Samsung, Sony Ericsson,...



Arquitectura



Modelo de aplicaciones

¿Cómo son las aplicaciones Android?



- Las aplicaciones están escritas en Java y empaquetadas en formato Android package (.apk)
- Cada aplicación es independiente
 - Se ejecuta en su propio proceso de Linux
 - Cada proceso tiene su propia máquina virtual Java
 - Cada aplicación tiene asignado un ID de usuario de Linux



Modelo de aplicaciones

Componentes de una aplicación



- **Activities**
- **Services**
- **Broadcast receivers**
- **Content providers**



Modelo de aplicaciones



Qué es una activity y para qué sirve

- Activity
 - Presenta una interfaz de usuario enfocada a una tarea que el usuario desea realizar
 - Reproducir un archivo de audio, acceder a una determinada URI, realizar una llamada telefónica...
 - Se puede pensar en una actividad como el elemento análogo a una ventana en cualquier otro lenguaje visual
 - Una aplicación Android normalmente consistirá en una ó más actividades que trabajan de modo colaborativo
 - Una de las actividades que componen la aplicación será la que se ejecute al iniciarse la aplicación

Modelo de aplicaciones

¿Qué es un servicio Android?



- Service
 - A diferencia de una Activity, no está asociado a una interfaz de usuario
 - Sirve para realizar tareas en background durante un tiempo indefinido
 - Ejemplo: Reproductor de audio
 - Necesita de una interfaz para interactuar con el usuario(activity)
 - Inicio de la reproducción, parada, etc
 - Otros componentes ó aplicaciones pueden acceder al servicio

Modelo de aplicaciones

Broadcast receivers



- Broadcast receivers
 - No realizan acciones por sí mismos
 - Recibe eventos de tipo broadcast (eventos globales) y reacciona ante ellos
 - Muchos de ellos están originados por el sistema
 - Batería baja, SMS recibido, Tarjeta SD insertada...
 - Las aplicaciones pueden lanzar un broadcast si necesitan que otra aplicación realice alguna acción por ellas
 - Un broadcast receiver no tiene interfaz de usuario, pero puede lanzar una para atender la petición

Modelo de aplicaciones

Content providers



- Content providers
 - Es el mecanismo que se ha definido en Android para compartir datos entre aplicaciones
 - Es posible compartir datos de nuestra aplicación sin mostrar detalles sobre su almacenamiento interno, su estructura, o su implementación
 - Los datos pueden estar almacenados en cualquier lugar: fichero, SQLite, internet...
 - Nuestra aplicación podrá acceder a los datos de otra a través de los content provider que se hayan definido
 - Hace uso de un ContentResolver para acceder a los datos expuestos por un content provider



Modelo de aplicaciones

¿Cómo se activan los distintos componentes?



- Content providers
 - Al recibir una petición desde un Content resolver
- Activities, Services y Broadcast receivers
 - Al recibir un mensaje asíncrono llamado *Intent*
- Intent
 - Es el elemento básico de comunicación entre los distintos componentes Android. Se pueden entender como los mensajes ó peticiones que son enviados entre los distintos componentes de una aplicación o entre distintas aplicaciones. Mediante un intent se puede mostrar una actividad desde cualquier otra, iniciar un servicio, enviar un mensaje broadcast, iniciar otra aplicación, etc.

Herramientas de Desarrollo



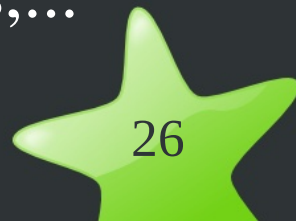
- **Android SDK**
 - Paquete con las herramientas necesarias para desarrollar aplicaciones
 - Desde el SDK 1.6, aparece Android SDK and AVD Manager, permite actualizar y descargar nuevas versiones del SDK de forma cómoda
 - Incluye emulador, driver para el dispositivo (Windows), herramientas de depuración, etc.



Herramientas de Desarrollo



- **Android SDK**
 - Android Development Tools (ADT)
 - Plug-in para Eclipse
 - Incluye vista DDMS
 - Procesos en ejecución
 - Navegación por el sistema de ficheros
 - Logs
 - Proyecto, editor de vistas, compilación de recursos, firma y exportación de aplicaciones,...



Herramientas de Desarrollo



- Android NDK
 - Añadido al SDK
 - Permite escribir partes críticas de nuestra aplicación en código nativo (C/C++)
 - Soporte para OpenGL ES 2.0
 - Android >1.5
 - Ideal para videojuegos



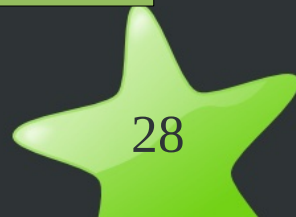
Herramientas de Desarrollo



GUÍA DE INSTALACIÓN DEL SDK DE ANDROID

1. JDK Java
2. Eclipse
3. SDK Android
4. ADT Plugin para Eclipse

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



Instalando Android SDK en Linux



- Necesitamos dos cosas:
 - SDK Android
 - Plataforma de desarrollo (Eclipse)
 - No usar la versión que está disponible en los repositorios, ya que carece de algunos elementos necesarios



Instalando Android SDK en Linux




- **Paquetes necesarios**
- Antes de instalar el SDK de Android y Eclipse necesitamos algunos paquetes fundamentales para que todo funcione a la perfección. Lo más importante es tener instalado Java por lo que necesitamos el paquete sun-java6-bin:
 - `:~$ sudo aptitude install sun-java6-bin`
- Además, si tu máquina es de 64 bits vas a necesitar las librerías de 32 bits ya que el SDK de Android está programado en 32 bits:
 - `:~$ sudo aptitude install ia32-libs`



Instalando Android SDK en Linux



- Descargar Eclipse

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



Eclipse IDE for Java Developers, 99 MB

Downloaded 732,540 Times

[Details](#)



Linux 32 Bit

Linux 64 Bit



Instalando Android SDK en Linux



- Instalar Android SDK
- Descargar Android SDK
 - Debemos descargar el SDK de Android de la web oficial: Android Developers. Elegimos la versión para Linux y tras aceptar la licencia guardaremos el fichero en nuestro disco duro.
- Descomprimir Android SDK
 - Para descomprimirlo no hay más que pulsar sobre él con el botón derecho y hacer click en “Extraer aquí”, lo que creará una carpeta nueva, el fichero lo podéis eliminar. El Android SDK estará pues en la ruta:
- `/home/{usuario}/.../android-sdk-linux_86`

Instalando Android SDK en Linux



- Preparar Android SDK para la instalación
 - Antes de poder instalar nada tenemos que declarar el path para que el sistema encuentre los ejecutables. Para ello hay que editar el fichero *.bashrc* alojado en el directorio *home*:
 - `:~$ gedit ~/.bashrc`
 - Y añadir al final de fichero la siguiente línea:
 - `export PATH=${PATH}:/home/{usuario}/.../android-sdk-linux_86/tools`



Instalando Android SDK en Linux



- Instalar Android SDK
- Es necesario cerrar cualquier terminal abierto para que haga efecto el path que acabamos de definir. Abrir uno nuevo y escribir:
 - `:~$ android`
- Esto hará que se abra el mánager de Android donde podremos instalar y actualizar el SDK así como crear dispositivos virtuales de pruebas.



Instalando Android SDK en Linux



- Accedemos a la sección *Available Packages* donde pulsaremos el botón Refresh. Si obtenemos error al acceder a la URL, ir a la sección *Settings*, marcar la casilla *Force https://... sources to be fetched using http://...* y pulsar en *Save & Apply*
- Volvemos a *Available Packages* y pulsamos Refresh para que nos aparezca la lista de paquetes que podemos instalar. Es importante que NO esté seleccionada la casilla *Display update only* ya que esta primera vez queremos instalarlo todo. Seleccionamos todos los paquetes y pulsamos *Install Selected*
- Se abrirá una nueva ventana para aceptar las licencias de dichos paquetes. Seleccionamos *Accept All* y pulsamos *Install Accepted*. Comenzará la instalación del SDK de Android.



Instalando Android SDK en Linux



- Configurar Eclipse
- Seleccionar la ruta para guardar nuestro espacio de trabajo
- Ahora hay que instalar el plugin del SDK de Android así que vamos a *Help* > *Install New Software...* donde debemos pulsar el botón *Add...* y poner los siguientes datos:
 - Name: Android Plugin
 - Location: <https://dl-ssl.google.com/android/eclipse/>
- Aparecerá un *Developer Tools* con dos plugins colgando *Android DDMS* y *Android Development Tools*. Seleccionamos todo y pulsamos *Next* > para que compruebe las dependencias necesarias.

Instalando Android SDK en Linux



- En la pantalla siguiente debemos aceptar los términos de la licencia y pulsamos Finish
- Al finalizar habrá que reiniciar Eclipse
- Por último debemos referenciar el SDK. Para ello vamos a Window > Preferences > Android y donde pone SDK Location pulsamos el botón Browse... y buscamos la carpeta donde está instalado el SDK de Android
 - `/home/{usuario}/.../android-sdk-linux_86`
- Pulsamos Apply y esperamos mientras se carga el Android SDK Content Loader. Finalmente aparecerá la lista de todo lo que hemos instalado. Con esto conseguiremos importar el SDK de Android a Eclipse
- Listo. Ya tenemos todo preparado para empezar a desarrollar aplicaciones para Android desde Ubuntu Linux.



Estructura de un proyecto Android



¿De qué elementos se compone y cómo se organizan?

- Trabajaremos usando el plugin de Android para el IDE Eclipse
- Cuando creamos un nuevo proyecto Android en Eclipse se genera automáticamente la estructura de carpetas necesaria para poder generar posteriormente la aplicación. Esta estructura será común a cualquier aplicación, independientemente de su tamaño y complejidad
- Estructura inicial de un proyecto Android en Eclipse:



Estructura de un proyecto Android

Descripción de elementos principales



- Carpeta */res/*
- Contiene todos los ficheros de recursos necesarios para el proyecto: imágenes, vídeos, cadenas de texto, etc. Los diferentes tipos de recursos de deberán distribuir entre las siguientes carpetas:
 - */res/drawable/* Contiene las imágenes de la aplicación. Se puede dividir en */drawable-ldpi*, */drawable-mdpi* y */drawable-hdpi* para utilizar diferentes recursos dependiendo de la resolución del dispositivo.
 - */res/layout/* Contiene los ficheros de definición de las diferentes pantallas de la interfaz gráfica. Se puede dividir en */layout* y */layout-land* para definir distintos layouts dependiendo de la orientación del dispositivo.

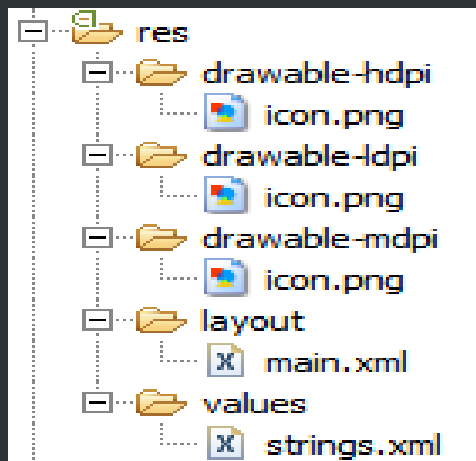


Estructura de un proyecto Android



Descripción de elementos principales

- */res/anim/* Contiene la definición de las animaciones utilizadas por la aplicación
- */res/menu/* Contiene la definición de los menús de la aplicación
- */res/values/* Contiene otros recursos de la aplicación como por ejemplo cadenas de texto (strings.xml), estilos (styles.xml), colores (colors.xml), etc
- */res/xml/* Contiene los ficheros XML utilizados por la aplicación
- */res/raw/* Contiene recursos adicionales que no se incluyan en el resto de carpetas de recursos. Archivos multimedia de audio y vídeo



Estructura de un proyecto Android



Descripción de elementos principales

- Carpeta */gen/*
- Contiene una serie de elementos de código generados automáticamente al compilar el proyecto. Cada vez que generamos nuestro proyecto, la maquinaria de compilación de Android genera por nosotros una serie de ficheros fuente en java dirigidos al control de los recursos de la aplicación.
- El más importante es el que se puede observar en la imagen, el fichero R.java, y la clase R.



Estructura de un proyecto Android



Descripción de elementos principales

- Esta clase R contendrá en todo momento una serie de constantes con los ID de todos los recursos de la aplicación incluidos en la carpeta */res/*, de forma que podamos acceder fácilmente a estos recursos desde nuestro código a través de este dato. Así, por ejemplo, la constante *R.drawable.icon* contendrá el ID de la imagen “icon.png” contenida en la carpeta */res/drawable/*. Veamos como ejemplo la clase R creada por defecto para un proyecto nuevo:

```
R.java ✖
+ /* AUTO-GENERATED FILE. DO NOT MODIFY.

package com.src;

public final class R {
    public static final class attr {
    }
    public static final class drawable {
        public static final int icon=0x7f020000;
    }
    public static final class layout {
        public static final int main=0x7f030000;
    }
    public static final class string {
        public static final int app_name=0x7f040001;
        public static final int hello=0x7f040000;
    }
}
```



Estructura de un proyecto Android

Descripción de elementos principales



- Carpeta */assets/*
 - Contiene todos los demás ficheros auxiliares necesarios para la aplicación (y que se incluirán en su propio paquete), como por ejemplo ficheros de configuración, de datos, etc.
 - La diferencia entre los recursos incluidos en la carpeta */res/raw/* y los incluidos en la carpeta */assets/* es que para los primeros se generará un ID en la clase R y se deberá acceder a ellos con los diferentes métodos de acceso a recursos. Para los segundos sin embargo no se generarán ID y se podrá acceder a ellos por su ruta como a cualquier otro fichero del sistema. Usaremos uno u otro según las necesidades de nuestra aplicación.



Estructura de un proyecto Android



Descripción de elementos principales

- Fichero *AndroidManifest.xml*
- Contiene la definición en XML de los aspectos principales de la aplicación, como por ejemplo su identificación (nombre, versión, icono, ...), sus componentes (pantallas, mensajes, ...), o los permisos necesarios para su ejecución
- Declaración de los permisos requeridos
 - *uses-permission*
 - Acceso a determinado estado del teléfono, acceso a internet, etc.
 - Los permisos que requiere se muestran al usuario al instalar la aplicación
 - Declaración de permisos propios
 - Restringe qué aplicaciones pueden hacer uso de un determinado componente



Estructura de un proyecto Android

AndroidManifest



```
<?xml version="1.0" encoding="utf-8"?>

  <manifest ... >

    <application ... >

      <activity android:name="com.gul.example.SampleActivity"
        android:icon="@drawable/small_pic.png"
        android:label="@string/sampleActivity"
        ... >

        </activity>

        ...

      </application>

    </manifest>
```



Referencias



- Android Developers
 - <http://developer.android.com/index.html>



FIN



Gracias por vuestra asistencia.



WELCOME TO
ANDROID WORLD!

- Sugerencias y dudas: www.gul.es – Lista del gul

