



## UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

**Materia:** Programación Orientada a Objetos

**Docente:** Carlos Alberto Fernández y Fernández

**Trabajo:** Investigación sobre ANDROID

**Alumnos:** Cruz Alonso Rafael

Gopar Mecinas Lilibeth

López López Iván

Moya García Edith

**Grupo:** 601

**Carrera:** Ingeniería en Computación.

---

## ANDROID

### INTRODUCCIÓN

#### HISTORIA

**Google presenta a Android, su sistema operativo para teléfonos móviles.**

Apple presenta al mundo su iPhone, un terminal destinado a revolucionar la telefonía móvil con un diseño innovador y una interfase de usuario muy sencilla de manejar.

Fue más o menos por esas fechas cuando surgieron los primeros rumores en los que se afirmaba que Google estaba preparando para ingresar al campo de la telefonía móvil. Con el antecedente del iPhone aún muy reciente, inmediatamente se comenzó a especular con la posibilidad de que en las filas del buscador estaban trabajando en un terminal propia en el que iban a integrar gran parte de sus servicios online, como es el caso de Google Maps o Gmail.



Bastaron esos rumores para que en poco tiempo la Red se llenara de artículos en los que se hablaba sobre las supuestas características técnicas que iba a tener este dispositivo y cómo iba a competir con los móviles de última generación que aparecían por aquellos días.

Pero lo cierto es que pasaron los meses y Google jamás presentó móvil alguno ni dio muestras de estar trabajando en uno; el Google estaba preparando su irrupción en el terreno de la telefonía móvil, eso era cierto, pero no con un terminal propio sino con una plataforma libre llamada a competir con Windows Mobile y Symbian que podrían adoptar cuantos fabricantes así lo desearan.

Varias semanas después Google finalmente, presenta al mundo su proyecto en el que ha estado trabajando. **Se trata de un sistema operativo basado en GNU/Linux al que llamaron Android.** Antes de haber sido mostrarlo en sociedad,

el buscador ha cerrado acuerdos con hasta 34 compañías del sector, entre las que se encuentran Samsung, HTC, Qualcomm, Motorola, Telefónica y T-Mobile, que se han comprometido a comercializar terminales impulsados por esta plataforma a partir del próximo año.

A todo esto surge una pregunta: ¿por qué ha desarrollado Google un sistema operativo de estas características? La respuesta, es de lo más lógica: **para extender su influencia en el campo de la publicidad online a los dispositivos móviles**, hasta ahora en manos de Microsoft gracias a Windows Mobile, el sistema operativo más utilizado en los terminales de última generación junto con Symbian OS.

Esta situación podría permitir a dichas compañías controlar la manera en que la gente accede a la Red e integrar sus propias plataformas publicitarias en los móviles del futuro, algo que no satisface en absoluto a Google ya que podría ver en peligro su supremacía en un terreno, que no lo olvidemos, supone el 99% de sus ingresos totales año tras año.

## **DEFINICIÓN DE ANDROID**

A continuación se muestran algunas definiciones de Android:

- ☀ Es una plataforma de software para dispositivos móviles que incluye un sistema operativo y aplicaciones base.
- ☀ Android es un conjunto de herramientas y aplicaciones vinculadas a una distribución Linux para dispositivos móviles. Por sí solo no es un sistema operativo.
- ☀ Android es de código abierto, gratuito y no requiere pago de licencias.
- ☀ Android es una plataforma de código abierto para dispositivos móviles que está basada en Linux y desarrollada por Open handset alliance, se prevee que los primeros teléfonos con Android aparezcan en el segundo semestre de 2008 y compañías poderosas como LG, Motorola y HTC ya han diseñado alguno de los prototipos que incorporarán el sistema Android.
- ☀ Es una stack de software para dispositivos móviles que incluye un sistema operativo, middleware y aplicaciones base. Los desarrolladores pueden crear aplicaciones para la plataforma usando el SDK de android. Las solicitudes se han escrito utilizando el lenguaje de programación Java y se ejecutan en Dalvik, una máquina virtual personalizada que se ejecuta en la parte superior de un núcleo de Linux.

## **LINUX KERNEL**

Android se basa en la versión de Linux para 2.6 sistema de servicios básicos tales como seguridad, gestión de memoria, gestión de procesos, la pila de red, y el conductor modelo. El kernel también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

Android permiten a los desarrolladores crear aplicaciones móviles y sacar el máximo provecho que todo dispositivo tiene que ofrecer. Está construido para ser realmente abierto. Por ejemplo, una aplicación puede llamar a cualquier función básica de un teléfono, como hacer llamadas, enviar mensajes de texto, o usar la cámara. Android se basa en el Kernel Linux. Además, utiliza una máquina virtual personalizada que ha sido diseñada para optimizar la memoria y los recursos de hardware en un entorno móvil.

Android va a ser de código abierto, puede ser libremente ampliado para incorporar nuevas tecnologías de vanguardia a medida que vayan surgiendo. La plataforma continuará evolucionando a medida que la comunidad de desarrolladores trabajan juntos para crear innovadoras aplicaciones móviles.

Android no diferencia entre el núcleo del teléfono y las aplicaciones de terceros. Todas pueden ser construidas para tener igual acceso a un teléfono y tienen la capacidad de ofrecer a los usuarios un amplio espectro de aplicaciones y servicios. Con los dispositivos construidos en la Plataforma Android, los usuarios podrán adaptarse plenamente al teléfono para sus intereses.

## **APLICACIONES ROMPE FRONTERAS**

Android rompe las barreras en la creación de nuevas e innovadoras aplicaciones. Por ejemplo, un programador puede combinar la información de la web con los datos sobre un individuo de telefonía móvil (como los contactos del usuario, calendario o ubicación geográfica) para proporcionar una mayor notabilidad en la experiencia del usuario. Con Android, un desarrollador puede crear una aplicación que permite a los usuarios ver la ubicación de sus amigos y ser alertado cuando se encuentran cerca, por ejemplo a unas cuadras de donde se encuentra el usuario, dándoles la oportunidad de conectarse.

## **RÁPIDO Y FÁCIL DESARROLLO DE APLICACIONES**

Android proporciona acceso a una amplia gama de útiles bibliotecas y herramientas que pueden ser utilizadas para construir aplicaciones variadas. Por ejemplo, Android permite a los desarrolladores obtener la ubicación del dispositivo. Android incluye un conjunto completo de herramientas que se han construido desde el inicio junto a la plataforma con los desarrolladores para proporcionar una elevada productividad y una profunda comprensión de sus aplicaciones.

## **ANDROID RUNTIME**

Android incluye un conjunto de bibliotecas básicas que proporciona la mayor parte de la funcionalidad disponible en las principales bibliotecas del lenguaje de programación Java.

Cada aplicación de Android se ejecuta con su propio proceso, con su propio ejemplo de Dalvik la máquina virtual. Dalvik se ha escrito de manera que un dispositivo pueda ejecutar varias máquinas virtuales de manera eficiente. Dalvik VM ejecuta archivos en los Dalvik ejecutables (. DEX) formato que se ha optimizado para la memoria mínima. VM es un registro de base, y ejecuta las clases compiladas por un compilador de lenguaje Java que se han transformado en el. Dex formato de los

incluidos "dx" herramienta. Dalvik VM se basa en el núcleo de Linux para la funcionalidad subyacente y el bajo nivel de gestión de memoria.

## **HERRAMIENTAS DE DESARROLLO**

"Android SDK" incluye una variedad de herramientas especialmente diseñadas para ayudar en el desarrollo de aplicaciones móviles sobre la plataforma Android. La herramienta más importante es el "Android Emulador" y las "Android Development Tools" plugin para Eclipse", pero el SDK también incluye un surtido de otras herramientas para depuración, empaquetado e instalación de aplicaciones en el dispositivo o emulador.

### **EMULADOR ANDROID**

El "Android Emulador" es un dispositivo virtual que corre en la computadora. El emulador tiene como finalidad ayudarte a diseñar y depurar tus aplicaciones en un ambiente similar al que existe en un dispositivo real. Existen distintas versiones del emulador tanto para Windows como para Mac Os como para Linux.

### **ANDROID DEVELOPMENT TOOLS PLUGIN PARA ECLIPSE IDE**

El "ADT plugin" agrega poderosas extensiones al ambiente integrado de Eclipse haciendo que la creación y depuración de las aplicaciones Android sea fácil y rápida. Si el desarrollo esta usando Eclipse, el "ADT plugin" da un increíble estímulo para el desarrollo de las aplicaciones Android. A continuación se mencionan algunas de sus características:

- Provee acceso a otras herramientas de desarrollo de Android desde el entorno de Eclipse IDE. Por ejemplo, "ADT" permite el acceso a muchas de las capacidades de la herramienta "DDMS", tales como tomar fotografías de la pantalla, administrar el redireccionamiento de puertos, fijar puntos de quiebre y examinar la información "thread" y procesos directamente dentro de Eclipse.
- Provee un asistente para la creación de proyectos Android, el cual ayuda a crear rápidamente todos los directorios y archivos necesarios para crear una nueva aplicación Android.
- Automatiza y simplifica el proceso de construcción de una aplicación.
- Provee un editor de código Android que ayuda a escribir XML válido para el archivo "AndroidManifest.xml".

### **DALVIK DEBUG MONITOR SERVICE - DDMS**

La "Dalvik Debug Monitor Service" es una herramienta integrada con la "Dalvik Virtual Machine", y permite administrar los procesos que corren en una instancia de emulador/dispositivo además de asistir en la depuración de ellos. Se puede usar esta herramienta para terminar la ejecución de un proceso, seleccionar un determinado proceso para depurar, generar reportes a partir de información de bitácoras, examinar el "heap" y la información de "thread, tomar fotografías de la pantalla del emulador/dispositivo y mucho más.

## **ANDROID DEBUG BRIDGE - ADB**

La herramienta "adb" nos permite instalar aplicaciones (archivos ".apk") en una instancia de emulador/dispositivo y acceder a una instancia de emulador/dispositivo usando comandos de línea. También podemos utilizarlo para enlazar un depurador estándar al código de una aplicación que esté corriendo en una instancia de emulador/dispositivo.

## **HERRAMIENTA ANDROID ASSET PACKAGING - AAPT**

La herramienta "aapt" permite crear archivos ".apk", los cuales contienen las imágenes binarias de tu código y recursos de tus aplicaciones.

## **ANDROID INTERFACE DESCRIPTION LANGUAGE - AIDL**

Permite generar código para una interfase de interproceso, como la que un servicio podría utilizar.

## **SQLITE3**

Esta herramienta ha sido incluida para comodidad de los desarrolladores. Provee acceso a los archivos de datos "SQLite" creados y usados por las aplicaciones Android.

## **TRACEVIEW**

Esta herramienta produce una vista gráfica del análisis de información contenida en bitácoras que puede ser generada desde una aplicación Android.

## **MKSDCARD**

Esta herramienta ayuda a crear una imagen de disco que se puede usar con el emulador, para simular la presencia de una tarjeta de almacenamiento externa (tal como una tarjeta "SD").

## **Dx**

La herramienta "dx" convierte los archivos de "bytecode" estándar (".class") en archivos "Android bytecode" (".dex").

## **ACTIVITYCREATOR**

Es un "script" que genera archivos "Ant build" que se puede utilizar para compilar aplicaciones Android. Si el proyecto se está desarrollando con Eclipse y "ADT plugin", no necesitas utilizar "activityCreator".

## **¿CÓMO EMULAR ANDROID?**

Es muy sencillo emular un teléfono con Android en un sistema operativo, sólo se debe descargar el SDK de Android, una vez descargado, se descomprime el archivo, se busca la carpeta tools que se encuentra dentro del archivo que se descargó

y se ejecuta el archivo emulador que se encuentra dentro de la carpeta tools. Siguiendo estos pasos se obtendrá un emulador completo para un dispositivo ejecutado en Android.

## CARACTERÍSTICAS DE ANDROID.

- ☀ **Framework de aplicaciones:** Habilitando para la reutilización y el reemplazo de componentes.
- ☀ **La máquina virtual Dalvik:** Optimizada para dispositivos móviles.
- ☀ **Navegador integrado:** Basado en el motor del proyecto abierto WebKit.
- ☀ **Gráficos optimizados:** Suministrados por una librería de gráficos 2D. Los gráficos 3D están basados en la especificación OpenGL ES 1.0, con soporte para aceleración gráfica por hardware (opcional).
- ☀ **SQLite:** Para estructurar el almacenamiento de datos.
- ☀ **Soporte multimedia:** Común para audio, video, imágenes, soportando varios formatos (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
- ☀ **Telefonía GSM** (Si el hardware lo soporta).
- ☀ **Bluetooth, EDGE, 3G, y WiFi** (Si el hardware lo soporta).
- ☀ **Camera, GPS, compass y accelerometer** (Si el hardware lo soporta).
- ☀ **Completo entorno de desarrollo:** Incluye un dispositivo emulador, herramientas de depuración, y un plugin para el IDE Eclipse

## ARQUITECTURA DE ANDROID

Los componentes del sistema operativo de Android, cada sección se describe en detalle a continuación:

- ☀ **Aplicaciones:** Las aplicaciones base incluyen un cliente de email, programa de SMS, calendario, mapas, navegador, contactos, y otros. Todas las aplicaciones son escritas en el lenguaje de programación Java.
- ☀ **Framework de aplicaciones:** Los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede hacer luego uso de esas capacidades (sujeto a reglas de seguridad del framework). Éste mismo mecanismo permite que los componentes sean reemplazados por el usuario. Una capa de servicios disponibles para las aplicaciones incluye:
  - Un completo y extensible conjunto de vistas que pueden ser utilizadas para desarrollar una aplicación: listas, grillas, cajas de texto, botones e incluso un web browser.

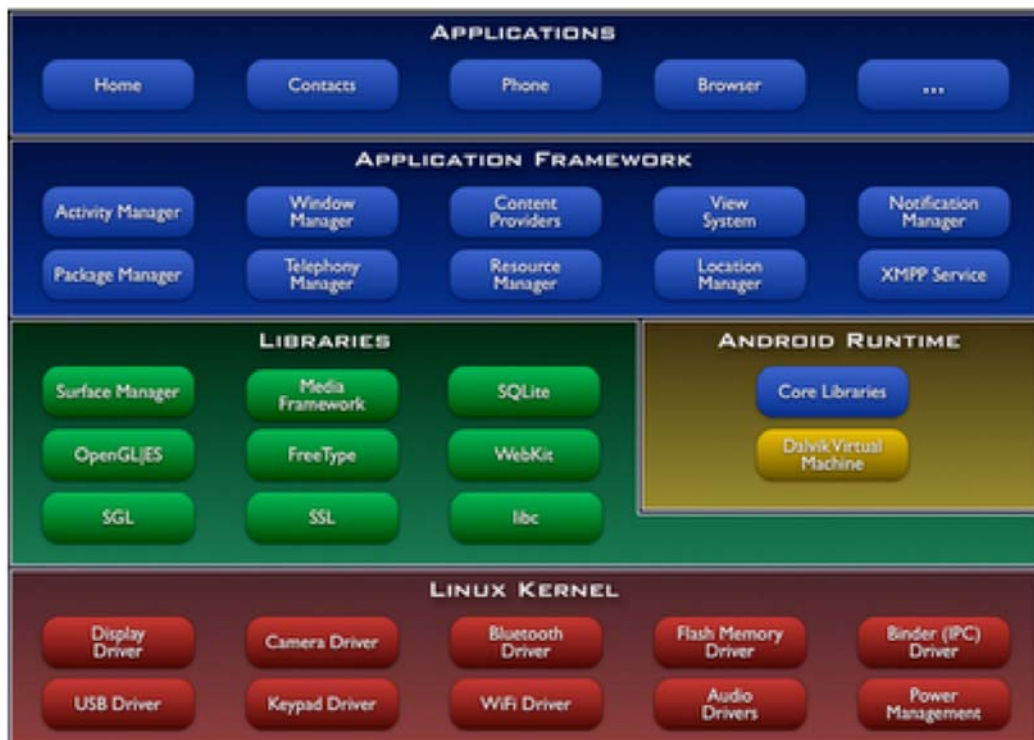
- Proveedores de contenidos que permiten el acceso a datos provenientes de otras aplicaciones (cómo Contactos), o a compartir sus propios datos.
- Un administrador de recursos, que provee acceso a recursos como cadenas, gráficos, y archivos.
- Un administrador de notificaciones que permite a todas las aplicaciones mostrar alertas personalizables en la barra de estatus.
- Un administrador de actividades que maneja el ciclo de vida de las aplicaciones y provee un comportamiento común en la navegación.

☀ **Librerías:** Android incluye un conjunto de librerías C/C++ usadas por varios componentes del sistema Android. Estas capacidades se exponen a los desarrolladores a través del framework de aplicaciones de Android. Algunas son: System C library (implementación librería C standard), librerías de medios, librerías de gráficos, 3d, SQLite, entre otras.

☀ **Runtime de Android:** Android incluye un conjunto de librerías base que proveen la mayor parte de las funcionalidades disponibles en las librerías base del lenguaje de programación Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalkiv ha sido escrito de forma que un dispositivo puede correr en múltiples máquinas virtuales de forma eficiente. Dalkiv ejecuta archivos en el formato Dalvik Executable (.dex), el cual está optimizado para memoria mínima. La Máquina Virtual está basada en registros, y corre clases compiladas por el compilador de Java que han sido transformadas al formato.dex por la herramienta incluida "dx".

☀ **Núcleo - Linux:** Android depende de un Linux versión 2.6 para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, stack de red, y modelo de drivers. El núcleo también actúa como una capa de abstracción entre el hardware y el resto del stack de software.

El siguiente diagrama muestra los principales componentes del sistema operativo Android:



## REQUISITOS DEL SISTEMA

### Android funciona bajo los siguientes sistemas operativos:

- ☀ Windows XP o Vista
- ☀ Mac OS X 10.4.8 o posterior (solo X86)
- ☀ Linux (Testeado en Linux Ubuntu Dapper Drake)

### Requisitos para poder programar en Android:

- ☀ Descargar SDK
- ☀ Descargar Eclipse
- ☀ Descargar JDK6
- ☀ Descargar el Plugin de Eclipse (ADT)

### Entornos de desarrollo

#### ☀ IDE de Eclipse:

- ❖ Eclipse 3.2, 3.3 (Europa)
  - Eclipse JDT plugin (incluido en la mayoría de los paquetes Eclipse IDE)
  - WST (opcional, pero necesario para el editor de Android, incluido en la mayoría de los paquetes Eclipse IDE).

#### ☀ Otros entornos de desarrollo o IDE:

- ❖ Apache Ant 1.6.5 o posterior para Linux y Mac, 1,7 o posterior para Windows.
- ❖ JDK 5 or JDK 6(JRE por sí solo no es suficiente).
- ❖ No es compatible con compiladores de GNU para Java (gcj).
- ❖ El plugin de herramientas de desarrollo de Android (opcional).

**Nota:** Si el JDK se ha instalado en su computadora en donde se encuentra desarrollando su proyecto Android, asegúrese de que la computadora cumpla con los requisitos enumerados anteriormente. En particular, tenga en cuenta que algunas distribuciones de Linux pueden incluir el JDK 1.4 o compiladores de GNU para Java, que no son compatibles para el desarrollo de un proyecto Android.

### DESCARGAR EL SDK

El SDK se tiene para los 3 sistemas operativos mencionados anteriormente. Además de servir para programar aplicaciones, viene con un emulador de un teléfono Android para que se pueda probar las aplicaciones y manejar un teléfono con Android desde la PC.

### DESCARGAR EL ECLIPSE

Con la ayuda de la herramienta eclipse se puede instalar el SDK de Android. Es un IDE muy bueno para programar en Java.







```
package com.google.android.hola;

import android.app.Activity;
```

## DESCARGAR JDK6 (JAVA DEVELOPMENT KIT)

Es un paquete de programación de software (SDK) para producir programas en Java. El JDK está desarrollado por la división JavaSoft de Sun Microsystem. Las versiones más recientes incluyen la arquitectura de componentes JavaBeans y soporte para JDBC.

## BIBLIOTECAS

Android incluye un conjunto de librerías de C / C + + usadas por diversos componentes del sistema Android. Estas bibliotecas están expuestas a los desarrolladores a través de la aplicación Android. Algunas de las bibliotecas principales se enumeran a continuación:

- ☀ **System C library:** Una implementación derivada del estandar de C (libc), sintonizado para Linux.
- ☀ **Media Libraries:** Basado en PacketVideo del OpenCORE; las bibliotecas de apoyo de reproducción y grabación de audio más populares y formatos de vídeo, así como archivos de imagen estática, incluyendo MPEG4, H.264, MP3, AAC, AMR, JPG y PNG.
- ☀ **Surface Manager:** Gestiona el acceso de el subsistema de pantalla y la perfección de la composición de 2D y 3D de las capas gráficas de varias aplicaciones.
- ☀ **LibWebCore:** Un moderno navegador web utilizado por Android.
- ☀ **SGL:** El subyacente motor de gráficos 2D.
- ☀ **3D libraries:** Una aplicación basada en OpenGL ES 1,0 APIs. Las bibliotecas utilizan cualquier hardware 3D (si está disponible), altamente optimizado del software 3D.
- ☀ **FreeType:** Mapa de bits y vectores de la renderización de fuentes.
- ☀ **SQLite:** Un potente y ligero motor de base de datos relacional a disposición de todas las solicitudes.

## INSTALAR EL PLUGIN DE ECLIPSE (ADT)

Si va a utilizar el IDE de Eclipse como su entorno para el desarrollo de aplicaciones de Android, puede instalar un plugin llamado **herramientas de desarrollo de Android (ADT)**, que añade soporte integrado para proyectos de Android y herramientas. El plugin ADT incluye una variedad de poderosas herramientas que hacen la creación, funcionamiento y aplicaciones de depuración de Android más

rápida y sencilla. Si no va a usar el IDE de Eclipse, no es necesario descargar o instalar el plugin de ADT.

Mucho se ha hablado sobre Android, el nuevo sistema operativo para dispositivos móviles desarrollado por Google. Todas parecen ser ventajas, empezando por la posibilidad de desarrollar aplicaciones de forma libre (se puede acceder al SDK de Android desde Google sin ningún problema). En el Mundo PDA hemos probado un emulador del sistema y lo cierto es que comparándolo en muchos aspectos (fluidez, sencillez y accesibilidad) parece que mucho va a tener que trabajar Microsoft para ponerse a la altura si no quiere perder el tren en los dispositivos portátiles. De momento algunos fabricantes ya han declarado su intención de apostar por el nuevo sistema operativo en sus próximos lanzamientos.

## **INSTALACIÓN Y CONFIGURACIÓN**

### **2. INSTALANDO JAVA**

1.1 Descargamos java desde su sitio oficial (<http://www.java.com/es/download/>) una vez abierta la página presionamos el botón descarga gratuita de java (botón en color verde) y nos lleva a otra página (imagen de abajo) donde nos da la opción de instalarlo online o guardar el archivo en la computadora, nosotros escogeremos la de instalarlo online para ello presionamos el botón que nos dice Windows XP/Vista/2000/2003 En línea y nos saldrá un dialogo que muestra la imagen, presionamos el botón guardar archivo.



1.2 Buscamos el archivo que se guardó en nuestra computadora (el lugar donde se guardó es donde lo indique en la configuración del explorador de Internet) se busca el archivo y se ejecuta, el archivo se deberá llamar algo así jre-6u3-windows-i586-p-iftw.exe. al menos tiene que coincidir la primera parte de jre y la palabra Windows si tu sistema operativo es Windows.

## 2- INSTALANDO ANDROID

2.1 Descargamos Android desde el sitio oficial (<http://code.google.com/android/download.html>) y seleccionamos el archivo como se indica en la imagen de abajo.

**Version m3-rc37a**

December 14, 2007 - [Release Notes](#)

Platform	Package	Size	MD5 Checksum
Windows	<a href="#">android_sdk_windows_m3-rc37a.zip</a>	58 MB	5db5aea20a2c2f010baefc4b1091a575
Mac OS X (intel)	<a href="#">android_sdk_darwin_m3-rc37a.zip</a>	54 MB	0b22e73fbd07b4af4009387afce3a37f
Linux (i386)	<a href="#">android_sdk_linux_m3-rc37a.zip</a>	54 MB	41285beecc4f9926e6ecf5f12610b356

For more information on the SDK:

- [Development Requirements](#)
- [Guide to Installing the SDK](#)

2.2 El archivo se descarga dentro del mismo directorio donde se ha descargado el archivo de java, se busca y lo descomprimos.

2.3 Descomprimido el archivo, este contendrá una carpeta llamada android\_sdk\_windows\_m3-rc37a (o similar dependiendo de la versión vigente). Esta carpeta se corta y se pega dentro de la unidad C de la PC.

## 3- INSTALANDO ECLIPSE

3.1 En la página oficial de Eclipse (<http://www.eclipse.org/downloads/>) se descarga el archivo indicado en la imagen siguiente:

### Eclipse Downloads

To download Eclipse, select a package below or choose one of the third party Eclipse distros. **You will need a Java runtime environment (JRE) to use Eclipse (Java 5 JRE recommended).** All downloads are provided under the terms and conditions of the [Eclipse Foundation Software User Agreement](#) unless otherwise specified.

Problems extracting the ZIP file? Please read these [Known Issues](#).

#### Eclipse Europa Fall Maintenance Packages - Windows [\(compare packages\)](#)



##### Eclipse IDE for Java Developers - Windows (78 MB)

The essential tools for any Java developer, including a Java IDE, a CVS client, XML Editor and Mylyn. [Find out more...](#)



##### Eclipse IDE for Java EE Developers - Windows (126 MB)

Tools for Java developers creating JEE and Web applications, including a Java IDE, tools for JEE and JSF, Mylyn and others. **Java 5 (or higher) required.** [Find out more...](#)



##### Eclipse IDE for C/C++ Developers - Windows (63 MB)

An IDE for C/C++ developers. [Find out more...](#)



##### Eclipse for RCP/Plug-in Developers - Windows (153 MB)

A complete set of tools for developers who want to create Eclipse plug-ins or Rich Client Applications. It includes a complete SDK, developer tools and source code. [Find out more...](#)



##### Eclipse Classic 3.3.1.1 - Windows (140 MB)

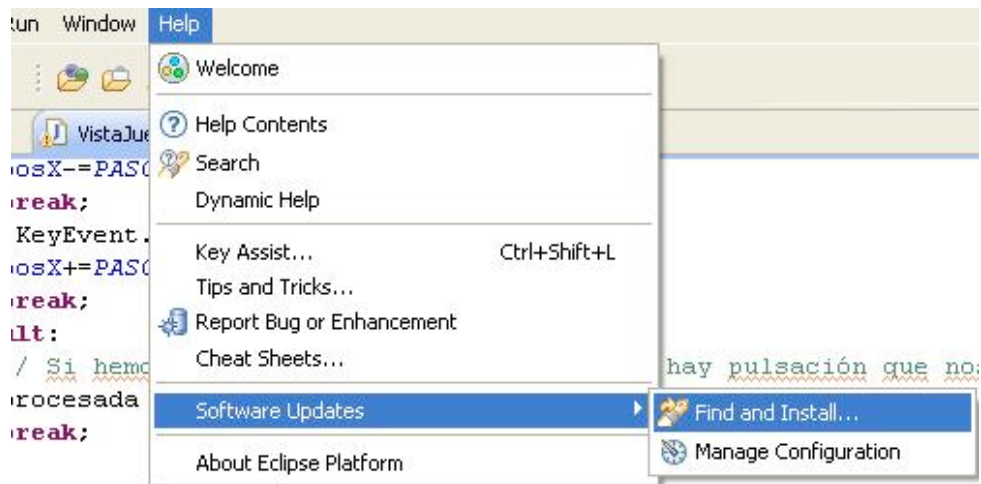
The classic Eclipse download: the Eclipse Platform, Java Development Tools, and Plug-in Development

3.2 Una vez descargado, también se encuentra comprimido en .zip, se descomprime y se guarda la carpeta contenida dentro de la unidad C de la computadora.

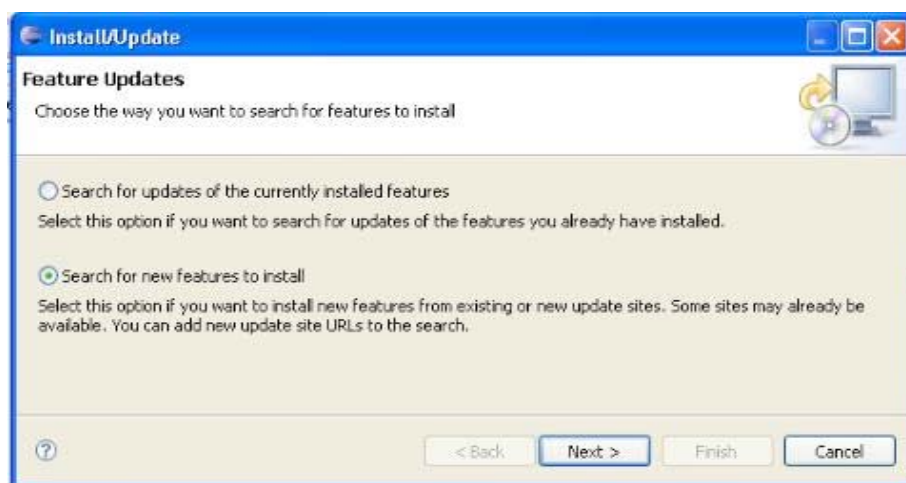
#### 4 INSTALANDO EL PLUGIN DE ANDROID PARA ECLIPSE

Ahora solo queda instalar el plugin de Android para eclipse, para realizar esto tienes que seguir los pasos siguientes:

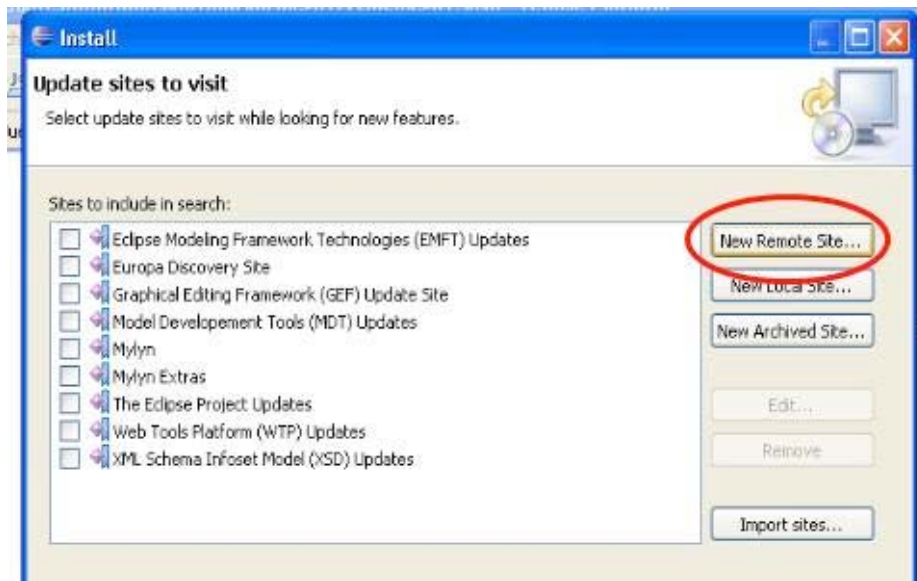
- 4.1 Ejecuta el archivo eclipse.exe, se abrirá un dialogo donde te pregunta la ruta donde guardara los proyectos que crees en eclipse. Por defecto apuntara al directorio de tu sesión de usuario (usuarios Windows). Escoge la ruta que quieras o deja esta por defecto.
- 4.2 Ahora instalaremos el plugin de Android para eclipse, una vez ejecutado eclipse nos dirigimos al menú Help / Software updates / Find and Install.



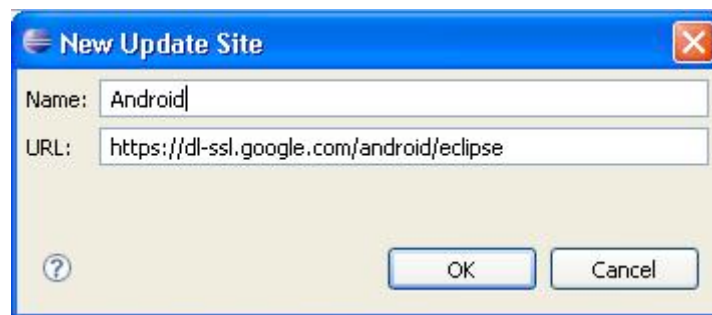
- 4.3 Se abrirá el dialogo siguiente donde seleccionaremos la opción Search for New Features to Install y presionamos next.



- 4.4 A continuación obtendremos el plugin vía Internet desde el sitio oficial de Android, para ello en el dialogo siguiente presionamos el botón New Remote Site.



4.5 Finalmente en el siguiente dialogo indicamos la url siguiente (<https://dl-ssl.google.com/android/eclipse>) donde obtendrá eclipse el plugin y le daremos un nombre para identificarlo, nosotros pondremos Android. presionamos OK.



4.6 Ahora en la lista nos tendrá que salir entre las opciones Android, seleccionamos la casilla de Android y presionamos el botón finish. Se cerrara el diálogo y se buscara el plugin en la ruta indicada, si todo sale bien se abrirá un nuevo diálogo donde se realizara la instalación del plugin solo debemos aceptar las condiciones y presionar ok. Y finalmente eclipse nos solicitara que reiniciemos el programa, le haremos caso y lo reiniciamos para poder utilizar Android correctamente.

## INSTALANDO ANDROID ECLIPSE EN UBUNTU

1.- descargar ANDROID SDK, puedes hacerlo con el navegador ([http://dl.google.com/android/android\\_sdk\\_linux\\_m3-rc20a.zip](http://dl.google.com/android/android_sdk_linux_m3-rc20a.zip)) o por consola con wget. Por tanto abrimos una consola y nos situamos en nuestro home. Descomprimos y cambiamos el nombre del directorio "android\_sdk\_linux\_m3-rc20a" por "android\_sdk".

```
user@ubuntu: ~$ wget
user@ubuntu: ~$ unzip android_sdk_linux_m3-rc20a.zip
user@ubuntu: ~$ mv android_sdk_linux_m3-rc20a android_sdk
```

2.- Finalmente para terminar la instalación tendremos que añadir el SDK al path, por

tanto editaremos el archivo `.bashrc` de nuestro `/home` con cualquier editor de texto (`vi`, `gedit`, `kate`...) y añadiremos las siguientes des líneas al final del documento:

```
user@ubuntu:~$ vi /home/user/.bashrc
```

```
export SDK_ROOT=/home/user/android_sdk
export PATH=$SDK_ROOT/tools:$PATH
```

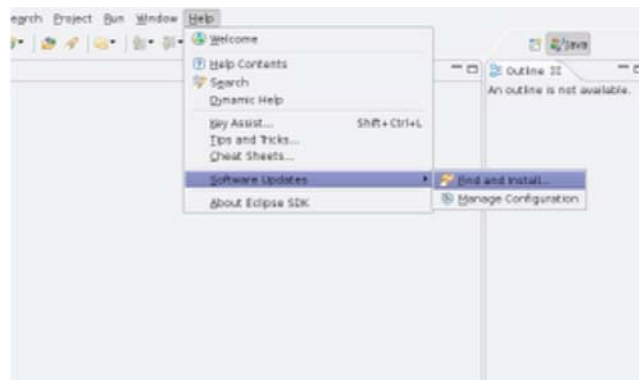
3.- Ya tenemos el SDK instalado y es posible acceder a sus aplicaciones desde consola, por tanto ahora nos queda instalar un IDE para programar cómodamente nuestras aplicaciones. Para eso el IDE idóneo es ECLIPSE ya que existen plugins para trabajar con ANDROID de forma sencilla. Por tanto, en caso de no tenerlo instalado, pasamos a instalar ECLIPSE:

```
user@ubuntu:~$ apt-get install eclipse
```

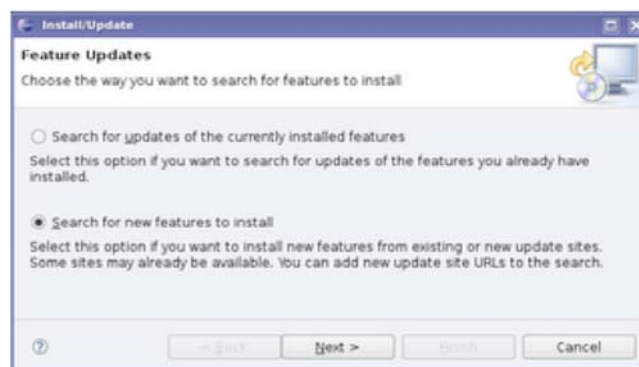
4.- Tras descargarse, se instalará automáticamente gracias a APT por lo que una vez finalizada la instalación ya podremos lanzar eclipse. En caso de que no dispongas de ningún acceso directo a la aplicación se puede acceder por consola.

```
user@ubuntu:~$ eclipse
```

5.- Una vez iniciado eclipse nos vamos al menú **Help > Software Updates > Find and Install....**

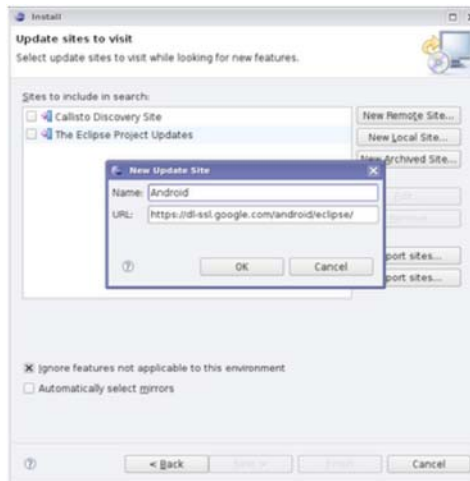


6.- Seleccionamos **Search for new features to install** y pulsamos **Next**.



7.- Pulsamos **New Remote Site** y se muestra un dialogo que pedirá una URL y un nombre para identificar el "repositorio". Por nombre podemos poner "Android" y en la URL introducimos la siguiente dirección: <https://dl-ssl.google.com/android/eclipse/>

8.- Pulsamos **OK**.

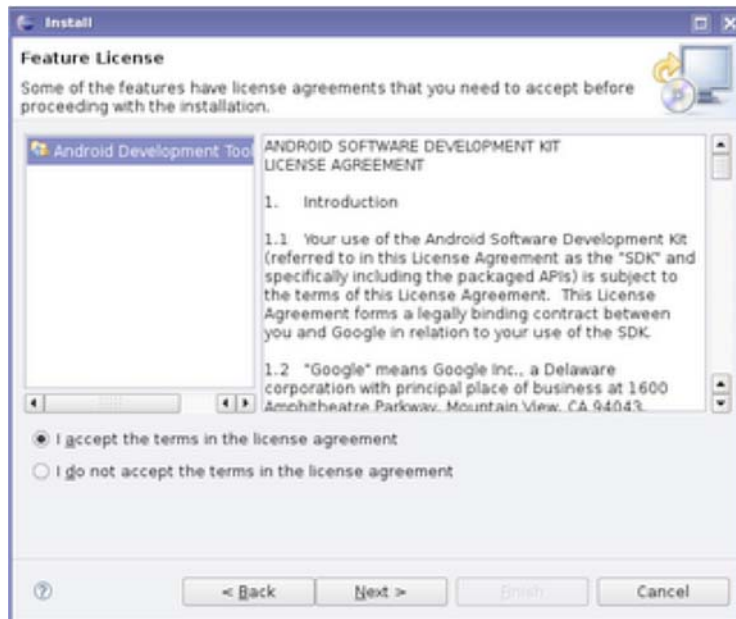


9.- Se nos añadirá un "repositorio" a la lista, lo seleccionamos y pulsamos **Finish**.

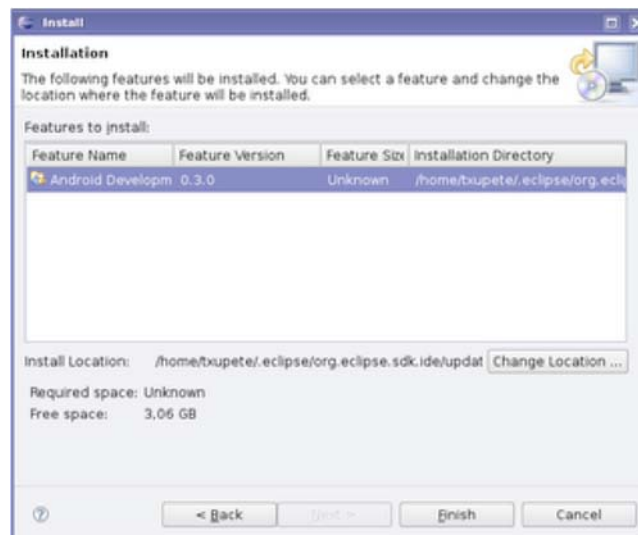
10.-Nos aparecerá ahora un árbol con la siguiente estructura: **Android Plugin > Eclipse Integration > Android Development Tools**, los seleccionamos todos y pulsamos **Next**.



11.- Aceptamos las condiciones de la licencia y pulsamos **Next**.



12.- Finalmente nos aparecerá una lista de tareas a instalar, y nos aparecerá "Android Developm..", y pulsamos **Finish**.

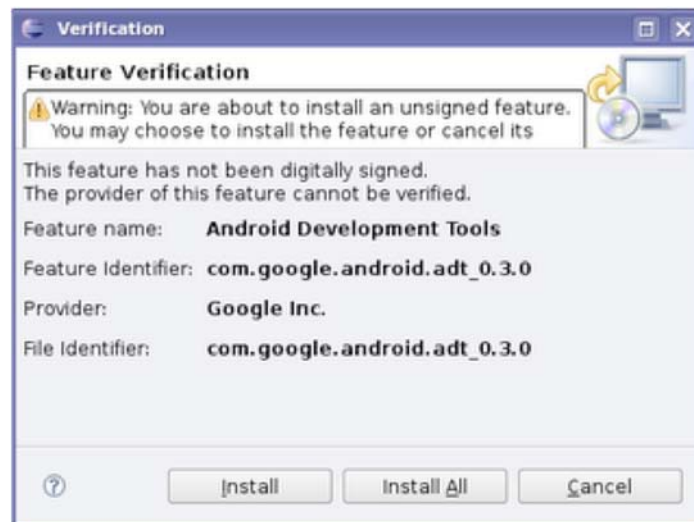




13.- El instalador descargará el paquete y lo instalará automáticamente.



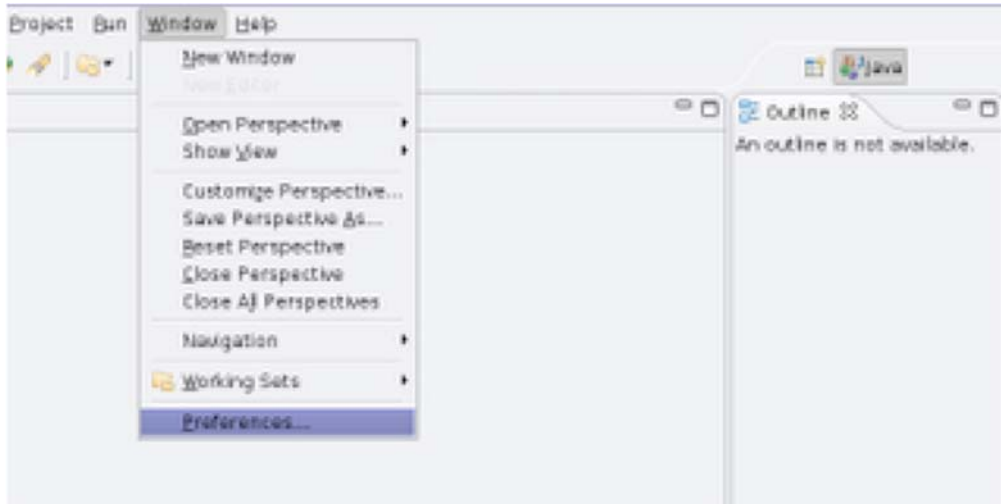
14.- El plugin ADT no está firmado por lo que debemos aceptar su instalación a pesar de ello por lo que pulsaremos **Install All**.



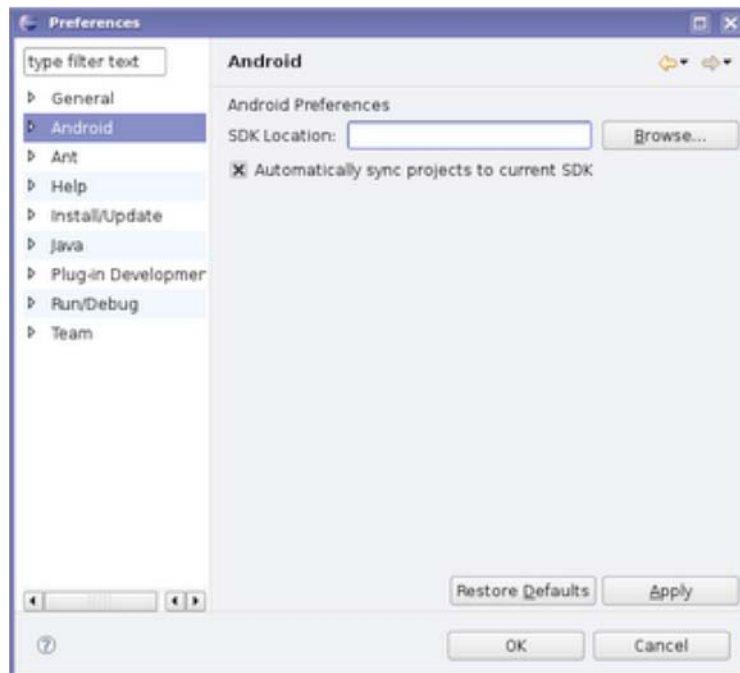
15.- Ya está el plugin completamente instalado, y solo falta reiniciar Eclipse.



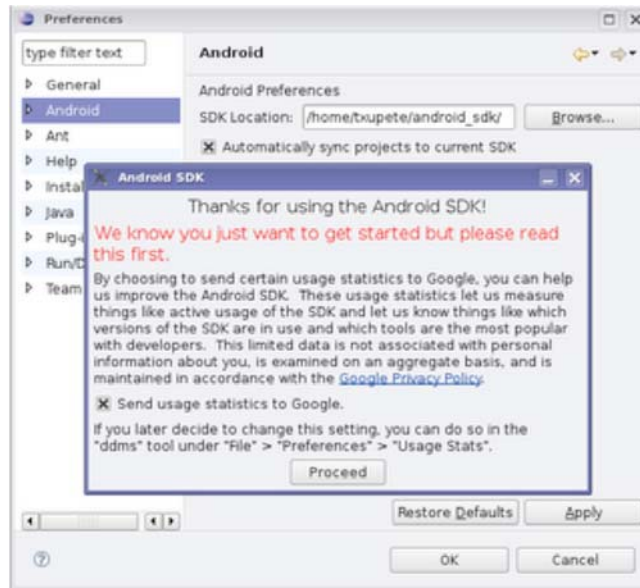
16.- Una vez reiniciado solo falta configurar el plugin indicándole la ruta al SDK, por lo tanto accedemos al menú **Window > Preferentes**



17.- Seleccionamos **ANDROID** e introducimos la ruta a nuestro SDK, que en nuestro ejemplo era **/home/user/android\_sdk/**.



18.- Finalmente pulsamos **Apply** y nos aparecerá una ventana con un aviso de que el plugin ya esta instalado y configurado correctamente.



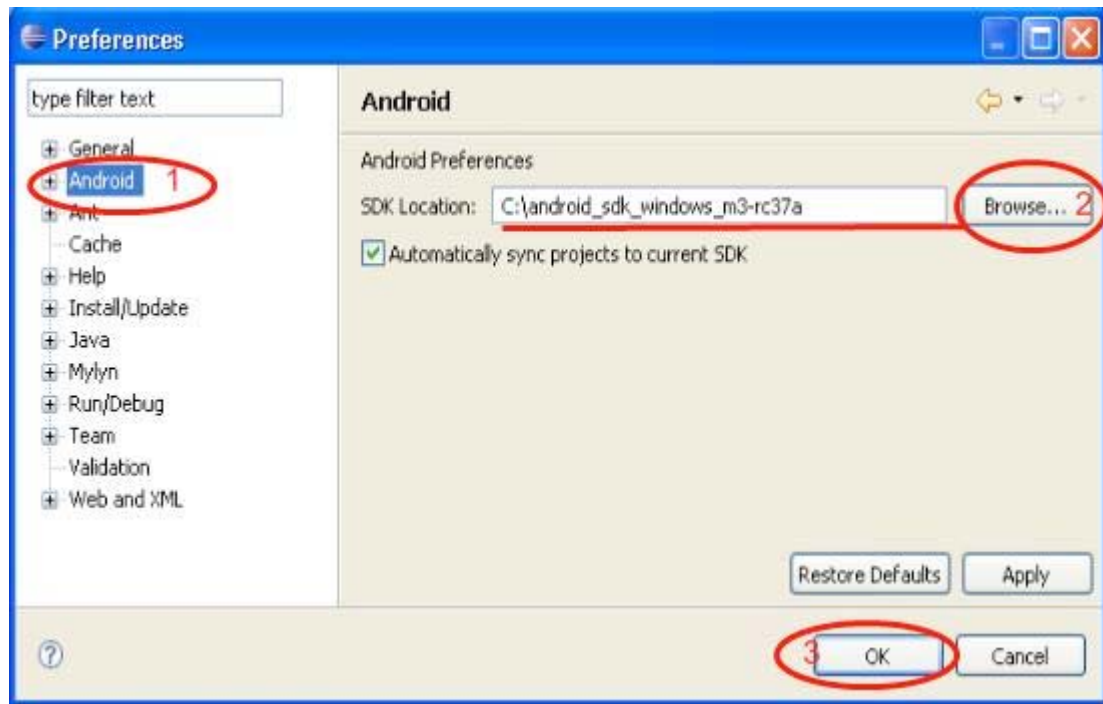
Ya tenemos nuestro entorno de desarrollo completamente instalado.

## CREAR UN NUEVO PROYECTO ANDROID

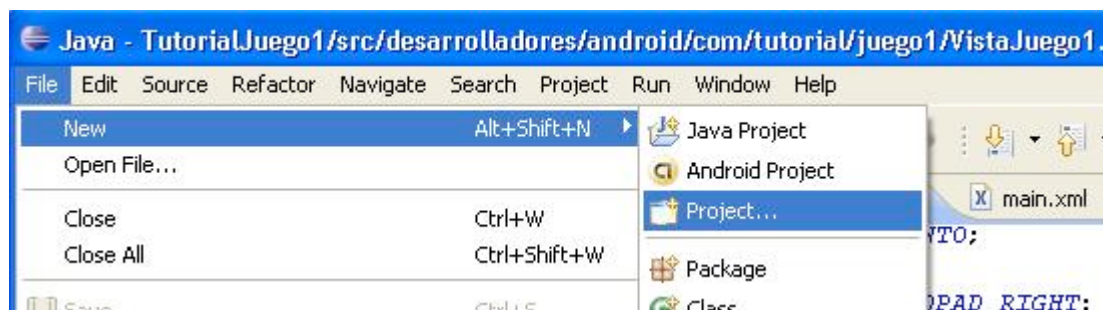
Ahora con todo instalado y todo en su sitio crearemos un nuevo proyecto Android, donde haremos una pequeña aplicación que visualice la frase "Hola Mundo" por pantalla. Sigue los siguientes pasos:

Antes que nada tendremos que hacer un pequeño ajuste en eclipse, esto consiste en añadir la referencia al sdk de android, y lo haremos de la siguiente forma.

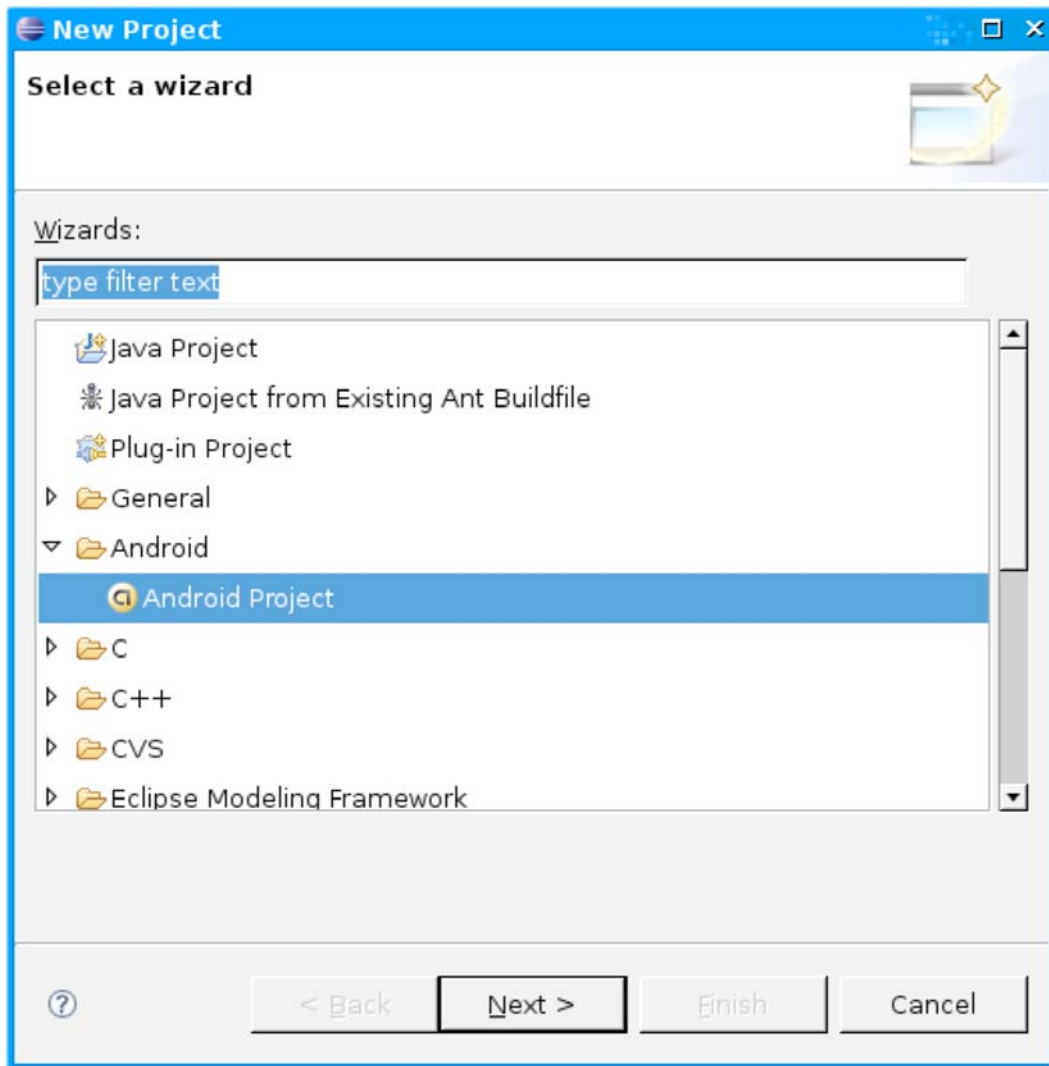
1. Nos dirigimos al menú Window / Preferences donde se nos abrirá el siguiente diálogo, donde seleccionaremos en la parte derecha Android (1), luego presionaremos el botón Browser (2) y se nos abrirá un nuevo diálogo donde buscaremos el SDK de Android (lo buscaremos en C ya que lo habíamos guardado allí) seleccionamos la carpeta y presionamos aceptar, finalmente en SDK Location nos aparecerá la ruta y nombre del directorio que contiene el sdk y presionaremos el botón OK (3).



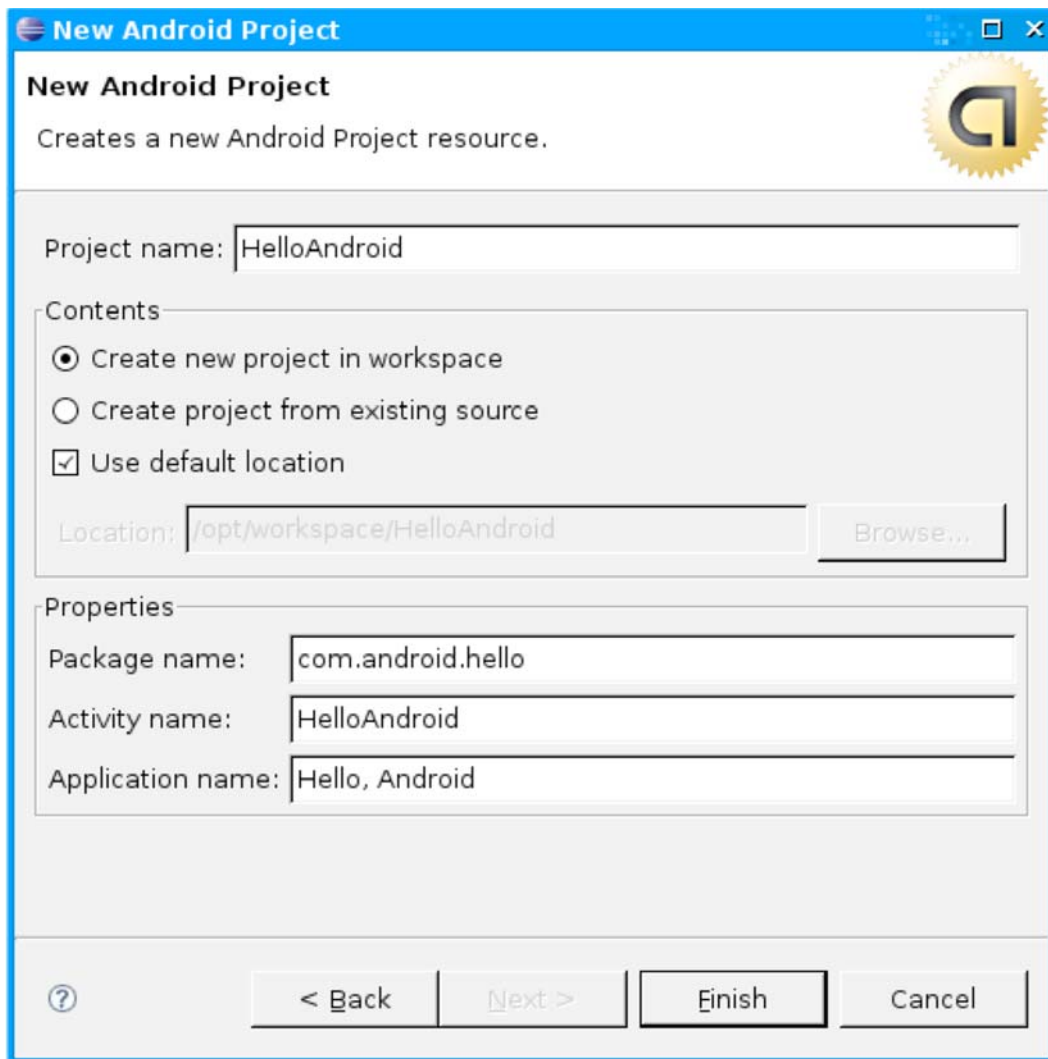
2. Ahora con la referencia establecida a Android ya estamos en condiciones de crear un nuevo proyecto en Android. Para ello nos dirigimos al menu File / New / Project, o bien File / New / Android Project.



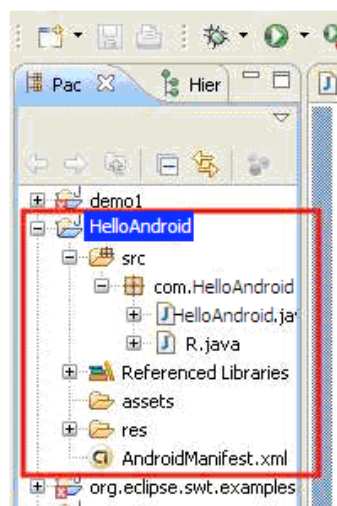
3. Si escogemos la primera opción (cuando sea la primera vez que creamos un proyecto Android no saldrá como un opción y tendremos que elegir Project como la imagen anterior). se nos desplegara el siguiente diálogo donde seleccionamos Android Project y presionamos el botón next.



4. Ahora se nos abrirá un nuevo diálogo donde tendremos que indicar 1º el nombre del proyecto, 2º el nombre del package (el nombre del package esta formada por dos palabras unidas con un punto como por ejemplo com.aplicacion) este paquete es el que contendrá todos los archivos y es obligatorio. Estos dos campos son obligatorios y no se pueden dejar en blanco, y por último (3º) el nombre que queremos darle a nuestra aplicación y el nombre del arrancador de la aplicación (seria el Main de java).



5. Finalmente si todo sale bien, se creará nuestro proyecto y se podrá visualizar desde el panel de área de trabajo (workspace) que se encuentra a la izquierda de eclipse (imagen siguiente). Dentro del proyecto si lo desplegamos presionando el botón + veremos el contenido. Primero la carpeta src la cual al igual que en j2me contendrá los archivos java y el paquete, por defecto se crea el archivo HelloAndroid.java y el R.java, el primero es el que modificaremos para que muestre Hola Mundo por pantalla. Mas abajo veremos mas carpetas y archivos, estas las explicaremos más adelante.



6. Si todo sale bien, se creara nuestro proyecto y se podrá visualizar desde el panel de área de trabajo (workspace) que se encuentra a la izquierda de eclipse (imagen siguiente). Dentro del proyecto si lo desplegamos presionando el botón + veremos el contenido. primero la carpeta src la cual al igual que en j2me contendrá los archivos java y el paquete, por defecto se crea el archivo HelloAndroid.java y el R.java, el primero es el que modificaremos para que muestre Hola Mundo por pantalla. Mas abajo veremos mas carpetas y archivos, estas las explicaremos más adelante.

```
Package com.HelloAndroid;  
import android.app.Activity;  
import android.os.Bundle;  
public class HelloAndroid extends Activity {  
    /** Called when the activity is first created. */  
    @Override  
    public void onCreate(Bundle icle) {  
        super.onCreate(icle);  
        setContentView(R.layout.main);  
    }  
}
```

7. Modificamos el código anterior por este otro:

```
package com.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. */  
    @Override  
    public void onCreate(Bundle icle) {  
        super.onCreate(icle);  
        TextView tv = new TextView(this);  
        tv.setText("Hola Android");  
        setContentView(tv);  
    }  
}
```

8. Finalmente para ejecutar el programa presionamos el botón Run o desde el menú Run, recuerda tener seleccionado el package HelloAndroid para que ejecute tu aplicación HelloAndroid y no lance ningún error. Antes de ejecutar te preguntará que tipo de aplicación es, tu debes seleccionar Android. Si todo sale bien se abrirá una ventana con un móvil enorme donde te visualizara por la pantalla la palabra "Hola Android".



## **REFERENCIAS BIBLIOGRAFICAS**

- <http://www.tufuncion.com/android-movil>
  - <http://www.ubuntips.com.ar/2007/11/13/emula-un-movil-con-android-en-tu-ubuntu/>
  - <http://code.google.com/android/>
  - <http://code.google.com/android/what-is-android.html>
- Fecha de acceso: 3 de mayo de 2008