

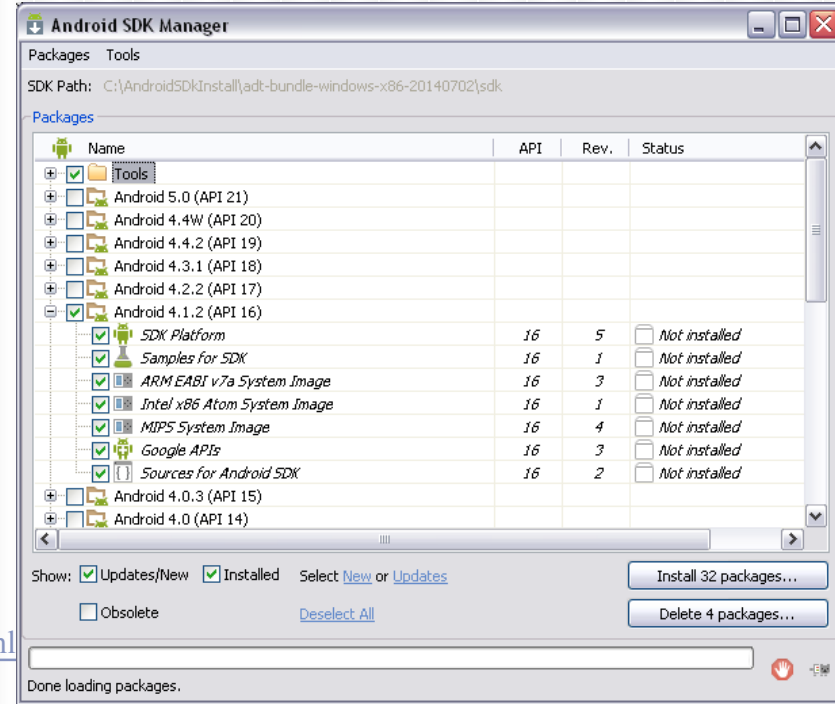
Instalación Entorno Desarrollo

Crear una Aplicación Android

Instalar entorno de Desarrollo

1. Bajar el SDK Android y Eclipse
<https://developer.android.com/sdk/index.html#download>
2. Ejecutar el archivo bajado e instalar el SDK.
<https://developer.android.com/sdk/installing/index.html?pkg=adt>
3. Entrar en la SDK Manager para Actualizar las herramientas (Tools) a última versión

de Seleccionar: Tools, Extras y Versión/es de Android a Desarrollar.

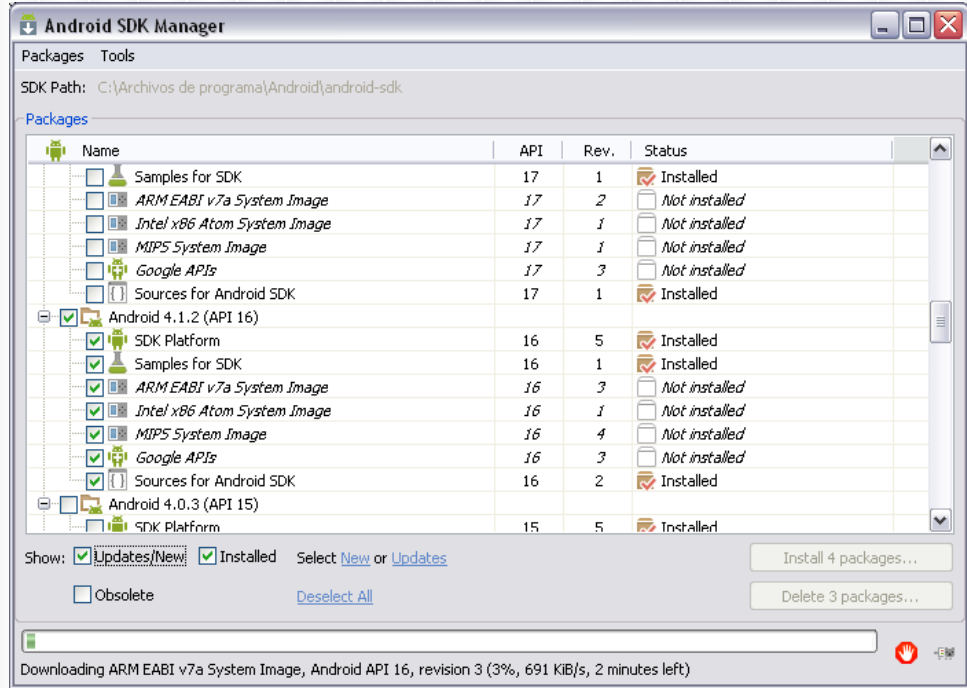


Instalar Entorno de Desarrollo

Importante:

Como mínimo Seleccionar:

1. Android SDK Tools
2. Android SDK Platform-tools
3. Android SDK Build-tools (por ahora la versión más reciente)
4. Una o más versiones de la plataforma Android (Elegir según lo que se va a desarrollar).
5. Android Support Repository (extras)
6. Google Play Services (extras)
7. Google Repository (extras)



Instalar Entorno de Desarrollo

Cómo Instalar los Paquetes

Una vez seleccionado todos los paquetes, continuar con instalar:

1. Click en **Install X packages**.
2. En la ventana siguiente, aceptar la licencia de aceptación.
3. Click **Install**.

El progreso de la bajada de paquetes se muestra en la parte inferior de la pantalla.

Instalar Entorno de Desarrollo

4. Bajar el JDK Java Developer Kit

<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html?ssSourceSiteId=otnes>

Para que funcione la Virtual Machine de Java la **Variable PATH de windows** tiene que incluir la ruta donde se encuentra el archivo javaw.exe (C:\Archivos de Programa\Java\bin\)

Si al abrir el Eclipse les aparece “Error de Virtual Machine”

(Solución → Editar archivo **eclipse.ini** y sacar últimas sentencias: **-Xms512m y -Xmx1024m**)

Más información en:

<http://www.aprendeandroid.com/YAF/default.aspx?g=posts&t=190>

Instalar entorno de Desarrollo

Ahora a Construir algo!

Con los paquetes instalados ya esta listo el SDK de Android.
Se puede comenzar a crear una aplicación.

Simplemente como continuamente hay nuevas herramientas y APIs disponibles, simplemente hay que abrir el SDK Manager para descargar los nuevos paquetes del SDK.

Eclipse con ADT

Android Developer Tools (Herramientas de Desarrollo Android)

- **ADT (Android Developer Tools) es un plugin para Eclipse** que proporciona un conjunto de herramientas que se integran con el IDE de Eclipse.
- Ayudan a desarrollar aplicaciones para Android de forma rápida porque ofrece acceso a muchas funciones.
- ADT ofrece acceso guiado a las interfaces de usuario de muchas de las herramientas del SDK de línea de comandos.
- Permite la creación rápida de prototipos, diseño y construcción de interfaz de usuario de la aplicación. con la herramienta de diseño de interfaz de usuario.

Eclipse con ADT



- ADT + Eclipse = un IDE muy completo: con muchas funciones junto con las características específicas de Android.
- Eclipse permite realizar en el Proyecto Android:
 - la creación,
 - la construcción,
 - el empaquetado,
 - la instalación y
 - la depuración
- ADT integra integra muchas de las tareas de flujo de trabajo de desarrollo en Eclipse, por lo que es fácil desarrollar rápidamente y probar tus aplicaciones de Android.

Características Eclipse + ADT

- **La integración de las Herramientas del SDK:** Muchas herramientas del SDK están integradas en los menús, las vistas, o como parte de los procesos que se ejecutan de fondo por el ADT.
- **Lenguaje de programación Java y editores XML:** El editor de lenguaje de programación Java contiene características comunes en el IDE, tales como la comprobación de sintaxis al momento de compilar el código fuente, auto-completado del código, y la documentación integrada para las APIs del marco de Android. ADT también ofrece editores de XML personalizados que le permiten editar archivos XML específicos de Android en una interfaz de usuario basada en formularios. Un editor gráfico le permite diseñar interfaces de usuario con una interfaz de arrastrar y soltar (drag-and-drop).
- **Documentación integrada para APIs Framework Android:** Se puede acceder la documentación de métodos, clases o variables. Y en [Página de Cambios Recientes](#), los cambios y nuevas características del Framework.

Editores de Eclipse

- **Editor para Layout Gráfico:** Permite editar y diseñar los archivos XML de layout desde una interfaz drag-and-drop. El editor ofrece una previsualización mientras diseñamos los layouts de las pantallas. Es invocado cuando se abre un archivo xml que tiene declarado una vista (generalmente declarado en `res/layout`.) Para más información, ver [Graphical Layout Editor](#).
- **Editor del archivo Manifiesto del Proyecto Android:** Permite editar el archivo con una interfaz gráfica simple. Es invocado cuando uno abre un archivo `AndroidManifest.xml`.
- **Editor del Menú:** Permite editar los ítems de menú y grupos de ítems desde una interfaz gráfica. Es invocado cuando se abre un archivo XML con un tag “<menu>” (carpeta `res/menu`).
- **Editor de Recursos:** Edita los recursos con una simple interfaz gráfica. Es invocado cuando se abre un archivo XML que contiene un tag “<resources>”.
- **Editor de Recursos XML:** Edita los recursos XML interfaz gráfica simple. Es invocado cuando se abre un archivo XML.

Eclipse - Integración SDK

Integración con las Herramientas del SDK

Muchas de las herramientas que se pueden iniciar o ejecutar desde la línea de comandos están integrados en el ADT. Se incluye:

- **Traceview:** Permite crear perfiles de ejecución de su programa (Ventana> Abrir perspectiva> Traceview).
- **Android:** Proporciona acceso al Administrador de Android SDK y AVD Manager. Y a otras características de Android como la creación o actualización de los proyectos (aplicación y biblioteca) se integran dentro del IDE de Eclipse.
- **Visor de Jerarquías:** Permite visualizar jerarquía de vistas de la aplicación para encontrar ineficiencias (Ventana> Abrir perspectiva> Visor de jerarquía).

Eclipse - Integración SDK

- **Pixel Perfect:** Permite examinar de cerca su interfaz de usuario para ayudar con el diseño y la construcción. (Ventana> Abrir perspectiva> Pixel Perfect).
- **DDMS:** Proporciona características de depuración incluyendo: captura de pantalla, el hilo de ejecución (thread) y la información de contexto (heap), y de Logcat (Ventana> Abrir perspectiva> DDMS).
- **ADB:** Proporciona acceso a un dispositivo de su sistema de desarrollo. Algunas características de adb se integran en ADT como la instalación del proyecto (menú de ejecución de Eclipse), transferencia de archivos, la enumeración de dispositivos, y Logcat (DDMS). Usted debe acceder a las funciones más avanzadas de ADB, como comandos del shell y de la línea de comandos.
- **ProGuard:** Permite revisión de código permitiendo la reducción y optimización. ADT integra ProGuard como parte de la construcción, si se lo activa.

Eclipse - Manejo Recursos

ADT ofrece mejoras para saltar rápidamente a las declaraciones de los distintos tipos de recursos, tales como cadenas o archivos de diseño.

Se puede acceder, manteniendo pulsada la tecla de control y haciendo clic en los diferentes ítems:

- **Un elemento**, como **R.id.button1**, abre la definición XML de la vista. Una declaración en el archivo **R.java**, como "public static final int Button01 = 0x7f050000", salta a la **definición XML correspondiente**.
- **Una actividad o definición de servicio** en el archivo AndroidManifest.xml, como **<activity android:name=".TestActivity">**, salta a la **clase Java correspondiente**. Se puede navegar por la definición de actividad (o la definición de servicio) en la clase correspondiente.
- Se puede saltar a cualquier **definición de valor** (por ejemplo **@string:foo**), independientemente del **archivo XML "foo"** donde se haya definido.

Eclipse - Manejo Recursos

- Cualquier **declaración basada en archivos**, como por ejemplo **@layout/bar**, abre el archivo.
- Los **recursos no-XML**, como **@drawable/icon**, abre la aplicación de Eclipse **definida** por defecto para el tipo de archivo correspondiente, que en este caso es una imagen.
- Recursos del **namespace @android**, abre los **recursos que se encuentran instalados del SDK**.
- Un **atributo XML** como **@android:string/ok** or **android.R.string.id** en código Java abre el **archivo que declara las cadenas (strings)**. Se abre una pestaña XML cuando se hace esto, no el editor.

Instalar entorno de Desarrollo

Obtener la “Android Support Library” para APIs adicionales

La Android Support Library provee un set extendido de APIs que son compatibles con la mayoría de las versiones de Android.

Open the **Extras** directory and select:

- **Android Support Repository**
- **Android Support Library**

La Support Library es requerida para:

- **Android Wear** : Para dispositivos pequeños y potentes usados en el cuerpo. Herramientas para acceder a temas relacionados con el fitness.
- **Android TV**
- **Google Cast**: es una tecnología que permite generar experiencias multi-screen, que permite a los usuarios enviar contenido desde dispositivos móviles y computadoras personales a sus TVs. Se puede usar Google Cast Android API para castear la funcionalidad de las aplicaciones Android y ver el contenido en la pantalla grande.

Instalar entorno de Desarrollo

Obtener servicios de Google Play

Para desarrollar con APIs de Google, es necesario bajar el paquete de servicios Google Play:

Abrir SDK Manager, abrir directorio **Extras** y seleccionar:

- **Google Repository**
- **Google Play services**

Nota: APIs de Servicios Google Play no están disponibles en todos los dispositivos Android, pero están disponibles para todos los dispositivos que usen Google Play Store. Para usar estas APIs en el emulador Android, hay que instalar las **APIs** del sistema **Google** de la última versión del directorio Android X.X en el SDK Manager.

Las APIs de servicios Google Play provee una variedad de funciones y servicios para las aplicaciones Android, como ser:

- Autenticación de Usuario
- Google Maps
- Google Cast
- Juegos