

# Anexo IV - Configuración del Entorno de Desarrollo

Guía de puntos de interés de la Ciudad de Madrid

## 1. Índice

<b>Anexo IV - Configuración del Entorno de Desarrollo .....</b>	<b>1</b>
1. Índice .....	2
2. Entorno de Desarrollo.....	3
Eclipse.....	3
Los proyectos en Eclipse.....	3
Perspectivas en Eclipse .....	3
Instalación y configuración.....	3
Android SDK .....	3
Instalación y configuración.....	4
Integración del Android SDK en Eclipse .....	5
Android VIRTUAL DEVICE .....	6
Perspectiva DDMS .....	9
Ejecutando la aplicación en AVD o terminal real.....	10
3. Google Maps .....	12
3.1 Huella digital de certificado MD5.....	12
3.2 Obtención del Maps API Key .....	13

## 2. Entorno de Desarrollo

### Eclipse

---

[Eclipse](#) es un entorno de desarrollo integrado, de código abierto y multiplataforma.

La plataforma Eclipse se utiliza frecuentemente para crear entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse).

El entorno de desarrollo integrado (IDE) de Eclipse emplea módulos (también llamados plug-ins) para proporcionar toda su funcionalidad. Cualquier desarrollador puede crear nuevos plug-ins para la realización de funcionalidades específicas.

La definición que da el [proyecto Eclipse](#) acerca de su software es: "una especie de herramienta universal - un IDE abierto y extensible para todo y nada en particular".

De esta forma, Eclipse proporciona al programador frameworks muy ricos para el desarrollo de aplicaciones: gráficas, definición y manipulación de modelos de software, aplicaciones web, etc. Por ejemplo, GEF (Graphic Editing Framework - Framework para la edición gráfica) es un módulo de Eclipse para el desarrollo de editores visuales que pueden ir desde procesadores de texto hasta editores de diagramas UML, interfaces gráficas para el usuario, etc.

#### Los proyectos en Eclipse

El entorno Eclipse organiza sus desarrollos en **proyectos**. Podríamos definir un proyecto Eclipse como un conjunto de software relacionado, y que puede enmarcarse en una tipología concreta (proyectos Web, proyectos J2EE, proyecto Java). Adicionalmente, Eclipse introduce el concepto de **workspace** o espacio de trabajo, que permite ubicar en un marco común proyectos de distintas tipologías.

#### Perspectivas en Eclipse

Un proyecto puede contener diferentes tipos de recursos. Las perspectivas se utilizan a modo de filtro para visualizar únicamente la información que nos interesa obtener en cada momento y ocultar otra que sea irrelevante. Existen diferentes perspectivas predefinidas que vienen de serie con el producto, pero también pueden crearse perspectivas nuevas. Un ejemplo de perspectiva puede ser la perspectiva CVS, que muestra al usuario los depósitos CVS configurados en su entorno de desarrollo local.

#### Instalación y configuración

La configuración consiste simplemente en descomprimir el fichero en el directorio dónde se desea instalar.

### Android SDK

---

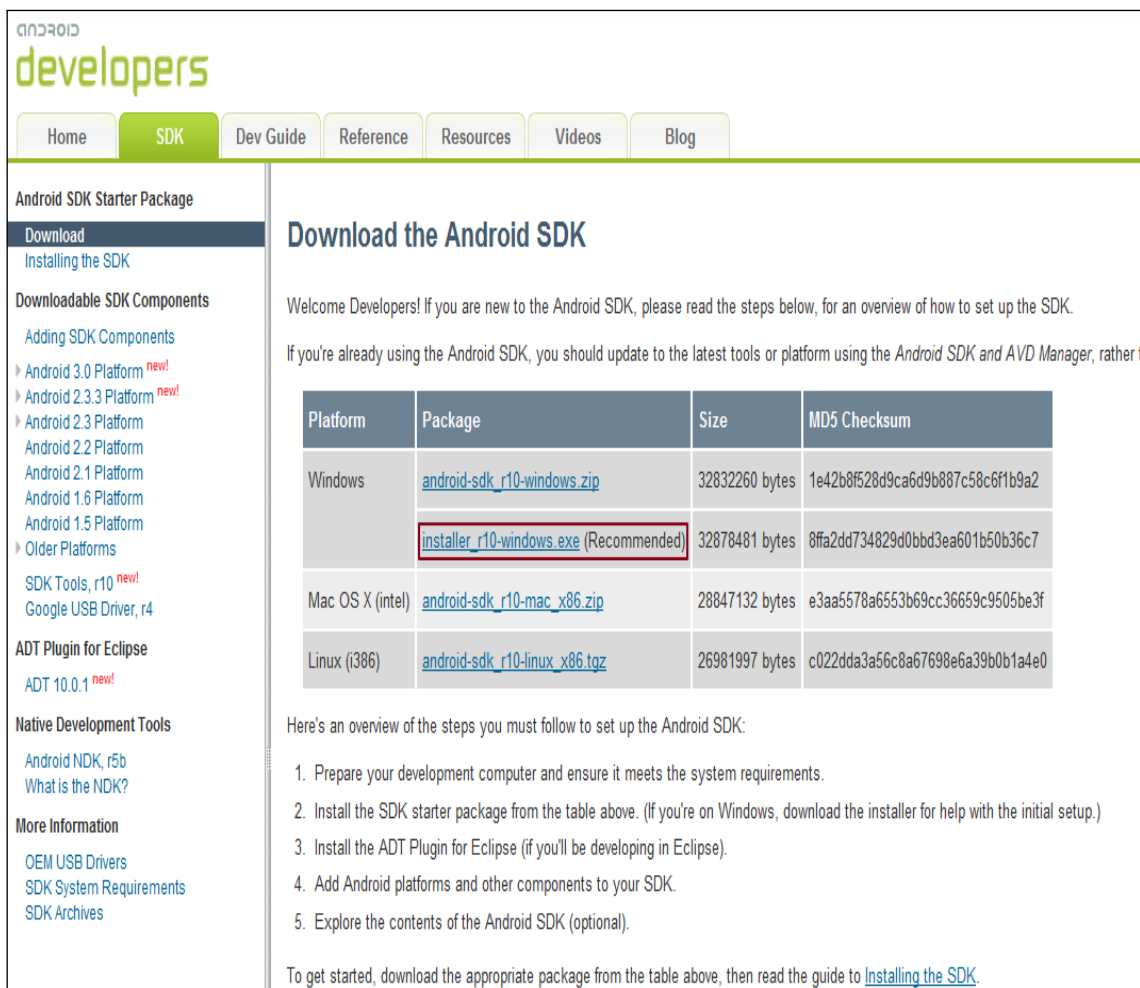
[Android SDK](#), es el kit desarrollo estándar de Android, necesario para crear aplicaciones que funcionen sobre esta plataforma.

En el kit de desarrollo Android podemos encontrar todas las herramientas que los programadores necesitan: entornos de desarrollo y depuración, librerías, un emulador de móviles, documentación o tutoriales.

El Android SDK funciona en sistemas Windows XP o Vista, Mac OS X 10.4.8 o superior y en Ubuntu 6.06 o superior, también es necesario Java 1.5. Está preparado para la integración con el entorno de desarrollo Eclipse mediante plugins.

Tal y como se observa en la ilustración 1, se ha elegido la versión r10 del SDK, recomendada en el momento de configuración del entorno.

Todas las instrucciones necesarias para la instalación se encuentran en la misma web (<http://developer.android.com/index.html>), que se constituye en la referencia de cabecera para el desarrollador.



The screenshot shows the 'Download the Android SDK' page on the Android Developers website. The page features a navigation menu with 'SDK' selected, a sidebar with various links, and a main content area with a table of download links. The 'installer\_r10-windows.exe (Recommended)' link is highlighted with a red box.

Platform	Package	Size	MD5 Checksum
Windows	<a href="#">android-sdk_r10-windows.zip</a>	32832260 bytes	1e42b8f528d9ca6d9b887c58c6f1b9a2
	<a href="#">installer_r10-windows.exe (Recommended)</a>	32878481 bytes	8ffa2dd734829d0bbd3ea601b50b36c7
Mac OS X (intel)	<a href="#">android-sdk_r10-mac_x86.zip</a>	28847132 bytes	e3aa5578a6553b69cc36659c9505be3f
Linux (i386)	<a href="#">android-sdk_r10-linux_x86.tgz</a>	26981997 bytes	c022dda3a56c8a67698e6a39b0b1a4e0

Ilustración 1 - Fuente principal para desarrolladores en Android

## Instalación y configuración

Una vez obtenido el software, el siguiente paso será instalarlo. Para ello, ejecutaremos el instalador descargado. Este instalador cuenta con un ayudante que nos guiará durante el proceso.

## Integración del Android SDK en Eclipse

El [plugin ADT](#) para eclipse permite utilizar las funcionalidades del SDK integradas en el entorno Eclipse. Existen varias formas de instalación de ADT, hemos elegido la siguiente:

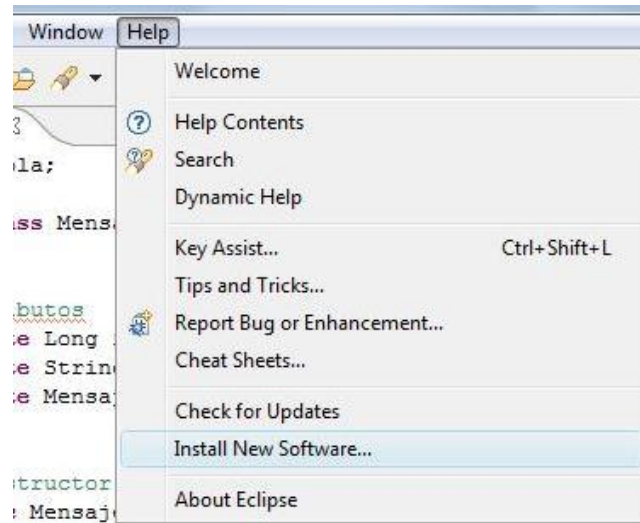


Ilustración 2 - Instalar nuevo software

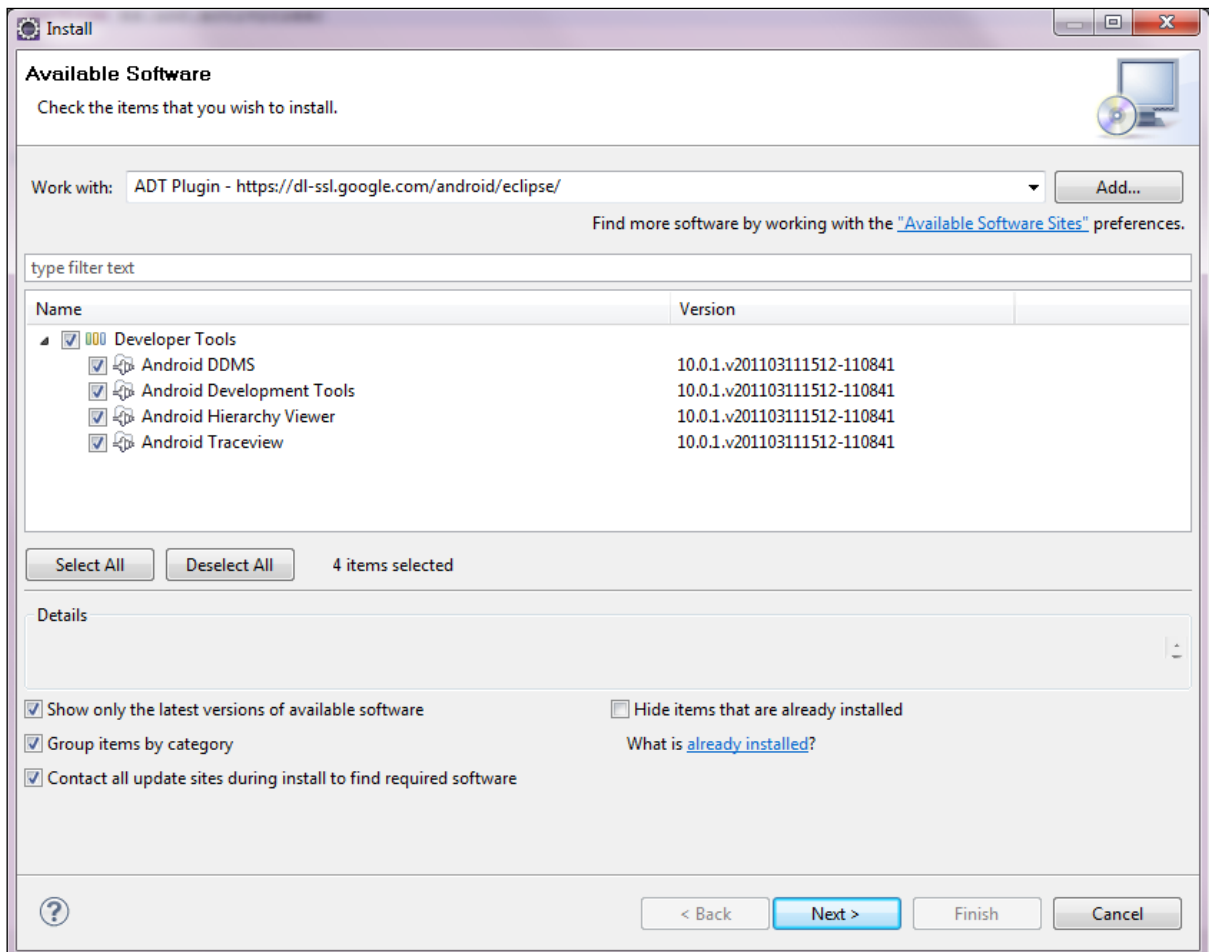


Ilustración 3 - Instalación de ADT en Eclipse

Al finalizar la instalación podremos observar la aparición de una nueva tipología de proyecto (**proyecto Android**), así como nuevas perspectivas (**DDMS**) y opciones (**AVD**). En **Preferences** de Eclipse contaremos con un nuevo apartado dedicado a Android.

Para finalizar el proceso, debemos informar a Eclipse de la ubicación en la que se encuentra Android SDK. Para ello debemos ir a la opción **Window → Preferences → Android**, y cumplimentar el campo **SDK Location**, tal y como se muestra en la Ilustración 4.

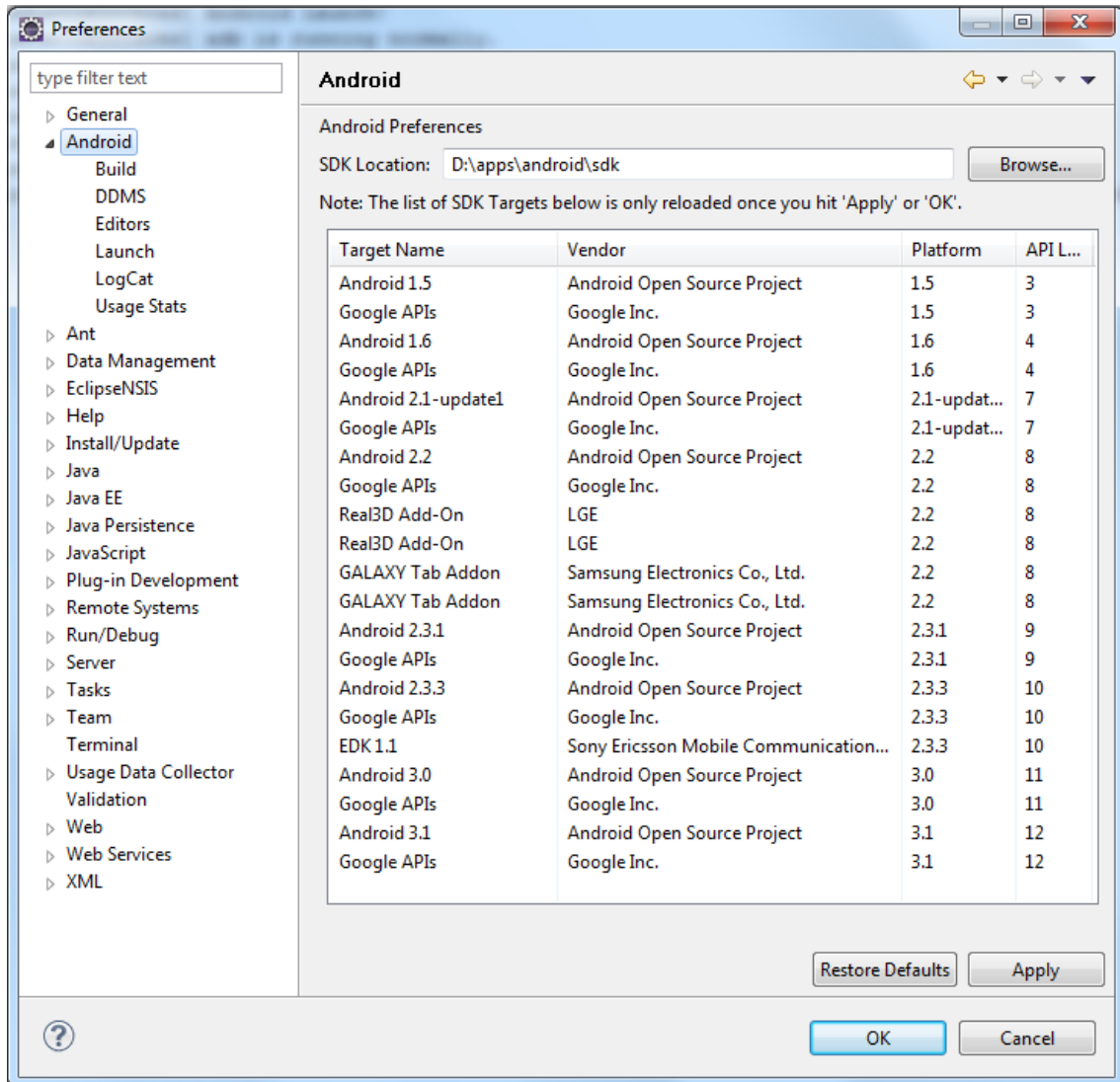


Ilustración 4 - Configuración del SDK en Eclipse

## Android VIRTUAL DEVICE

El **Android Virtual Device** (en adelante AVD) es un emulador de terminales Android. Se utiliza para pruebas y es distribuido como parte del Android SDK. Podremos acceder a él mediante el nuevo icono aparecido en la barra de herramientas de Eclipse tras la instalación del ADT.



Ilustración 5 - Acceso a AVD

Será necesario al menos un AVD para ejecutar las aplicaciones. La forma de crearlo es la siguiente:

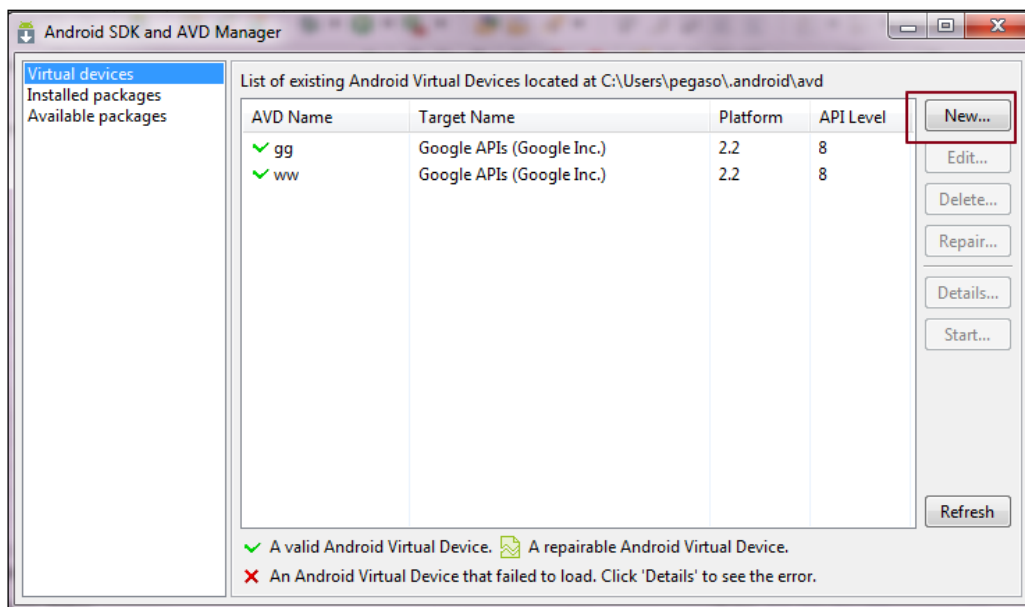


Ilustración 6 - Pantalla principal del AVD Manager

Pulsando "New" accedemos a la pantalla de creación del AVD, mostrada en la ilustración 8.

En esta pantalla podremos configurar todas las características del AVD. Estas características van desde el tamaño y definición de la pantalla, al soporte de tarjetas SD, el acelerómetro o el soporte para GPS.

Una característica muy importante del emulador es el **target**. El **target** está constituido por:

- Versión de Android para la que se va a desarrollar
- Nivel de los APIs de Google

Para el desarrollo de este proyecto se ha optado por la plataforma **Android 2.2 (FROYO)** debido a su gran implantación en el mercado y por ser una versión muy estable, adoptada por la mayoría de fabricantes actualmente. En la ilustración 8 se ofrece una comparativa entre la [implantación de versiones Android en el mercado](#).

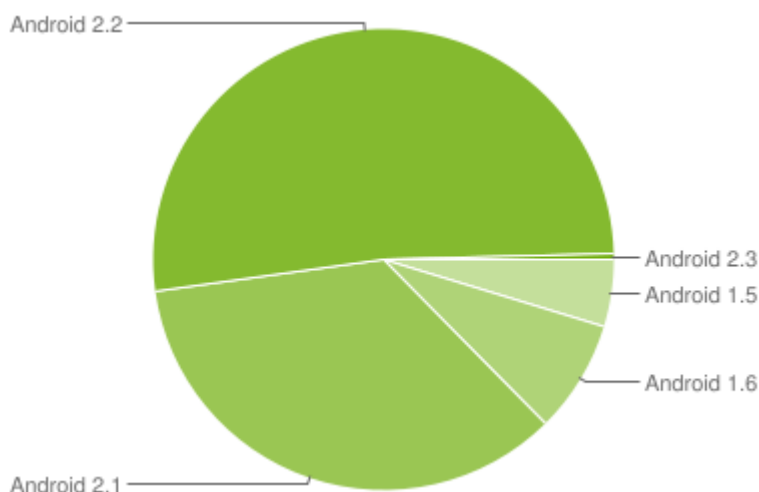


Ilustración 7 - Reparto de cuotas de mercado entre versiones Android - Diciembre 2010

En el apartado **Skin** podemos configurar aspecto del teléfono y resolución de la pantalla del **AVD**. Existen diversos Skins, descargables a través de la Web, de teléfonos disponibles en el mercado, de esta forma obtendremos una percepción más exacta de cómo se visualizará el software en un terminal determinado. Pueden cargarse Skins descargados desde la opción **Built-in** del apartado Skin.

En la opción **Resolution** podemos seleccionar diferentes resoluciones. Existen diversos tipos de [resoluciones para terminales móviles](#).

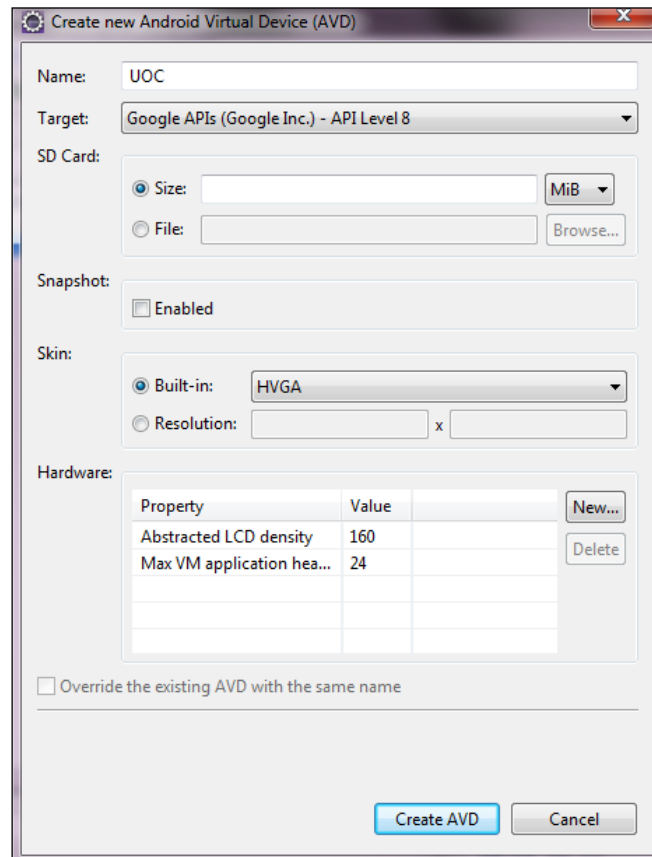


Ilustración 8 - Creación AVD

Además, añadiremos al AVD el soporte para GPS, ya que lo utilizaremos en algunas funcionalidades de la Guía.

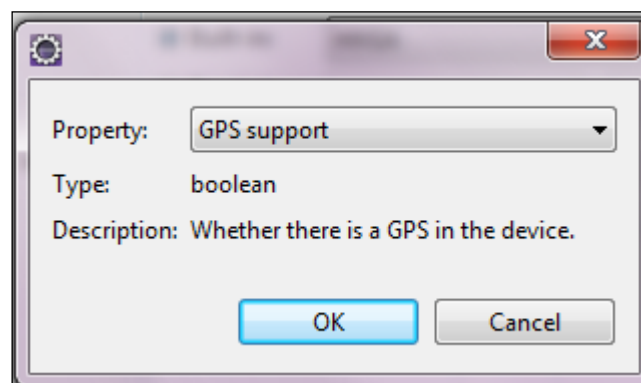


Ilustración 9 - Soporte para GPS



## Perspectiva DDMS

Desde esta perspectiva de Eclipse podremos manejar los AVDs que hayamos creado. La perspectiva se distribuye en varias pestañas, tal y como se muestra en la ilustración 10.

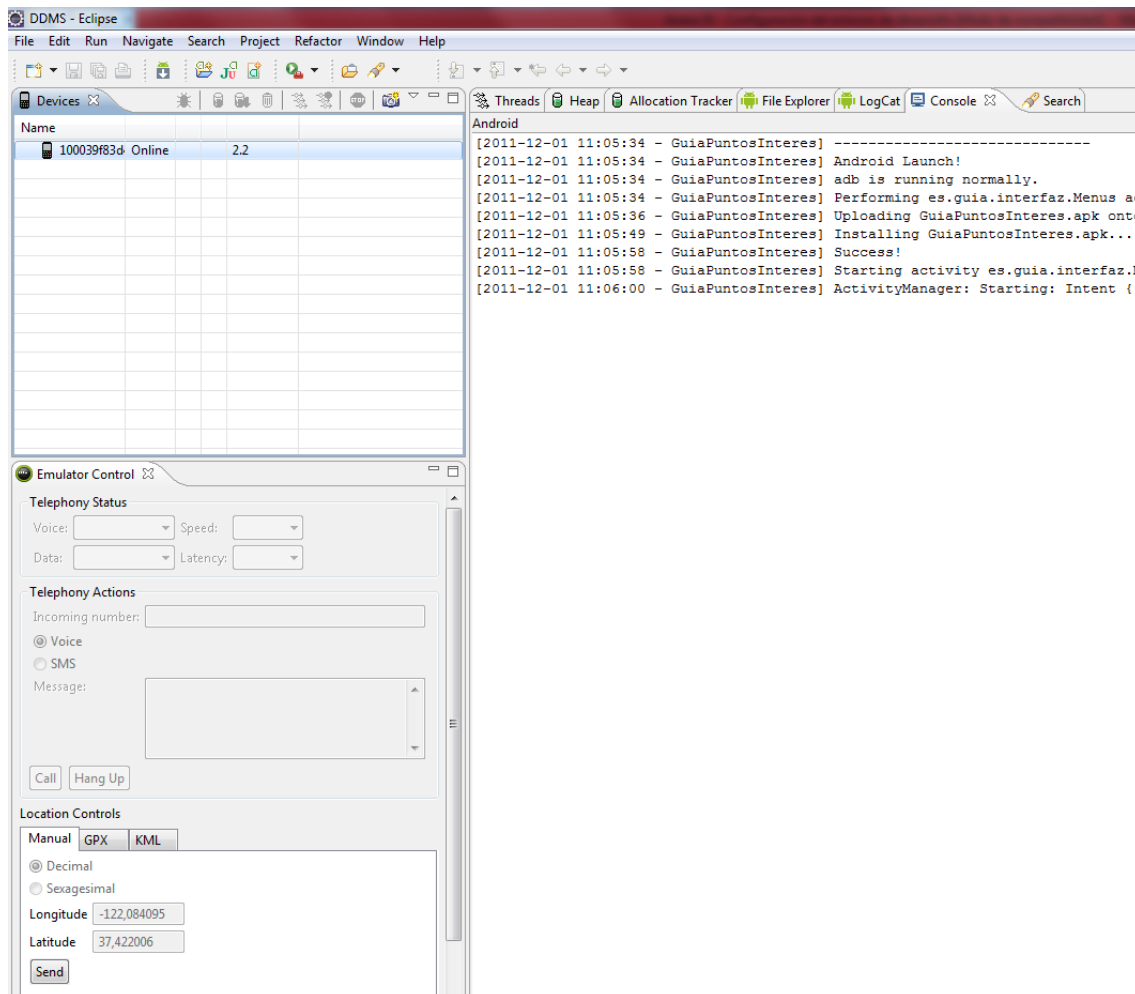


Ilustración 10 - Perspectiva DDMS

En la pestaña **Devices** aparecerán todos los AVDs (o terminales reales) arrancados en el sistema. Marcando uno de ellos podremos modificar sus características en tiempo de ejecución. Los elementos a manejar más importantes son los siguientes:

- Las coordenadas del dispositivo: Podemos manejarlas desde la pestaña **Emulator Control**.
- Los ficheros existentes en el dispositivo: Podemos gestionarlos desde la pestaña **File Explorer**. Esta pestaña nos permite introducir ficheros o descargarlos desde el terminal.
- Consola del sistema: **LogCat** es la consola del sistema. Podremos escribir en ella mensajes desde la aplicación. Estos mensajes están clasificados en niveles (debug, info, error...). Esta opción es muy útil para el proceso de depuración.

## Ejecutando la aplicación en AVD o terminal real

La ejecución de la aplicación puede realizarse desde la perspectiva Java de Eclipse, tal y como se muestra a continuación.

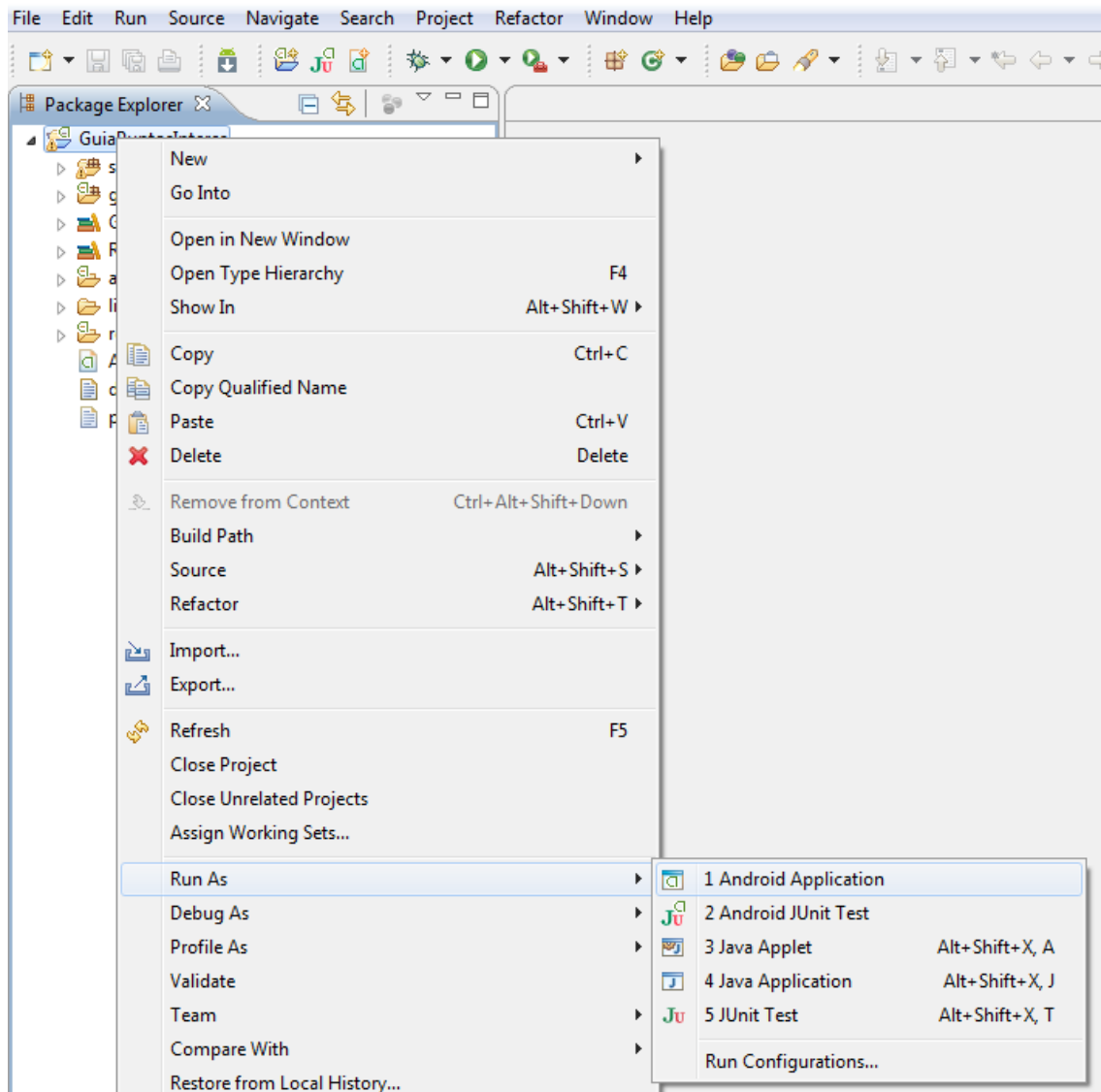


Ilustración 11 - Ejecución de la aplicación

Haciendo click sobre la opción *Android Application* aparecerá la pantalla de selección de dispositivo, mostrada en la ilustración 12.

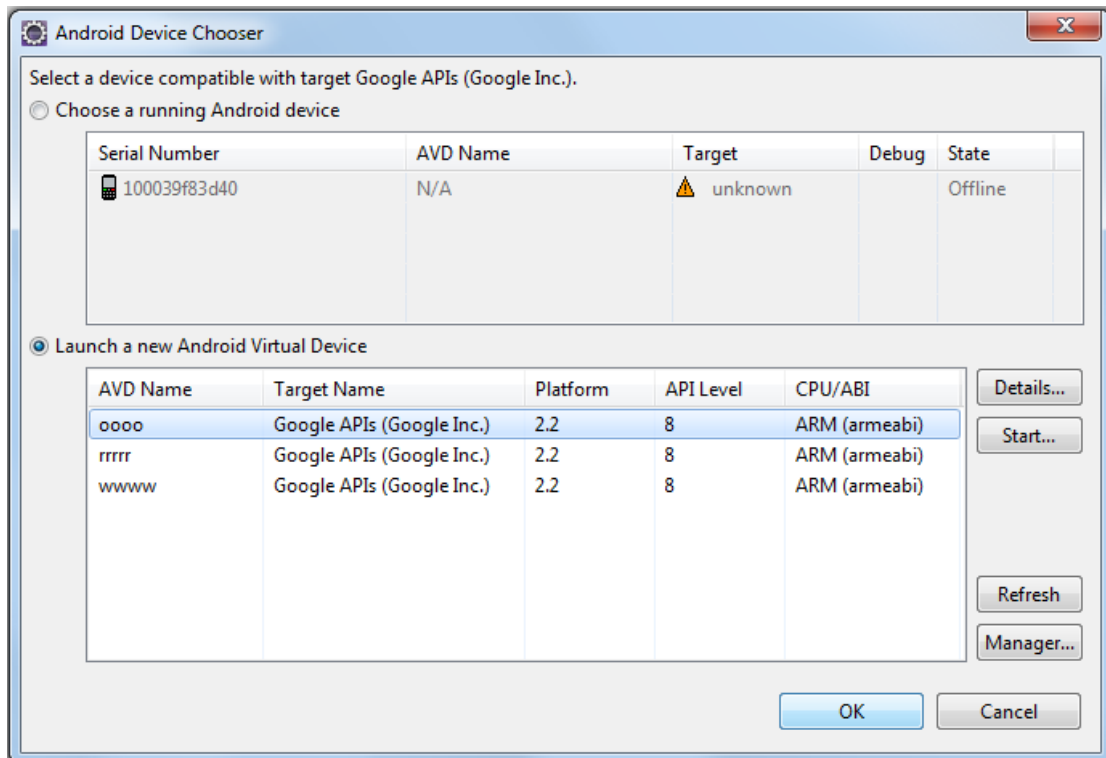


Ilustración 12 - Selección de dispositivo

La pantalla está dividida en dos partes. En la parte superior podremos seleccionar los dispositivos Android en ejecución, detectados por el sistema, en la parte inferior aparecerán todos los AVDs registrados pero que no se encuentran ejecución. Seleccionamos el terminal deseado y pulsamos OK.

El proceso de instalación y arranque de la aplicación en el terminal elegido se realizará automáticamente.



Ilustración 13 - Inicialización de AVD

### 3. Google Maps

Tal y como figura en los requisitos de usuario, la guía contará con algunas funciones propias de un Sistema de Información Geográfico. Una de estas funcionalidades consiste en localizar al usuario en un mapa a través de las coordenadas geográficas facilitadas por el GPS de su terminal. Esta funcionalidad está implementada utilizando los servicios proporcionados por Google Maps.

En este apartado se explican las acciones de configuración necesarias para el uso de estos servicios.

#### 3.1 Huella digital de certificado MD5

En primer lugar, y como paso previo a la configuración, será necesaria la obtención de un certificado MD5. Para ello puede utilizarse la herramienta **keytool**, disponible en cualquier instalación estándar de JDK (*Java Development Kit*). Mediante **keytool** generaremos un almacén de claves y una entrada en él, tal y como se muestra en la siguiente figura.

```
C:\>keytool -genkey -keystore almacen.keystore -alias UOC -storepass dani0
¿Cuáles son su nombre y su apellido?
 [Unknown]: Daniel Cortes
¿Cuál es el nombre de su unidad de organización?
 [Unknown]: UOC
¿Cuál es el nombre de su organización?
 [Unknown]: UOC
¿Cuál es el nombre de su ciudad o localidad?
 [Unknown]: Madrid
¿Cuál es el nombre de su estado o provincia?
 [Unknown]: Madrid
¿Cuál es el código de país de dos letras de la unidad?
 [Unknown]: ES
¿Es correcto CN=Daniel Cortes, OU=UOC, O=UOC, L=Madrid, ST=Madrid, C=ES?
 [no]: y

Escriba la contraseña clave para <UOC>
<INTRO si es la misma contraseña que la del almacén de claves>

C:\>keytool -list -keystore almacen.keystore
Escriba la contraseña del almacén de claves: dani0

Tipo de almacén de claves: jks
Proveedor de almacén de claves: SUN

Su almacén de claves contiene entrada 1

uoc, 28-nov-2011, keyEntry,
Huella digital de certificado <MD5>: 12:37:BE:E9:83:6D:3F:FA:EB:9D:DB:8D:09:1A:E
7:F8

C:\>
```

Ilustración 14 - Ejemplo de utilización Keytool

Otra opción válida es utilizar el almacén de claves que viene por defecto cuando instalamos el Android SDK. Para ello, primero deberemos localizar el lugar en el que se encuentra este almacén. Podemos localizarlo en **Window → Preference → Android → Build**.

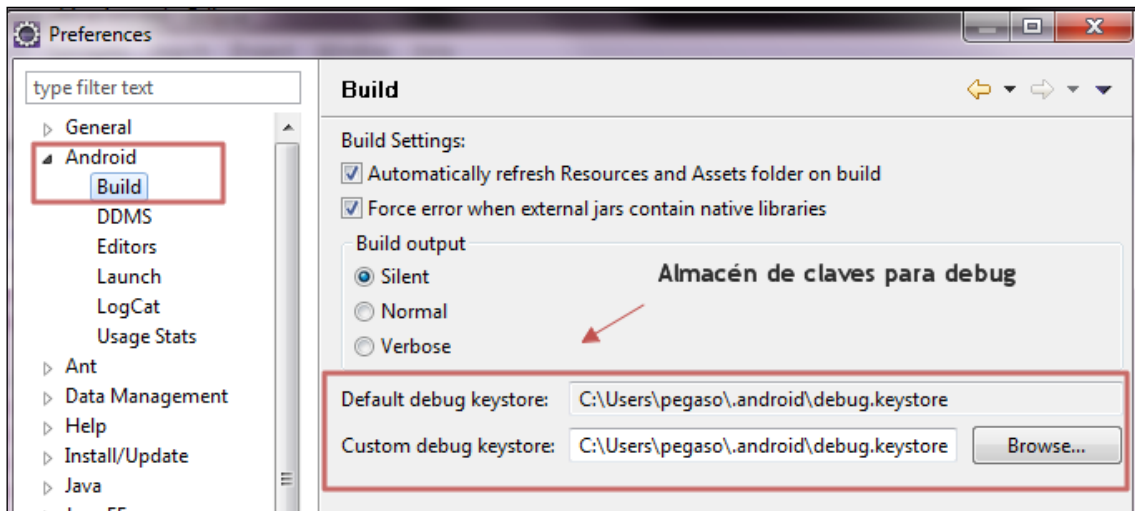


Ilustración 15 - Localización del debug.keystore

Situándonos en el directorio indicado, podemos obtener la huella MD5 del certificado, tal y como se muestra en la figura siguiente.

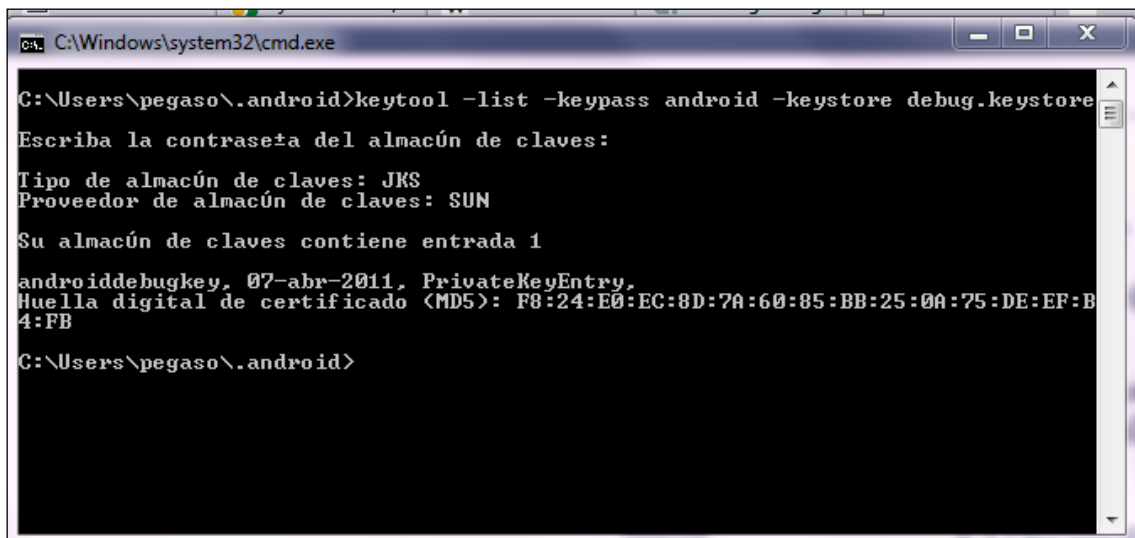


Ilustración 16 - Listado de certificados de un keystore

### 3.2 Obtención del Maps API Key

Una vez obtenida la huella del certificado, accederemos a la página de **google maps** destinada a la obtención de la key necesaria para manejar mapas: <http://code.google.com/intl/es-ES/android/maps-api-signup.html>

Esta página cuenta con un recuadro en el que pegaremos la huella MD5 y, tras pulsar el botón de generación, obtendremos un key para Maps. El proceso puede observarse en las ilustraciones 17 y 18.

code.google.com/intl/es-ES/android/maps-api-signup.html

Google Code

Search

Android Maps API Key Signup

Home Docs Maps Blog Android Blog

### Sign Up for the Android Maps API

The Android Maps API lets you embed [Google Maps](#) in your own Android applications. A single Maps API key is valid for all applications signed by a single certificate. See this [documentation page](#) for more information about application signing. To get a Maps API key for your certificate, you will need to provide its the certificate's fingerprint. This can be obtained using Keytool. For example, on Linux or Mac OS X, you would examine your debug keystore like this:

```

$ keytool -list -keystore ~/.android/debug.keystore
...
Certificate fingerprint (MD5): 94:1E:43:49:87:73:BB:E6:A6:88:D7:20:F1:8E:B5:98

```

If you use different keys for signing development builds and release builds, you will need to obtain a separate Maps API key for each certificate. Each key will only work in applications signed by the corresponding certificate.

You also need a [Google Account](#) to get a Maps API key, and your API key will be connected to your Google Account.

Android Maps APIs Terms of Service

Last Updated: October 13, 2008

Thanks for your interest in the Android Maps APIs. The Android Maps APIs are a collection of services (including, but not limited to, the "com.google.android.maps.MapView" and "android.location.Geocoder" classes) that allow you to include maps, geocoding, and other content from Google and its content providers in your Android applications. The Android Maps APIs explicitly do not include any driving directions data or local search data that may be owned or licensed by Google.

- Your relationship with Google.
  - Your use of any of the Android Maps APIs (referred to in

I have read and agree with the terms and conditions ([printable version](#))

My certificate's MD5 fingerprint: 12:37:BE:E9:83:6D:3F:FA:EB:9D:DB:8D:09:1A:E7:F8

Generate API Key

Ilustración 17 - Petición de Maps API Key

## API de Google Maps

[Página principal de Google Code](#) > [API de Google Maps](#) > Suscripción al API de Google Maps

### Gracias por suscribirte a la clave del API de Android Maps.

Tu clave es:

0-cw0fiGv77oZoWsMoU03sN5miyCp2y2DN3klzw

Esta clave es válida para todas las aplicaciones firmadas con el certificado cuya huella dactilar sea:

12:37:BE:E9:83:6D:3F:FA:EB:9D:DB:8D:09:1A:E7:F8

Incluimos un diseño xml de ejemplo para que puedas iniciarte por los senderos de la creación de mapas:

```

<com.google.android.maps.MapView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:apiKey="0-cw0fiGv77oZoWsMoU03sN5miyCp2y2DN3klzw"
/>

```

Consulta la [documentación del API](#) para obtener más información.

Ilustración 18 - Respuesta Maps API Key

La utilización del API Key generado será tratada en el capítulo 10 de la memoria final, en el apartado *Manejo de MapViews*.