**Introducció a serveis web en la plataforma Android**

**Ajuntament de Santa Coloma de Gramenet**

**Juliol 2013**

Taula de Continguts

[1 Introducció 6](#_Toc360922150)

[1.1 Finalitat del document 6](#_Toc360922151)

[1.2 Estructura del document 6](#_Toc360922152)

[2 Instal·lació de l’entorn de desenvolupament 7](#_Toc360922153)

[2.1 Descàrrega de l’entorn integrat 7](#_Toc360922154)

[2.2 Instal·lació de l’entorn de desenvolupament 7](#_Toc360922155)

[3 Annex A: Creació d’un dispositiu virtual 16](#_Toc360922156)

Taula de Figures

**No se encuentran elementos de tabla de ilustraciones.**

# Introducció

En el “Concurs de programació d’aplicacions mòbils” proposat per l’Ajuntament de Santa Coloma de Gramenet una de les possibles vies de disseny serà fer l’aplicació compatible amb la plataforma Android.

Aquest sistema operatiu per smartphones i tablets disposa de gran quantitat d’informació a internet. Al web oficial de la plataforma (<http://developer.android.com/>) es pot descarregar l’entorn de desenvolupament anomenat SDK (Software Developer Kit) i consultar la documentació de l’API, a més d’aspectes relacionats amb el disseny il·lustrats amb exemples i codi font.

## Finalitat del document

Un dels objectius d’aquest document és proporcionar una guia ràpida d’instal·lació de l’entorn de desenvolupament per poder realitzar aplicacions per la plataforma Android.

Altre és fer una introducció de com dissenyar una aplicació en Android que sigui capaç de connectar-se a un servei web. En aquest cas, es donarà una aplicació d’exemple que es connectarà al servei web del “Concurs de programació d’aplicacions mòbils” (d’ara en endavant servei web AppColoma) i que retornarà els idiomes disponibles en els que es poden trobar les fitxes de l’agenda. El programa de mostra és un client soap.

## Estructura del document

El document s’estructura en dos parts.

* Instal·lació de l’entorn de desenvolupament.
* Descripció d’un exemple d’aplicació que és capaç de connectar-se al servei web AppColoma.

# Instal·lació de l’entorn de desenvolupament

S’explicaran dues formes d’instal·lar l’entorn de desenvolupament Android a l’ordinador. Una d’elles és ben senzilla, només s’ha de descarregar l’arxiu comprimit subministrat del web oficial de la plataforma i l’altre serà útil per qui tingui ja un entorn de desenvolupament Java com Eclipse i vulgui continuar programant amb ell.

## Descàrrega de l’entorn integrat

Com ja s’ha dit és la forma més fàcil d’aconseguir un entorn de desenvolupament d’aplicacions per la plataforma Android. Aquest entorn s’anomena ***SDK ADT Bundle*** i està disponible pels sistemes operatius més utilitzats.

En el següent enllaç <http://developer.android.com/sdk/index.html> podem trobar a la part dreta un botó amb el títol de ***Download the SDK ADT Bundle for Windows*** i immediatament es produirà la descàrrega de l’arxiu comprimit.

En el cas de disposar d’un altre sistema operatiu a l’ordinador s’ha d’anar a l’apartat ***DOWNLOAD FOR OTHER PLATFORMS*** i a la secció ***ADT Bundle*** es pot trobar l’entorn integrat per diferents sistemes.

Un cop descarregat l’arxiu comprimit s’extreu la informació en una carpeta i llavors l’entorn de desenvolupament estarà preparat per fer-lo servir. Per iniciar el programa s’ha d’entrar a la carpeta anomenada **eclipse** i fer doble click sobre la icona d’aplicació que també s’anomena **eclipse**.

Un cop s’ha executat l’entorn de desenvolupament únicament cal preparar l’entorn de simulació, és a dir, crear un dispositiu virtual que permeti provar les aplicacions dintre de l’ordinador. Aquest procés es mostra a l’apartat **Annex A: Creació d’un dispositiu virtual**.

## Instal·lació de l’entorn de desenvolupament

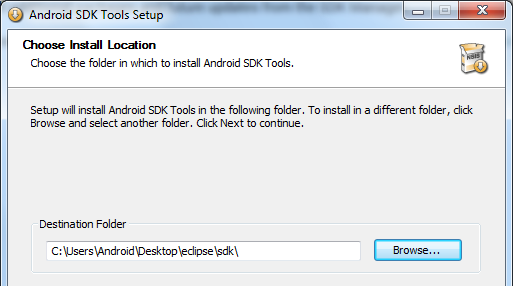
El següent procediment mostra com muntar l’entorn de desenvolupament partint des de zero, és a dir, qui tingui un entorn Eclipse instal·lat al seu ordinador podrà continuar aquest guia a partir del pas 2 per actualitzar el seu sistema.

Pas 1) Descarregar Eclipse del següent enllaç <http://www.eclipse.org/downloads/>. S’extreu l’arxiu i es desa en una carpeta, en aquest cas s’ha escollit la carpeta anomenada **eclipse** situada a l’escriptori (C:\Users\Android\Desktop\eclipse). S’executa l’entorn de desenvolupament Eclipse i es tanca.

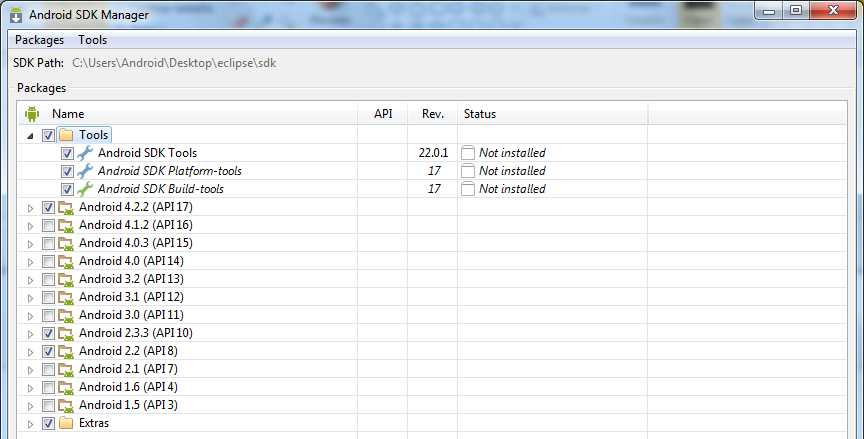
Les versions més usuals són la **Eclipse Standard** o la **Eclipse Classic**. Cal descarregar la versió adequada pel sistema operatiu que es tingui, 32 o 64 bits, però en el cas dels sistemes operatius de 64 bits les versions d’Eclipse de 32 bits també funciona correctament. S’ha de tenir en compte que la dimensió de les aplicacions en la versió d’Eclipse de 32 bits no pot ser major d’1,5 GB aproximadament, però en canvi també ocuparan menys espai en la memòria de l’ordinador, consumiran menys recursos i per tant aniran més ràpid que les desenvolupades en un Eclipse de 64 bits.

Pas 2) Descarregar l’instal·lador de l’SDK d’Android. En la direcció <http://developer.android.com/sdk/index.html> podem trobar a l’apartat ***DOWNLOAD FOR OTHER PLATFORMS*** i a la secció ***SDK Tools Only*** l’instal·lador, que en cas de Windows s’anomena [installer\_r22.0.1-windows.exe](http://dl.google.com/android/installer_r22.0.1-windows.exe), i que és ni més ni menys que un programa guiat per instal·lar l’SDK. A la redacció d’aquest document la versió actual de l’SDK és la número 22.

S’executa l’instal·lador. En aquest cas s’ha creat una carpeta anomenada **sdk** dintre del directori eclipse (C:\Users\Android\Desktop\eclipse\sdk). En el moment que demana on instal·lar l’SDK se l’indica la carpeta **sdk** tal com es mostra a la següent imatge:



Llavors aquesta carpeta serà on es guardi l’SDK. Quan ha acabat tot el procés, se li dóna a finalitzar seleccionant l’opció **Start SDK Manager**. Immediatament apareixerà en pantalla la següent finestra:



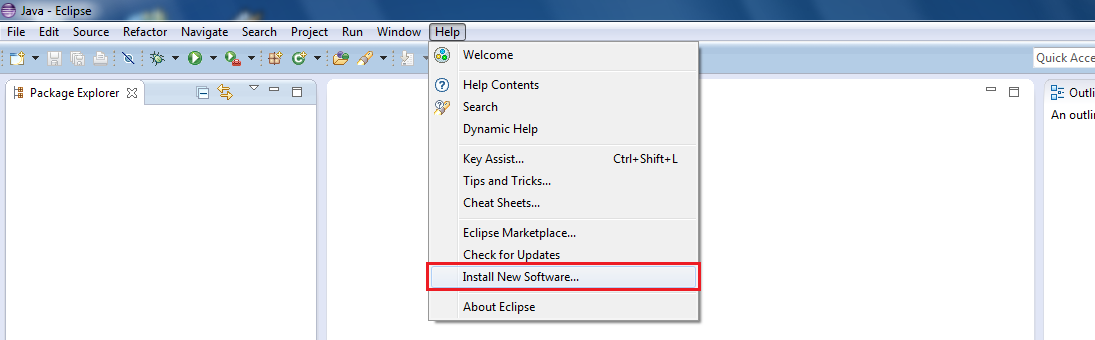
Aquí s’han de seleccionar les opcions que es veuen a la imatge:

* Tools
* Android 4.2.2
* Android 2.3.3
* Android 2.2
* Extras

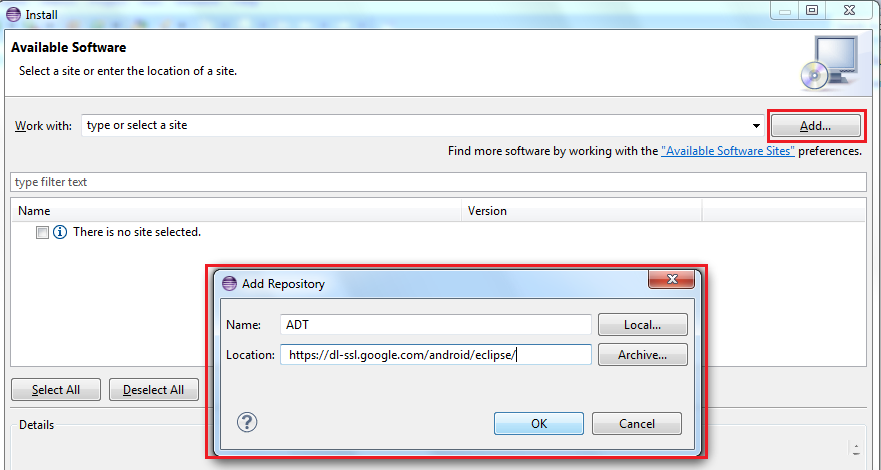
La raó per les que escollir les versions 2.2 i 2.3.3 d’Android és perquè són les més utilitzades avui en dia. És bona idea dissenyar per les versions 2.2 i 2.3.3 donat que si l’aplicació funciona en aquests nivells, també ho farà en nivells superiors.

En el cas que no s’hagi inicialitzat el SDK Manager de manera automàtica es pot accedir manualment a través del menú d’inici, dintre de tots el programes en la carpeta **Android SDK Tools** i **SDK Manager**. També anant a la carpeta on s’ha instal·lat l’SDK (C:\Users\Android\Desktop\eclipse\sdk) i executar l’**SDK Manager**.

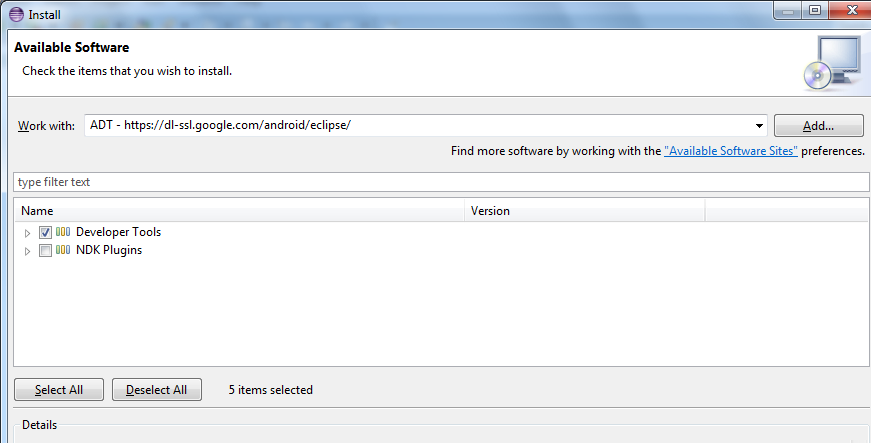
Pas 3) Ara s’instal·larà el plug-in d’Android a Eclipse. Seleccionant **Help** → **Install Software ...** com es veu a continuació:



S’entra en una finestra, es selecciona **Add** i s’obrirà un diàleg per emplenar.



Aquí s’ha d’introduir a **Location** la direcció <https://dl-ssl.google.com/android/eclipse/>. Es dóna a **OK** i llavors apareixerà la finestra actualitzada d’aquesta manera:

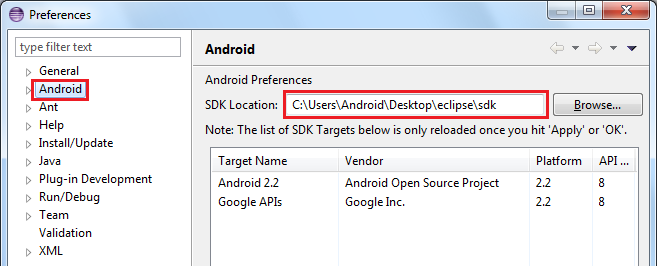


Es selecciona només l’opció **Developer Tools**.

Pas 4) Quan ha finalitzat el pas anterior es torna a iniciar l’Eclipse. Ara es pot veure a la barra superior de l’Eclipse les icones corresponents al plug-in d’Android.



Es verifica que l’entorn reconegui correctament la ubicació de l’SDK. Per veure-ho s’ha d’anar a **Window** → **Preferences** i a la finestra que s’obre, a l’apartat **Android** s’hauria de veure la direcció de la carpeta on està instal·lat l’SDK, tal com es pot veure a continuació:



En el cas de no ser així, es selecciona manualment el directori indicat en la instal·lació.

Pas 5) Finalment es crea un dispositiu virtual que permeti provar les aplicacions dintre de l’ordinador. Aquest procés es mostra a l’apartat **Annex A: Creació d’un dispositiu virtual**.

# Consumir servei web des d’una aplicació Android

En aquest apartat s’explicarà breument com realitzar una aplicació capaç de connectar-se amb el servei web, és a dir, un client soap. A més estarà disponible un codi font de l’aplicació sencera per poder provar-la. Si hi ha problemes per poder llegir correctament els comentaris del codi font s’ha de seleccionar el projecte, botó dret **Properties** → **Resource** → **Text File Encoding** → **Other: UTF - 8.**

## Aspectes a tenir en compte

Per poder connectar amb el servei web és necessari importar una llibreria o framework que permeti realitzar crides soap des del costat client, que és la banda on es troba l’aplicació a dissenyar. En l’exemple s’ha fet servir la llibreria kSOAP 2 (<http://ksoap2.sourceforge.net/>). També es pot trobar a la carpeta del codi font de mostra.

Un altre aspecte a tenir en compte és que a per poder realitzar l’enviament d’una petició tipus soap s’ha de fer en backgronud i per tant l’exemple farà servir un classe que hereti d’AsyncTask. No està permès fer una petició soap dintre del thread principal.

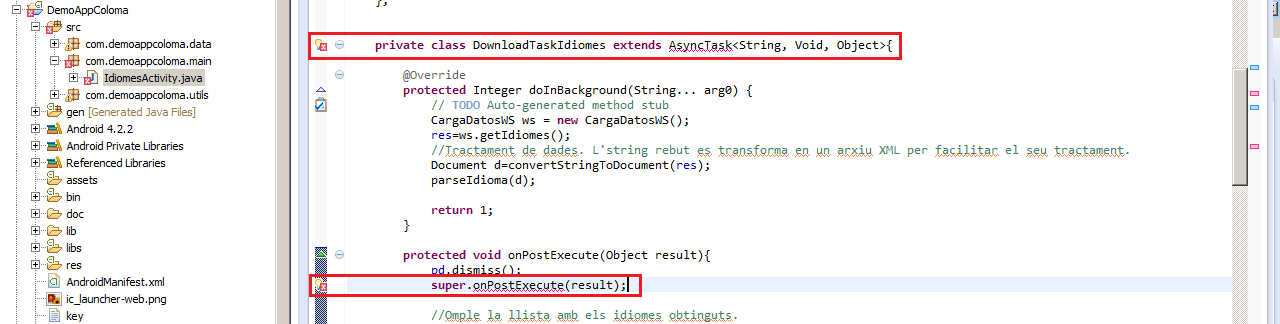
També s’han de donar els permisos necessaris per tal que l’aplicació pugui accedir a internet. Al Manifest.xml s’afegeix la següent instrucció:

|  |
| --- |
| <uses-permission android:name="android.permission.INTERNET"/> |

Per fer l’exemple compatible amb el major nombre de versions Android possible s’ha establert la versió de l’aplicació a una molt baixa, imposant al Manifest.xml la següent ordre:

|  |
| --- |
| <uses-sdk android:targetSdkVersion="3"/> |

Això pot provocar que l’Eclipse interpreti que hi ha errors en alguns del mètodes de la classe que deriva d’AsyncTask com es pot veure:



Aquest problema es soluciona fent **Project** → **Clean** del projecte on es troba l’aplicació. Després de fer-ho els errors desapareixen.

## Descripció del codi

L’esquema bàsic d’una petició soap client és la següent:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | **public** String getData(){  String res=**null**;  SoapObject rpc;  rpc=**new** SoapObject( namespace, "nomMetode");  //En el cas d'haver-hi paràmetres  //rpc.addProperty("nomAtribut", valorAtribut);    SoapSerializationEnvelope envelope = **new** SoapSerializationEnvelope(SoapEnvelope.*VER11*);  envelope.bodyOut=rpc;    envelope.dotNet=**true**;  envelope.encodingStyle=SoapSerializationEnvelope.*XSD*;    HttpTransportSE androidHttpTransport=**null**;    **try** {  androidHttpTransport = **new** HttpTransportSE(conexion);  androidHttpTransport.debug=**true**;    //SoapACtion  androidHttpTransport.call(namespace+"nomMetode", envelope);    res=androidHttpTransport.responseDump;  } **catch** (Exception e) { /\* Tractament d'errors.\*/}    **return** res;  } |

En les línies 4 i 5 es crea l’objecte fet servir com a protocol de comunicacions. En ell s’indiquen el namespace i el nom del mètode que es farà servir, que serà un String on s’indica el nom. La línia 9 es crea l’**Envelope**, que és el contenidor on s’envien totes les dades de la petició soap.

A les línies 20 i 21 es prepara l’enviament de l’**Envelope**. A la línia 24 s’envia el contenidor de la petició soap indicant el mètode del servei web que es vol executar. A la línia 26 s’obté la resposta del servei web.

Les variables de la connexió i del namespace tindran els següents valors:

|  |  |
| --- | --- |
| 1  2  3  4 | //Ubicació del servidor on es troba el servei web.  **public** String conexion = "http://195.235.66.115:10001/WSApp/WSApp/AppColoma.asmx";  //Namespace fet servir al servei web.  **public** String namespace = "http://www.gramene.net/appcoloma/"; |

A l’adreça <http://195.235.66.115:10001/WSApp/WSApp/AppColoma.asmx?op=idiomesFitxa> podem trobar la descripció del mètode **idiomesFitxa**, que és el que es fa servir a l’exemple. Allà apareix la descripció de la resposta WSDL:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | HTTP/1.1 200 OK  Content-Type: text/xml; charset=utf-8  Content-Length: length  <?xml version="1.0" encoding="utf-8"?>  <soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  <soap:Body>  <idiomesFitxaResponse xmlns="http://www.gramene.net/appcoloma/">  <idiomesFitxaResult>  <codiEstat>  <intCodiEstat>short</intCodiEstat>  <strDescripcioEstat>string</strDescripcioEstat>  <intTotalResultats>long</intTotalResultats>  </codiEstat>  <idiomes>  <Idioma>  <intIdioma>short</intIdioma>  <strIdioma>string</strIdioma>  </Idioma>  <Idioma>  <intIdioma>short</intIdioma>  <strIdioma>string</strIdioma>  </Idioma>  </idiomes>  </idiomesFitxaResult>  </idiomesFitxaResponse>  </soap:Body>  </soap:Envelope> |

De les línies 12 a la 28 hi ha les dades de la resposta. Es pot diferenciar:

* **CodiEstat** (línies 13 a la 17): indica el resultat de l’operació. Està format per un indicador numèric (línia 14) anomenat **intCodiEstat**, la descripció del qual es troba a **strDescripcioEstat**. El camp de la línia 16, **intTotalResultats** indica el total de respostes que s’han obtingut.
* Array d’objectes tipus **Idioma** (línies 18 a la 27): es pot observar com de la línia 19 a la 22 hi ha un objecte tipus **Idioma**, amb un element numèric anomenat **intIdioma** i un element descriptiu tipus text anomenat **strIdioma**, i de la línia 23 a la 26 hi ha un altre objecte **Idioma**. Això indica que aquesta part serà un array d’elements d’aquest tipus.

En aquest cas sabem que el mètode no necessita rebre cap paràmetre per poder treballar, perquè ens indica que el seu **Envelope** ha de ser de la següent forma:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | POST /WSApp/WSApp/AppColoma.asmx HTTP/1.1  Host: 195.235.66.115  Content-Type: text/xml; charset=utf-8  Content-Length: length  SOAPAction: "http://www.gramene.net/appcoloma/idiomesFitxa"  <?xml version="1.0" encoding="utf-8"?>  <soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  <soap:Body>  <idiomesFitxa xmlns="http://www.gramene.net/appcoloma/" />  </soap:Body>  </soap:Envelope> |

Es pot veure que de les línies 11 a la 13 no s’indica cap paràmetre a enviar. En canvi, si es té en compte el mètode **horarisFitxa** es pot veure com és necessari enviar-li els següents paràmetres:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | POST /WSApp/WSApp/AppColoma.asmx HTTP/1.1  Host: 195.235.66.115  Content-Type: text/xml; charset=utf-8  Content-Length: length  SOAPAction: "http://www.gramene.net/appcoloma/horarisFitxa"  <?xml version="1.0" encoding="utf-8"?>  <soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  <soap:Body>  <horarisFitxa xmlns="http://www.gramene.net/appcoloma/">  <idioma>int</idioma>  <idFitxa>int</idFitxa>  </horarisFitxa>  </soap:Body>  </soap:Envelope> |

A les línies 13 i 14 s’indica els tipus de paràmetre que necessita rebre el mètode. Un de tipus numèric anomenat **idioma** i un altre també de tipus numèric anomenat **idFitxa**. La seva preparació podria ser la següent al codi font:

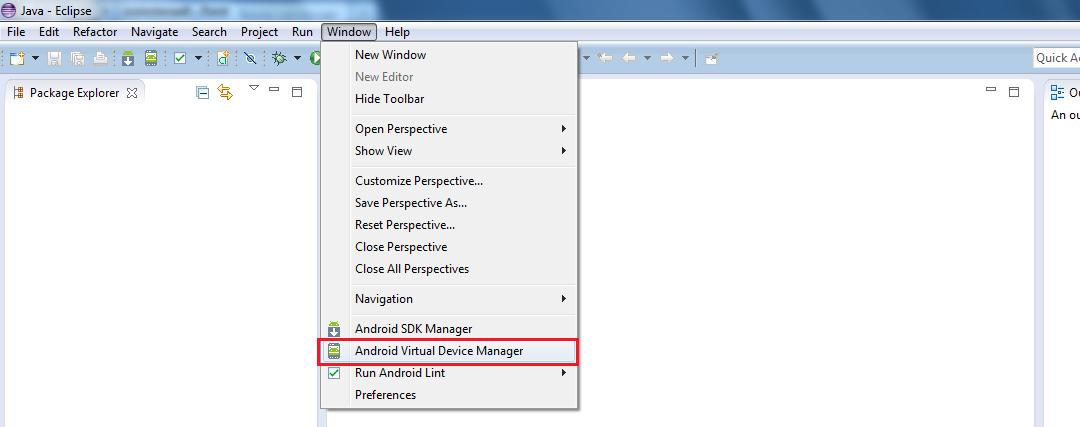
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | **…**  SoapObject rpc;  rpc=**new** SoapObject( namespace, "nomMetode");  //En el cas d'haver-hi paràmetres  rpc.addProperty("idioma", 1); rpc.addProperty("idFitxa", 11503);  SoapSerializationEnvelope envelope = **new** SoapSerializationEnvelope(SoapEnvelope.*VER11*);  … |

La resta de codi Java es manté igual.

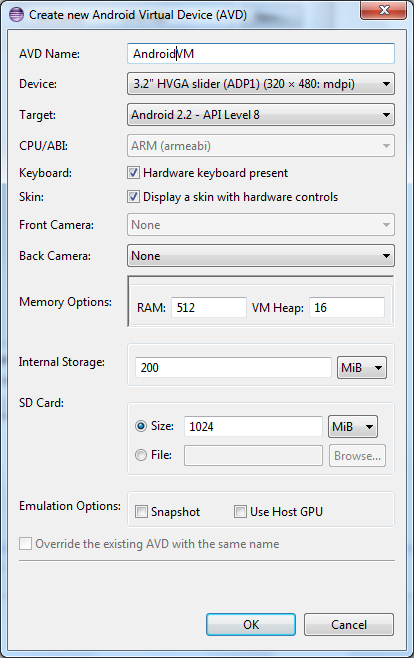
# Annex A: Creació d’un dispositiu virtual

A continuació es mostrarà com crear un dispositiu virtual d’Android per poder simular un smartphone a l’ordinador.

Un cop s’executa l’Eclipse es va a **Window** → **Android Virtual Device Manager**.



S’obrirà una finestra i es selecciona el botó **New** de la part de la dreta. Llavors s’obrirà la següent finestra:



Se li dóna un nom i es tria el dispositiu que es vulgui a **Device**. De manera automàtica s’estableix el nivell de l’API ja que va lligat al dispositiu. Una mida recomanable per **SD Card** és 1024. Després de consolidar els canvis, apareixerà la següent finestra, des de la que es pot arrancar el dispositiu creat prement el botó **Start**.

