

# Instalación en Ubuntu

## Sumario

- 1 Introducción
- 2 Instalación OpenJDK
- 3 Instalación do IDE Eclipse
  - ◆ 3.1 IDE Eclipse para tódolos usuarios
- 4 Instalación Android SDK
  - ◆ 4.1 Android SDK para tódolos usuarios
- 5 Instalación plugin ADT
- 6 Librarías de 32 bits precisas para Ubuntu 13.10 Linux de 64 bits
- 7 Configuración Eclipse e plugin ADT. Actualizacións
- 8 Carpetas de traballo
- 9 Fixar a icona ao Lanzador

## Introdución

- Como xa se dixo, a versión de Ubuntu na que se elaborou o curso é a **Ubuntu Desktop 13.10 de 64 bits**. Tamén se probou na versión **Ubuntu Desktop 14.04 de 64 bits**
- A razón de instalar cada compoñente por separado é que nos vai permitir ter un maior control de todo canto se precisa e ademais vai poder permitir que cada lector instale os compoñentes que precise en función dos que xa ten instalados.
- Compoñentes:



O **OpenJDK** (Java Development Kit), versión libre do JDK liderada por **Oracle**.

Como xa se indicou vaise instalar o JDK no canto do JRE (MV de Java), por se alguén desexa programar a maiores en Java en consola ou con outros IDEs.

Nesta ocasión vaise facer uso de **OpenJDK** e non do JDK de Oracle, porque xa vén nos repositorios de Ubuntu.

Quen o desexe pode descargar o JDK orixinal de Oracle e instalalo seguindo os pasos:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>.



O IDE (Integrated Development Environment) de **Eclipse**.



O **SDK** (Software Development Kit) de **Android**



O **ADT Plugin** (Android Development Tools) para Eclipse. Permitirá facer uso dende o IDE do SDK de Android.

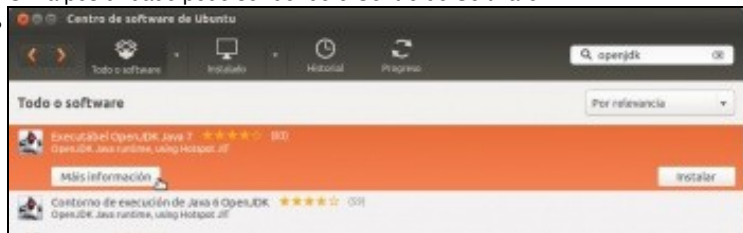
- Quen desexe instalar todo isto en Ubuntu con entorno Unity e non se sinta con soltura no seu manexo, na parte III do seguinte curso pode observar como interactuar con Unity: [Curso Platega: Ubuntu Desktop. Un sistema dual \(MS Windows / GNU/Linux\)](#)

## Instalación OpenJDK

- Vaise instalar a versión 7.
- En <http://openjdk.java.net> pódese obter máis información sobre esta plataforma.
- Instalación OpenJDK



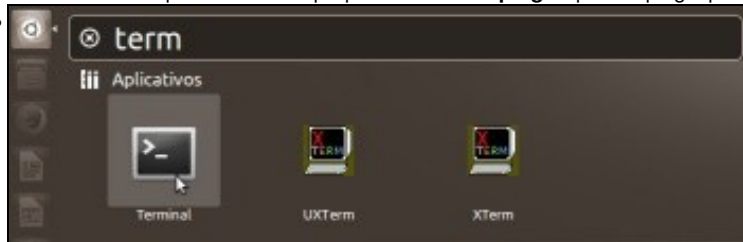
Unha posibilidade pode ser dende o Centro de Software.



Procurar OpenJDK.



Pódese marcar que se instale o paquete **icedtea-X-plugin** que é o plugin para os navegadores Web. Pero ...



... vaise instalar dende o Terminal. Premer no botón de Inicio, procurar **Terminal** e lanzalo.

```

admin@ubase:~$ apt-cache search openjdk
default-jdk - Standard Java or Java compatible Development Kit
default-jdk-doc - Standard Java or Java compatible Development Kit (documentation)
default-jre - Standard Java or Java compatible Runtime
default-jre-headless - Standard Java or Java compatible Runtime (headless)
icedtea-7-jre-jamvm - Alternative JVM for OpenJDK, using JamVM
icedtea-7-plugin - web browser plugin based on OpenJDK and IcedTea to execute Java applets
openjdk-7-doc - OpenJDK Development Kit (JDK) documentation
openjdk-7-jdk - OpenJDK Development Kit (JDK)
openjdk-7-jre - OpenJDK Java runtime, using Hotspot JIT
openjdk-7-jre-headless - OpenJDK Java runtime, using Hotspot JIT (headless)
openjdk-7-jre-lib - OpenJDK Java runtime (architecture independent libraries)
openjdk-7-source - OpenJDK Development Kit (JDK) source files
openjdk-7-dbg - Java runtime based on OpenJDK (debugging symbols)
openjdk-7-demo - Java runtime based on OpenJDK (demos and examples)
java-package - Utility for creating Java Debian packages
freemind - Java Program for creating and viewing Mindmaps
icedtea-6-jre-cacao - Alternative JVM for OpenJDK, using Cacao

```

Con **apt-cache search openjdk** vense os paquetes dispoñibles. Entre eles, o mencionado **icedtea-X-plugin** e o metapaquete **default-jdk**, que sempre vai instalar a versión máis recente do JDK de OpenJDK.

```

admin@ubase:~$ sudo apt-get install default-jdk icedtea-7-plugin
[sudo] password for admin:

```

Executar **sudo apt-get install default-jdk icedtea-7-plugin** para instalar os dous paquetes.

```

admin@ubase:~$ sudo apt-get install default-jdk icedtea-7-plugin
Paquetes suxeridos:
equivs libbonobo2-bin desktop-base libgnomevfs2-bin libgnomevfs2-extra gamln
fan gnome-mime-data libice-doc libsm-doc libxcb-doc libxt-doc openjdk-7-demo
openjdk-7-source visualvm sun-java6-fonts fonts-ipafont-gothic
fonts-ipafont-mincho ttf-telugu-fonts ttf-oriya-fonts ttf-kannada-fonts
ttf-bengali-fonts
Os seguintes paquetes NOVOS hanse instalar:
ca-certificates-java default-jdk default-jre default-jre-headless
fonts-dejavu-extra icedtea-7-jre-jamvm icedtea-7-plugin icedtea-netx
icedtea-netx-common java-common libatk-wrapper-java libatk-wrapper-java-jni
libbonobo2-0 libbonobo2-common libgconf2-4 libgif4 libgime2-0 libgime2-bin
libgime2-common libgimevfs2-0 libgimevfs2-common libice-dev libidl-common
libidl0 liborbit2 libpthread-stubs0 libpthread-stubs0-dev libsm-dev
libx11-dev libx11-doc libxau-dev libxcb1-dev libxdmcp-dev libxt-dev
openjdk-7-jdk openjdk-7-jre openjdk-7-jre-headless openjdk-7-jre-lib
tzdata-java x11proto-core-dev x11proto-input-dev x11proto-kb-dev
xorg-sgml-doctools xtrans-dev
0 anovados, 44 instalados, Vanse retirar 0 e deixar 11 sen anovar.
Ten que recibir 67,1 MB de arquivos.
Despós desta operación ocuparanse 113 MB de disco adicionais.
Quere continuar [S/n]?

```

Información sobre o que se vai instalar. Indicar que Si se desexa continuar.

```

admin@ubase:~$ javac -version
javac 1.7.0_25
admin@ubase:~$ java -version
java version "1.7.0_25"
OpenJDK Runtime Environment (IcedTea 2.3.12) (7u25-2.3.12-4ubuntu3)
OpenJDK 64-Bit Server VM (build 23.7-b01, mixed mode)
admin@ubase:~$

```

Comprobación das versións do compilador de Java (**javac -version**) e da MV de Java (JRE) (**java -version**).

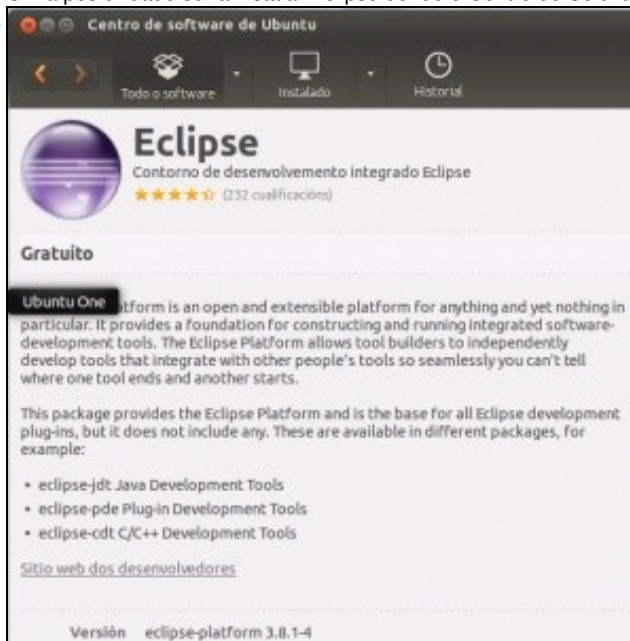
## Instalación do IDE Eclipse

- A continuación instalárase o IDE de Eclipse, cuxo proceso de instalación é simplemente descargar e descomprimir un paquete, sen máis.
- En <http://eclipse.org/> está o sitio do IDE.

- Instalación IDE Eclipse



Unha posible sería instalar Eclipse dende o Centro de Software.



Pero a versión do repositorio é a 3.8.1 e ...





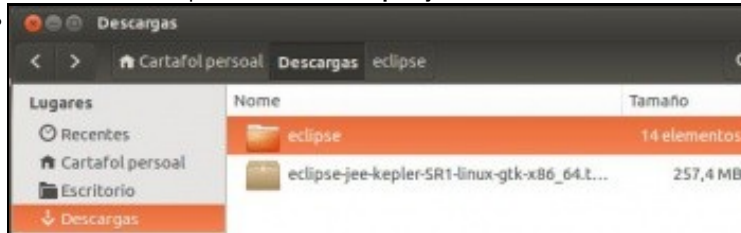
En <http://www.eclipse.org/downloads/> podemos atopar a última versión. A imaxe amosa a versión 4.3.1. Na actualidade é a 4.4.1. Neste caso descárgase a versión de 64 bits do **IDE de Java EE** (Enterprise Edition) por ser máis completa cá do IDE de Java, pero con esta última tamén sería suficiente. Cada quen que baixe a versión de 64 ou 32 bits en función do seu S.O.



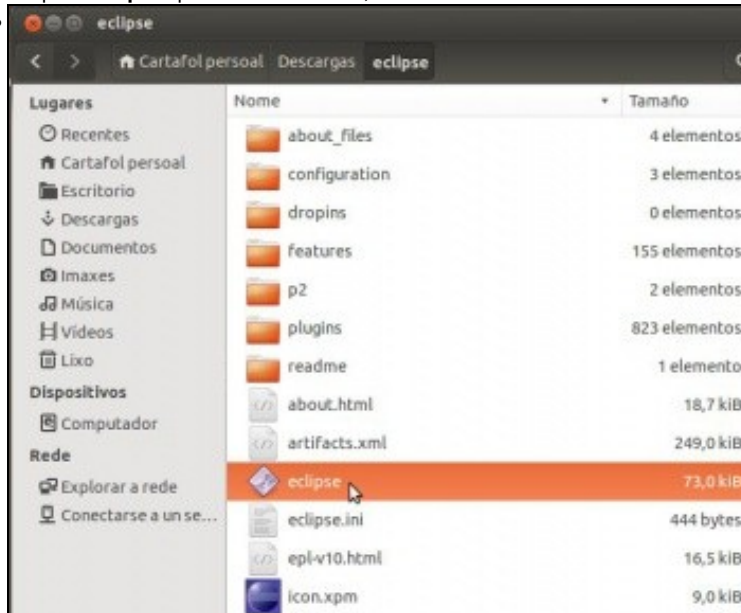
Descomprimir o ficheiro. **Nota:** na versión 13.10 de Ubuntu dá un erro ao descomprimir este paquete dende o entorno gráfico.

```
admin@ubase: ~/Descargas
admin@ubase:~$ cd Descargas/
admin@ubase:~/Descargas$ tar -xf eclipse-jee-kepler-SR1-linux-gtk-x86_64.tar.gz
```

Nese caso descomprimir con **tar xf eclipse-jee.....**



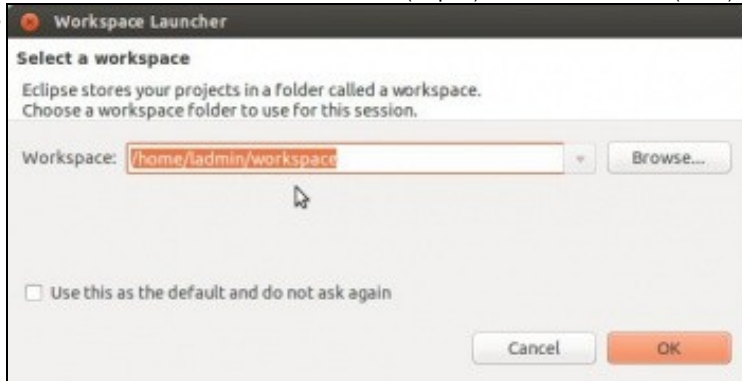
Carpeta **Eclipse** que xa contén o IDE, listo ...



... para traballar. Facer dobre click sobre o executable (asegurarse que teña permisos de execución).



Abríndose o IDE. Versión da imaxe: 4.3.1 (Kepler). Imaxe actual 4.4.1 (Luna)



Pregunta onde se desexan gardar os proxectos.



O IDE funcionando. Pero, pero !!!!! Só está dispoñible para o usuario que o baixou e non para os demais do sistema. Isto non é desexable nunha aula.

## IDE Eclipse para tódolos usuarios

- Para que o IDE Eclipse estea dispoñible para tódolos usuarios do sistema, só é preciso poñer o cartafol Eclipse nunha ruta accesible por todo usuario. Neste caso vaise facer uso da máis recomendada **/opt/eclipse**.
- Logo só resta axustar o propietario e os permisos do cartafol.
- Pódese facer todo isto dende o contorno gráfico pero vaise realizar dende a consola con **sudo**.
- Finalmente creárase un **menú** para que calquera usuario poida lanzar o IDE dende o Botón de Inicio do Lanzador.
- Instalación IDE Eclipse para tódolos usuarios

```

ladmin@ubase: ~
ladmin@ubase:~$ sudo mv Descargas/eclipse /opt/
[sudo] password for ladmin:

```

Mover o cartafol eclipse: **sudo mv Descargas/eclipse /opt/**

```

ladmin@ubase: ~
ladmin@ubase:~$ sudo chown root:root /opt/eclipse -R
ladmin@ubase:~$
ladmin@ubase:~$ sudo chmod 755 /opt/eclipse/ -R
ladmin@ubase:~$

```

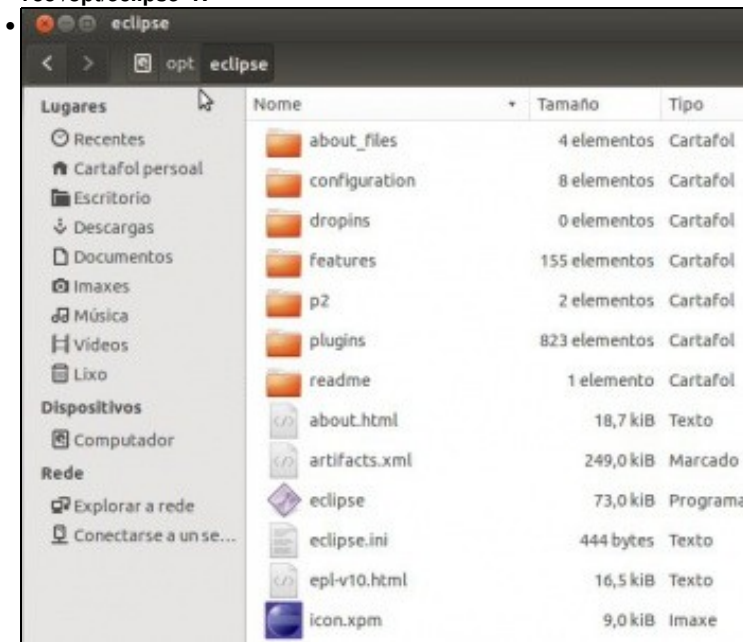
Cambiar o usuario e o grupo propietario dese cartafol e o seu contido: **sudo chown root:root /opt/eclipse -R**

```

ladmin@ubase: ~
ladmin@ubase:~$ sudo chown root:root /opt/eclipse -R
ladmin@ubase:~$
ladmin@ubase:~$ sudo chmod 755 /opt/eclipse/ -R
ladmin@ubase:~$

```

Dar permisos de control total ao propietario, aos demais usuarios de lectura e execución para ese cartafol e todo o seu contido: **sudo chmod 755 /opt/eclipse -R**



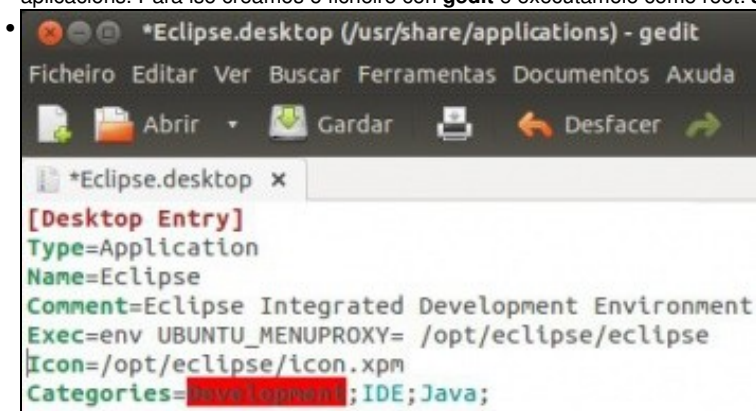
Un usuario normal podería ir ao cartafol **/opt/eclipse** e lanzar o IDE, pero mellor crear un menú.

```

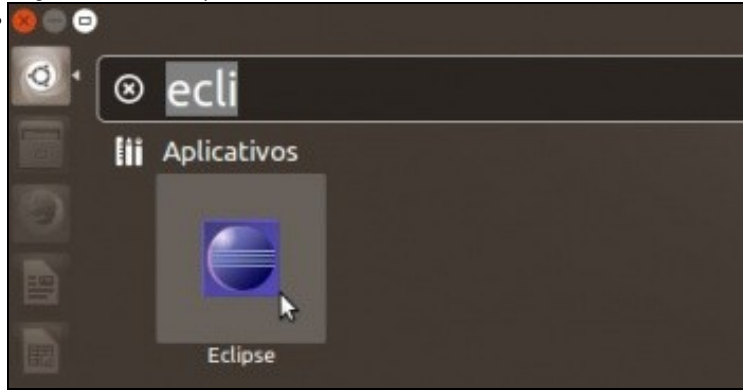
ladmin@ubase: ~
ladmin@ubase:~$ sudo gedit /usr/share/applications/Eclipse.desktop

```

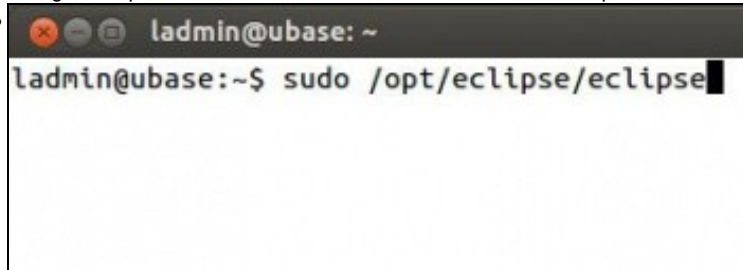
Para iso hai que crear un lanzador (extensión desktop) en **/usr/share/applications**. Nese directorio están os lanzadores (menús) das demais aplicacións. Para iso creamos o ficheiro con **gedit** e executámolo como root: **sudo gedit /usr/share/applications/Eclipse.desktop**



Pegamos o contido que se amosa dúas imaxes máis abaixo, modificando as rutas se fose o caso.



Se agora se preme no botón Inicio xa se ve o Lanzador de Eclipse.



Tamén se pode lanzar Eclipse dende a consola. Como root hai que usar o comando **sudo**, como usuario normal sen *sudo*.

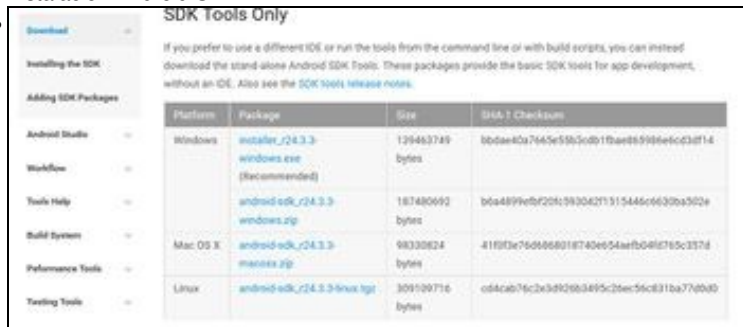
```
[Desktop Entry]
Type=Application
Name=Eclipse
Comment=Eclipse Integrated Development Enviroment
Exec=env UBUNTU_MENUPROXY= /opt/eclipse/eclipse
Icon=/opt/eclipse/icon.xpm
Categories=Development;IDE;Java;
```

**NOTA:** Na versión 13.10 de Ubuntu se non se engade en Exec **env UBUNTU\_MENUPROXY** non se amosarían os menús da aplicación.

## Instalación Android SDK

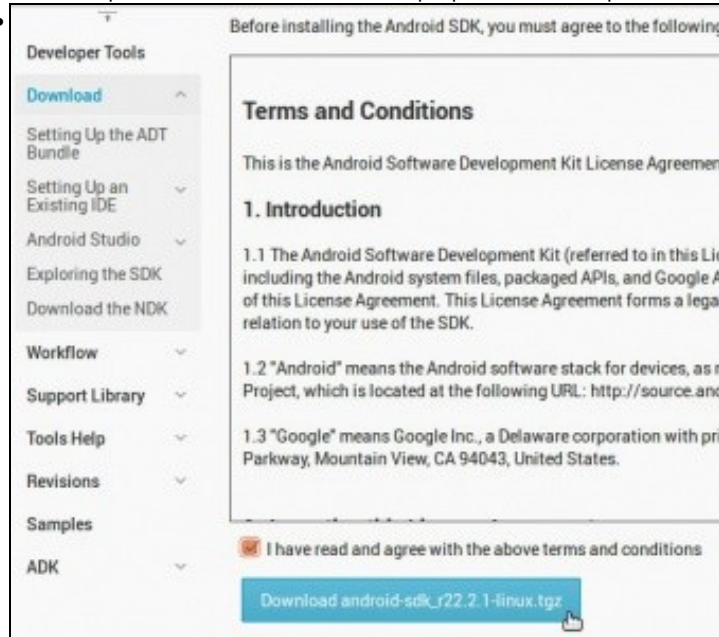
- O proceso de instalación é semellante ao do IDE Eclipse: descargar un paquete e descomprimilo.
- En <http://developer.android.com/sdk/index.html#Other> podemos comezar co proceso de instalación.
- Poderíamos descargar o **Bundle** asociado ao noso SO e este xa traería integrado nun só paquete: Eclipse + Android SDK + Plugin ADT. Só restaría instalar o JRE (ou JDK se se desexa).
  - ◆ Poderíase configurar para que puidera ser usado por calquera usuario do sistema do mesmo xeito que se fixo antes con Eclipse e que se vai facer a continuación co SDK de Android.
- Pero, como indicamos, vaise instalar cada compoñente por separado.

### Instalación Android SDK

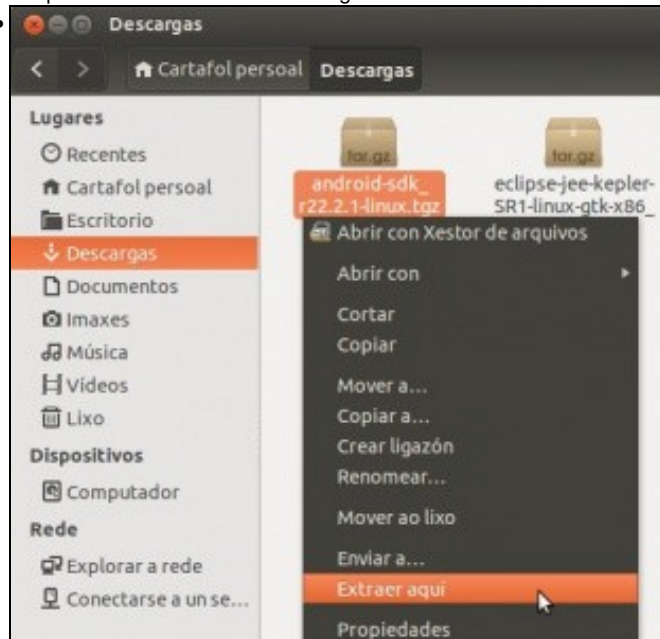




En <http://developer.android.com/sdk/index.html#Other> poderemos descargar o **SDK** para Android. asociado ao noso SO. Neste caso baixárase o SDK para Linux, fíxase que a versión é a mesma para 32 bits que para 64 bits, co cal é probable que nas versións do SO de 64 bits haxa que instalar **librarías de 32 bits** que precisen os compoñentes do SDK para ser executadas.



Aceptamos as condicións e descargamos.



Descomprimos o ficheiro. En modo gráfico ...

```
admin@ubase: ~/Descargas
ladningubase:~$ cd Descargas/
ladningubase:~/Descargas$
ladningubase:~/Descargas$ tar -xf android-sdk_r22.2.1-linux.tgz
ladningubase:~/Descargas$
```

... ou por consola.

- Agora xa teríamos unha carpeta co SDK, da que podería facer uso o usuario actual pero non todos os demais do sistema.
- Por tanto, igual que se fixo antes con Eclipse para configuralo para tódolos usuarios, vaise proceder agora có SDK de Android.

## Android SDK para tódolos usuarios

- Igual que antes: poñer o SDK nunha ruta accesible para tódolos usuarios do sistema: **/opt/android-sdk-linux**.
- Logo só resta axustar o propietario e os permisos do cartafol.
- Pódese facer todo isto dende o contorno gráfico pero vaise realizar dende a consola con sudo.

- Instalación Android SDK para tódolos usuarios

```
admin@ubase: ~
ladningubase:~$ sudo mv Descargas/android-sdk-linux/ /opt/
```

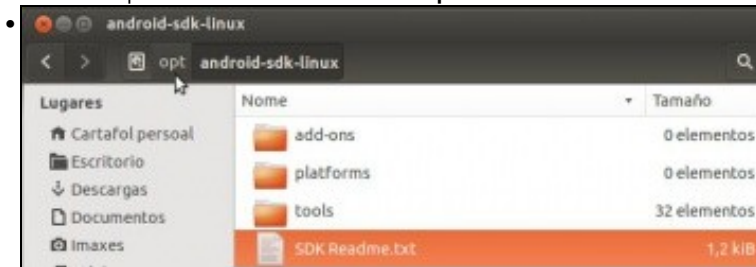
Mover o SDK: **sudo mv Descargas/android-sdk-linux /opt/**

```
admin@ubase: ~
ladningubase:~$ sudo chown root:root /opt/android-sdk-linux/ -R
ladningubase:~$
ladningubase:~$ sudo chmod 755 /opt/android-sdk-linux/ -R
ladningubase:~$
```

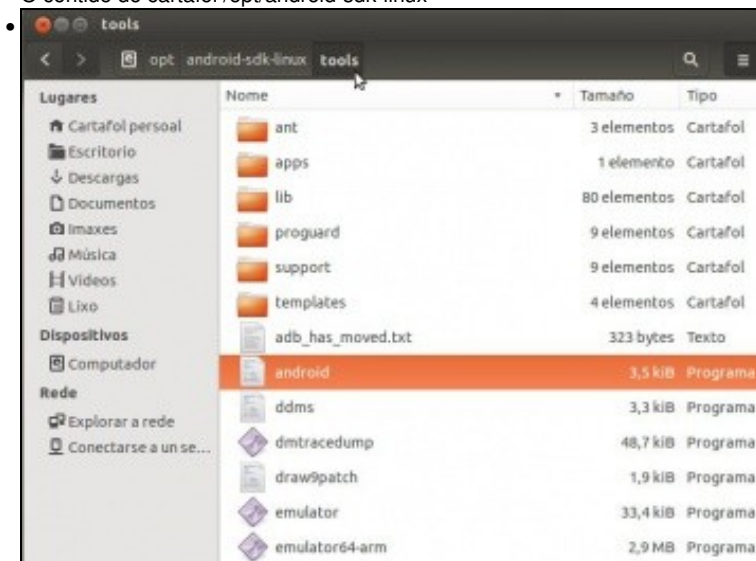
Axustar os propietarios do cartafol e o seu contido: **sudo chown root:root /opt/android-sdk-linux -R**

```
admin@ubase: ~
ladningubase:~$ sudo chown root:root /opt/android-sdk-linux/ -R
ladningubase:~$
ladningubase:~$ sudo chmod 755 /opt/android-sdk-linux/ -R
ladningubase:~$
```

Axustar os permisos ... **sudo chmod 755 /opt/android-sdk-linux -R**



O contido do cartafol /opt/android-sdk-linux



O contido do subcartafol **tools**. Hai un programa chamado **android** que permite, entre outras cousas, xestionar as APIs do SDK e os emuladores de dispositivos.

```
admin@ubase:~$ sudo /opt/android-sdk-linux/tools/android --help
Usage:
  android [global options] action [action options]
Global options:
-h --help           : Help on a specific command.
-v --verbose        : Verbose mode, shows errors, warnings and all messages.
--clear-cache       : Clear the SDK Manager repository manifest cache.
-s --silent         : Silent mode, shows errors only.

Valid actions are composed of a verb and an optional direct object:

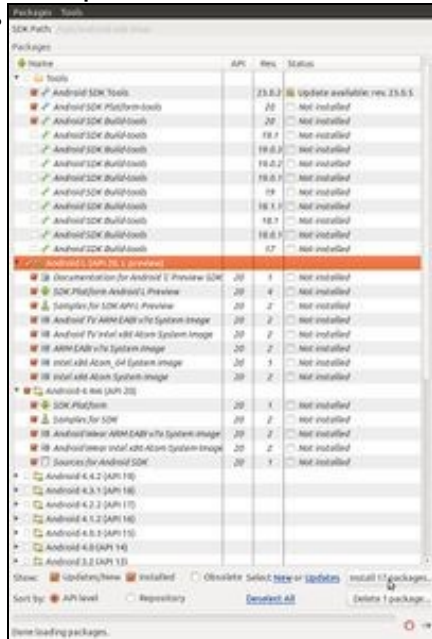
- sdk               : Displays the SDK Manager window.
- avd               : Displays the AVD Manager window.
- list              : Lists existing targets or virtual devices.
- list avd          : Lists existing Android Virtual Devices.
- list target       : Lists existing targets.
- list sdk           : Lists remote SDK repository.
- create avd        : Creates a new Android Virtual Device.
- move avd          : Moves or renames an Android Virtual Device.
- delete avd        : Deletes an Android Virtual Device.
- update avd        : Updates an Android Virtual Device to match the folders of a new SDK.
- create project    : Creates a new Android project.
- update project    : Updates an Android project (must already have an AndroidManifest.xml).
- create test-project : Creates a new Android project for a test package.
- update test-project : Updates the Android project for a test package (must already have an AndroidManifest.xml).
```

Podemos executalo dende a consola, indicando a súa ruta. Con **sudo** é executado como root, e sen sudo como un usuario normal.

**/opt/android-sdk-linux/tools/android --help.**

```
admin@ubase:~$ sudo /opt/android-sdk-linux/tools/android sdk
```

Como imos actualizar as APIs do SDK e descargar os seus compoñentes é preciso lanzar o **Xestor do SDK (SDK Manager)** como root: **sudo /opt/android-sdk-linux/tools/android sdk**. A opción **sdk** é a que se usa por defecto cando non se indica ningunha.



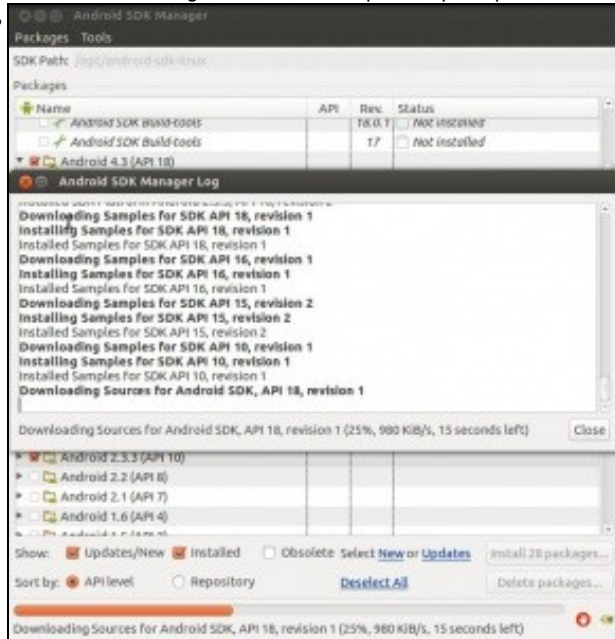




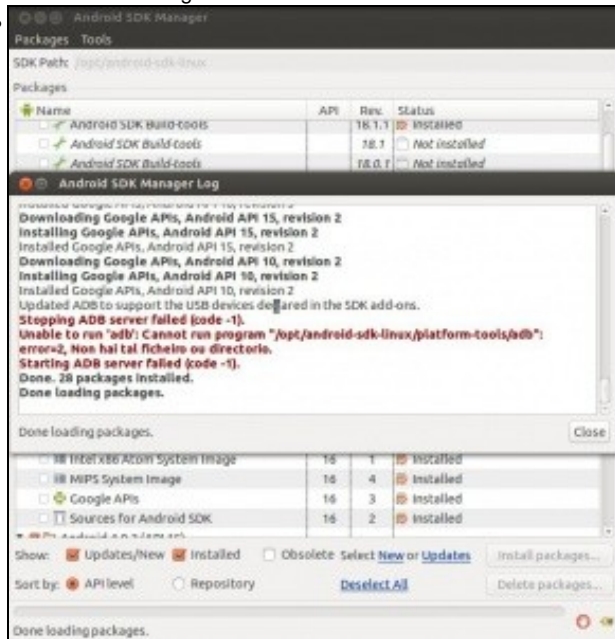
Olo!!! que abaixo hai máis licencias para aceptar. E premer en **Install**



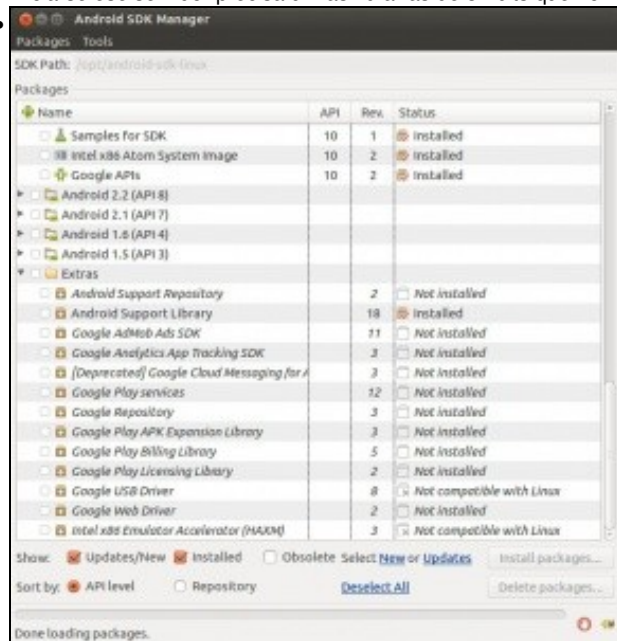
Comezo da descarga. Da un erro de que non pode parar o servidor ADB. Este compoñente verase nun apartado posterior.



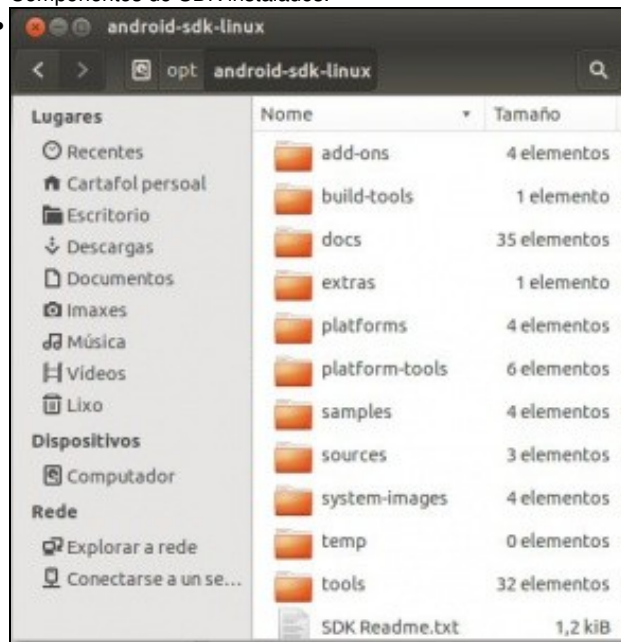
Continúa a descarga.



Erro de que non pode iniciar o servidor ADB, e isto si que é un problema. Isto pasa nas versión de 64 bits de Ubuntu 13.10, porque para iniciarse ese servidor precisa unhas librerías de 32 bits que non están instaladas. Logo arranxarase.



Compoñentes do SDK instalados.



Carpetas creadas e asociadas ao SDK.

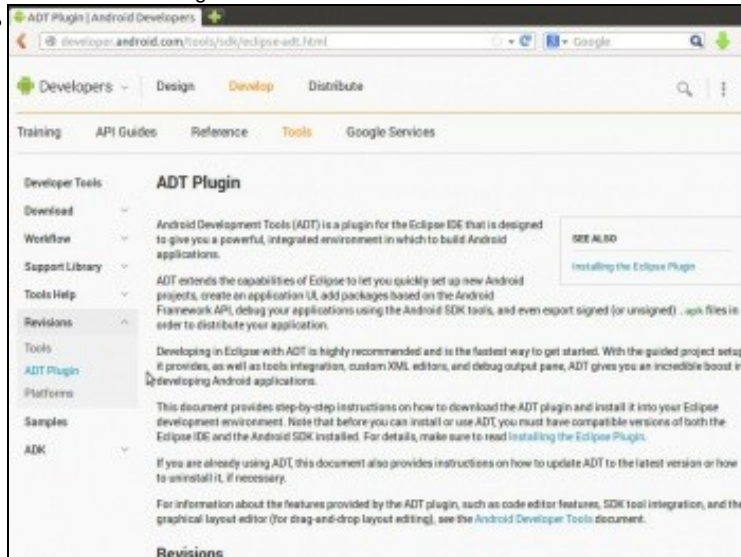
**NOTA::** Cando decidimos instalar unha nova versión do S.O. Android non é necesario marcar todas as opcións dentro desta API. Así:

- ◊ SDK: Necesario para poder desenvolver a aplicación. Sempre debe ser marcada.
- ◊ Samples for SDK: Documentación de exemplo. Non é necesario marcala.
- ◊ System Image (ARM/Intel/MIPS): Cando máis adiante utilizemos un dispositivo virtual para correr as nosas aplicacións, será necesario ter descargado a lo menos unha destas imaxes. O normal é que se temos un procesador INTEL descarguemos unha imaxe INTEL xa que despois poderemos instalar un acelerador hardware para dito micro. Se non temos intel poderíamos descargar calquera das opcións.
- ◊ Google API: O mesmo que o System Image pero para dar soporte a aplicacións que van facer uso de Google Maps. Se non imos traballar con el non será necesario descargarlas.
- ◊ Sources for Android SDK: Fontes das funcións do SDK. Non é necesario descargarlas.

## Instalación plugin ADT

- Só resta instalar o plugin en Eclipse que permita traballar dende este IDE co SDK de Android.
- Este plugin chámase: **Android Development Tools (ADT)** e estende as capacidades de Eclipse para xestionar dispositivos móbiles (emulados e reais), compilar e instalar aplicacións en dispositivos, etc.
- En <http://developer.android.com/tools/sdk/eclipse-adt.html> hai máis información sobre este compoñente e como se pode instalar no IDE.

### • Instalación ADT Plugin



En <http://developer.android.com/tools/sdk/eclipse-adt.html> premer en **Installing the Eclipse Plugin**

### • Download the ADT Plugin

1. Start Eclipse, then select **Help > Install New Software**.
2. Click **Add**, in the top-right corner.
3. In the Add Repository dialog that appears, enter "ADT Plugin" for the Name and the following URL for the Location:
4. Click **OK**.  
If you have trouble acquiring the plugin, try using "http" in the Location URL, instead of "https" (https is preferred for security reasons).
5. In the Available Software dialog, select the checkbox next to Developer Tools and click **Next**.
6. In the next window, you'll see a list of the tools to be downloaded. Click **Next**.
7. Read and accept the license agreements, then click **Finish**.  
If you get a security warning saying that the authenticity or validity of the software can't be established, click **OK**.
8. When the installation completes, restart Eclipse.

En <http://developer.android.com/sdk/installing/installing-adt.html> indica que podemos instalar o Plugin de dúas formas:

- **Troubleshooting ADT Installation**

If you are having trouble downloading the ADT plugin after following the steps above, here are some suggestions:

- If Eclipse can not find the remote update site containing the ADT plugin, try changing the remote site URL to use http, rather than https. That is, set the Location for the remote site to:  

```
http://dl-ssl.google.com/android/eclipse/
```
- If you are behind a firewall (such as a corporate firewall), make sure that you have properly configured your proxy settings in Eclipse. In Eclipse, you can configure proxy information from the main Eclipse menu in **Window** (on Mac OS X, **Eclipse**) > **Preferences** > **General** > **Network Connections**.

If you are still unable to use Eclipse to download the ADT plugin as a remote update site, you can download the ADT zip file to your local machine and manually install it:

1. Download the ADT Plugin zip file (do not unpack it):

Package	Size	MD5 Checksum
ADT-22.2.1.zip	14476845 bytes	97176754a1e6ad2e5e05f44dc7229e

2. Start Eclipse, then select **Help > Install New Software**.
3. Click **Add**, in the top-right corner.
4. In the Add Repository dialog, click **Archive**.
5. Select the downloaded ADT-22.2.1.zip file and click **OK**.
6. Enter "ADT Plugin" for the name and click **OK**.
7. In the Available Software dialog, select the checkbox next to Developer Tools and click **Next**.
8. In the next window, you'll see a list of the tools to be downloaded. Click **Next**.
9. Read and accept the license agreements, then click **Finish**.
10. When the installation completes, restart Eclipse.

If you get a security warning saying that the authenticity or validity of the software can't be established, click **OK**.

To update your plugin once you've installed using the zip file, you will have to follow these steps again instead of the default update instructions.

Unha descargando o paquete e logo instalalo dende Eclipse ...

- **Download the ADT Plugin**

1. Start Eclipse, then select **Help > Install New Software**.
2. Click **Add**, in the top-right corner.
3. In the Add Repository dialog that appears, enter "ADT Plugin" for the Name and the following URL for the Location:  

```
https://dl-ssl.google.com/android/eclipse/
```
4. Click **OK**.
5. In the Available Software dialog, select the checkbox next to Developer Tools and click **Next**.
6. In the next window, you'll see a list of the tools to be downloaded. Click **Next**.
7. Read and accept the license agreements, then click **Finish**.
8. When the installation completes, restart Eclipse.

If you have trouble acquiring the plugin security reasons).

If you get a security warning saying the authenticity or validity of the software can't be established, click **OK**.

**Configure the ADT Plugin**

... ou copiar o enlace ao repositorio no que se atopa o plugin ADT: <https://dl-ssl.google.com/android/eclipse/>. Usaremos esta forma.

```
admin@ubase: ~  
admin@ubase:~$ sudo /opt/eclipse/eclipse
```

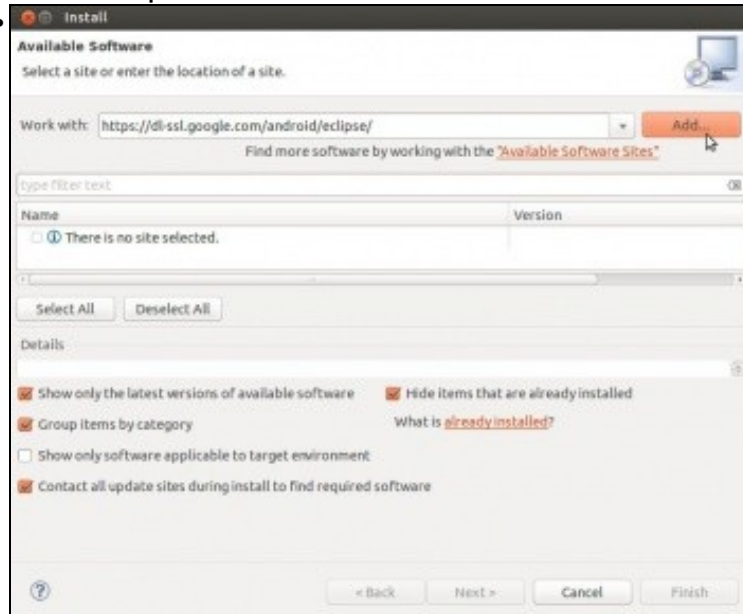
Lanzar eclipse como root.

- **Java EE - Eclipse**

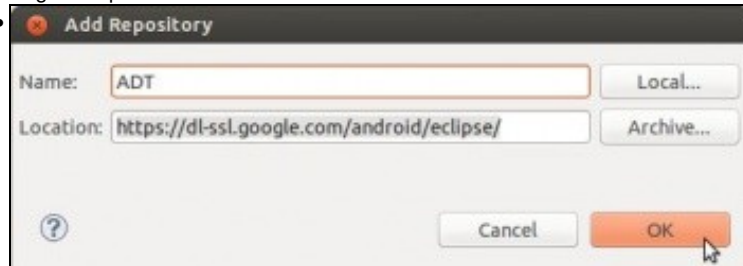




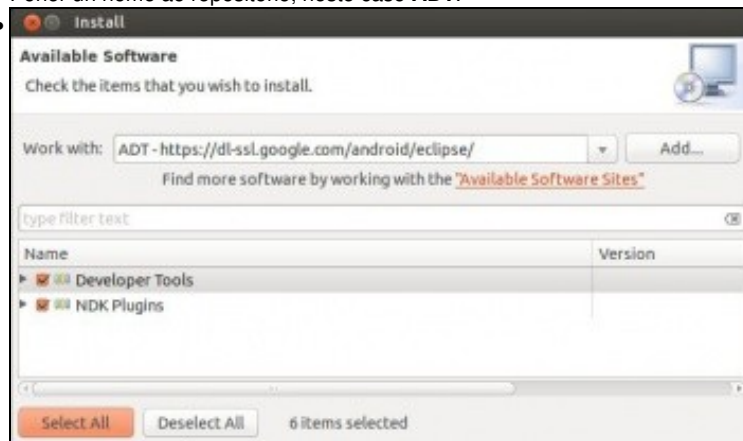
Premer en **Help -> Install New Software...**



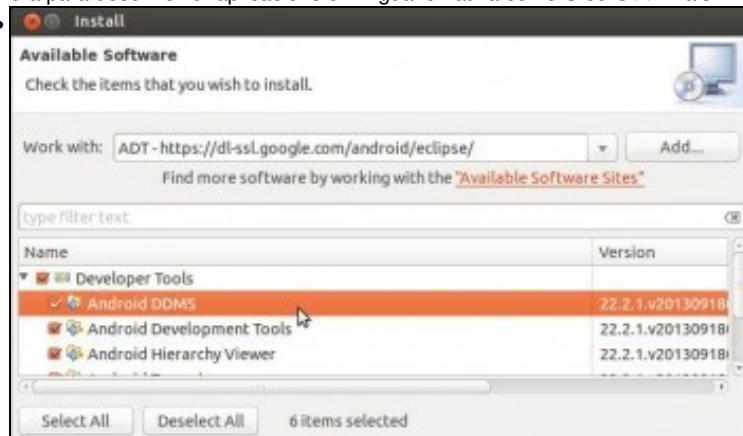
Pegar o repositorio ... Premer en **Add**.



Poñer un nome ao repositorio, neste caso **ADT**.



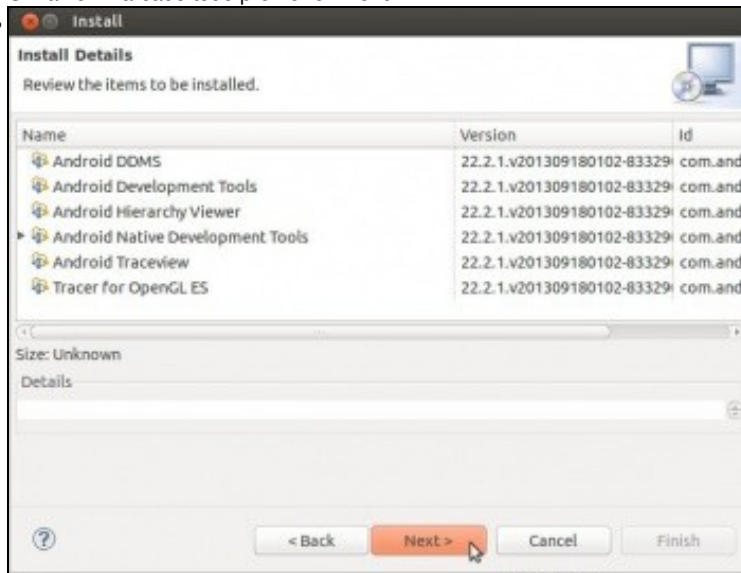
Amosa os compoñentes que se van instalar. Na última versión do ADT só hai un compoñente. O compoñente **NDK: Native Development Kit** era para desenvolver aplicacións en linguaxe nativa como C ou C++. Máis información: <http://developer.android.com/tools/sdk/ndk/index.html>



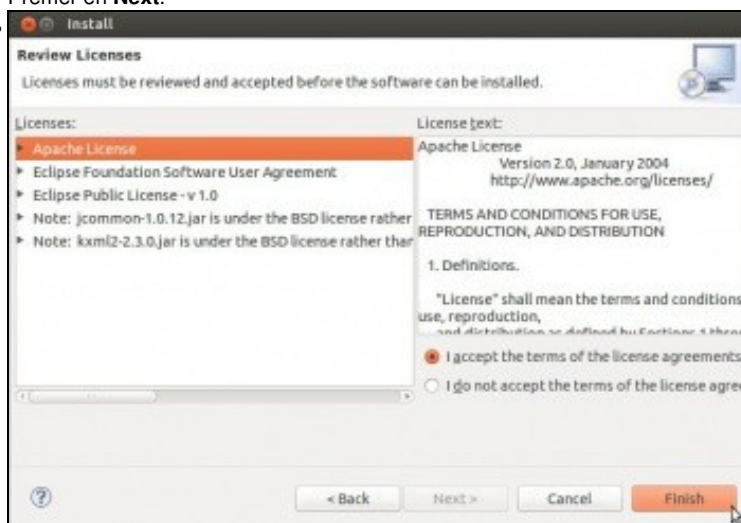
Dentro dos componentes de **Developer Tools** hai outros como **DDMS** que se verá nun apartado posterior.



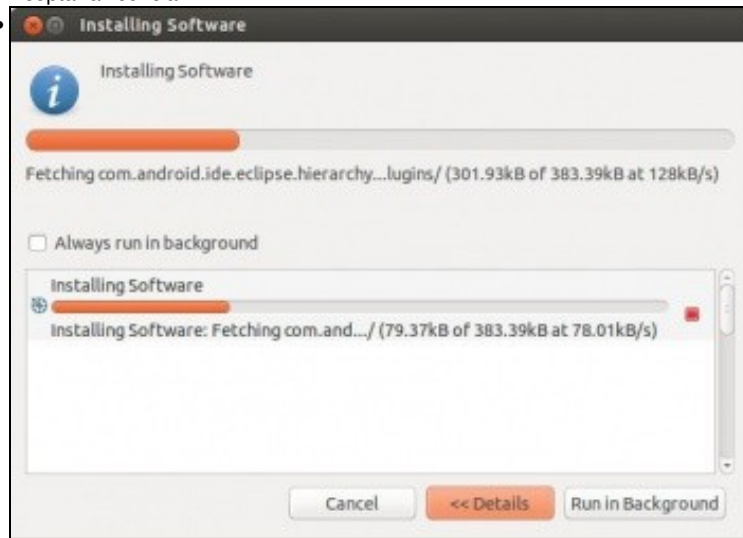
Unha vez marcado todo premer en **Next**.



Premer en **Next**.



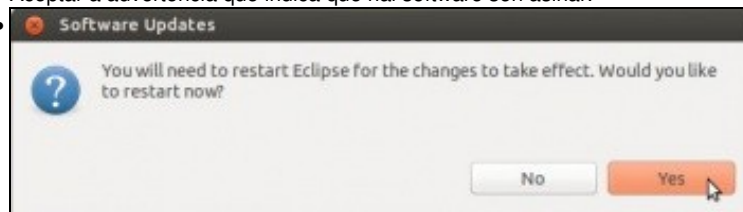
Aceptar a licencia.



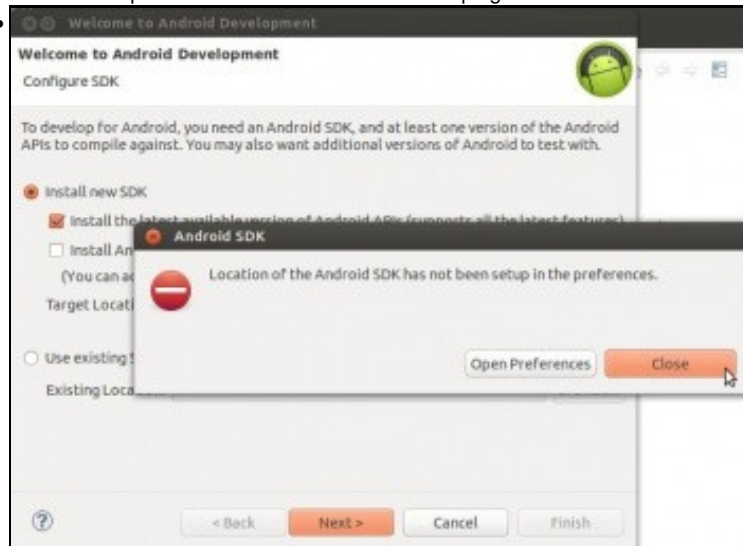
Proceso de descarga e instalación dos paquetes.



Aceptar a advertencia que indica que hai software sen asinar.



Reiniciar eclipse cando remate a instalación do plugin.



Advertencia que indica que non se atopa a ruta do SDK de Android. Pechar.

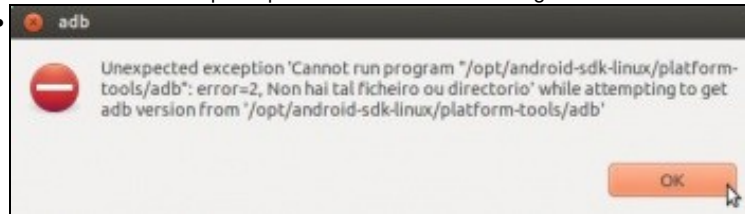


Unha vez instalado o Plugin ADT en Eclipse este permítenos que instalemos un novo SDK de Android descargándoo dende cero, ou usar un xa instalado. Escoller esta segunda opción e indicar a ruta **/opt/android-sdk-linux**

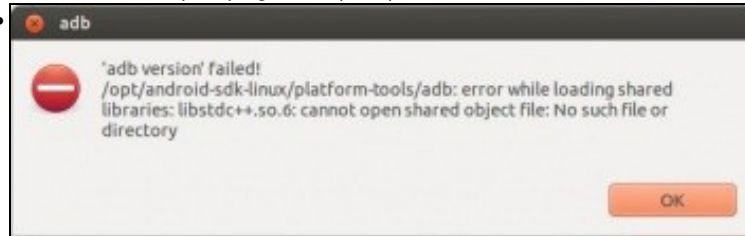




Indicar se se desexa participar nas estadísticas de Google.



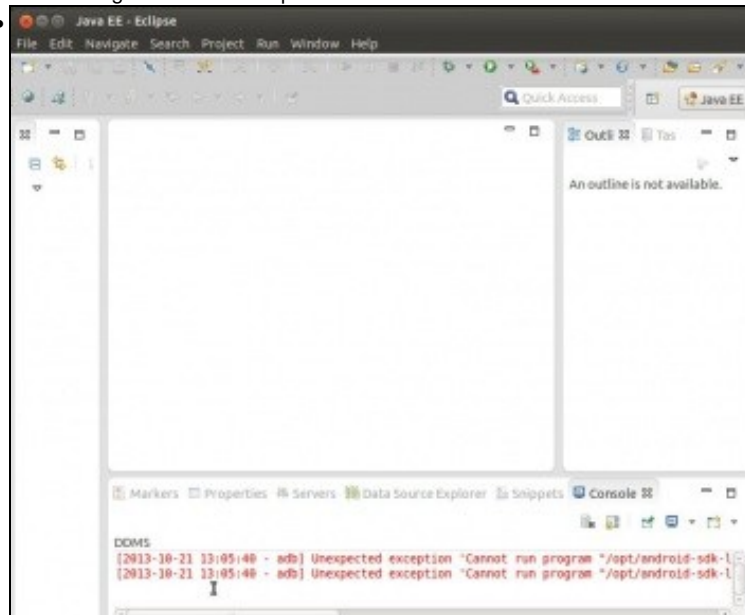
Na versión de Ubuntu de 13.10 de 64bits aparecen varios erros que fan mención o problema que se tiña antes co compoñente ADB. Este servizo é lanzado polo plugin ADT para poder usarse no DDMS como se verá máis tarde.



... aquí indica que precisa unha librería que non ten.



Entramos igualmente en Eclipse ...



E vemos dous erros na parte inferior avisando do mesmo.

## Librarías de 32 bits precisas para Ubuntu 13.10 Linux de 64 bits

- Se o lector está usando unha versión de Ubuntu de 32 bits xa pode pasar ao seguinte punto.
- Como se viu antes o SDK de Android era válido para as versións de 32bits e 64bits, pero se estamos nun SO de 64 bits, este non ten instaladas algunhas librarías de 32 bits que precisan algúns compoñentes do SDK, por exemplo o ADB.
- O **Android Debug Bridge (ADB)** é un comando que arranca un servidor que nos vai permitir comunicarnos e interactuar cos dispositivos físicos ou emulados.
- O ADB verase con profundidade nun apartado posterior, por agora so interesa que se poida executar.
- No enlace hai máis información: <http://developer.android.com/tools/help/adb.html>

- Instalación librería 32 bits

```
admin@ubase: ~  
admin@ubase:~$ /opt/android-sdk-linux/platform-tools/adb  
/opt/android-sdk-linux/platform-tools/adb: error while loading shared libraries:  
libstdc++.so.6: cannot open shared object file: No such file or directory  
admin@ubase:~$
```

Este compoñente atópase na ruta do SDK en: **/opt/android-sdk-linux/platform-tools/adb**. Ao executalo vese que nos indica un erro cunha librería.

```
admin@ubase: ~  
admin@ubase:~$ sudo apt-get install lib32stdc++6  
Lendo as listas de paquetes... Feito  
Construindo a árbore de dependencias  
Lendo a información do estado... Feito  
Os seguintes paquetes NOVOS hanse instalar:  
lib32stdc++6  
@ anovados, 1 instalados, Vanse retirar 0 e deixar 14 sen anovar.
```

Instalamos a librería, pero de 32 bits, como root: **sudo apt-get install lib32stdc++6**

```
admin@ubase: ~  
admin@ubase:~$ /opt/android-sdk-linux/platform-tools/adb  
Android Debug Bridge version 1.0.31  
  
-a - directs adb to listen on all interfaces for a connection  
-d - directs command to the only connected USB device  
-e - returns an error if more than one USB device is present.  
-e - directs command to the only running emulator.  
-e - returns an error if more than one emulator is running.  
-s <specific device> - directs command to the device or emulator with the given  
serial number or qualifier. Overrides ANDROID_SERIAL  
environment variable.  
-p <product name or path> - single product name like 'sooner', or  
a relative/absolute path to a product  
out directory like 'out/target/product/sooner'.  
If -p is not specified, the ANDROID_PRODUCT_OUT  
environment variable is used, which must  
be an absolute path.  
-H - Name of adb server host (default: localhost)  
-P - Port of adb server (default: 5037)  
devices [-l] - list all connected devices  
{'-l' will also list device qualifiers}  
connect <host>[:<port>] - connect to a device via TCP/IP  
Port 5555 is used by default if no port number is specified.  
disconnect [<host>[:<port>]] - disconnect from a TCP/IP device.  
Port 5555 is used by default if no port number is specified.  
Using this command with no additional arguments  
will disconnect from all connected TCP/IP devices.  
  
device commands:  
adb push <local> <remote> - copy file/dir to device  
adb pull <remote> [<local>] - copy file/dir from device  
adb sync [<directory> ] - copy host->device only if changed  
{'-l' means list but don't copy}  
{see 'adb help all'}  
adb shell - run remote shell interactively  
adb shell <command> - run remote shell command
```

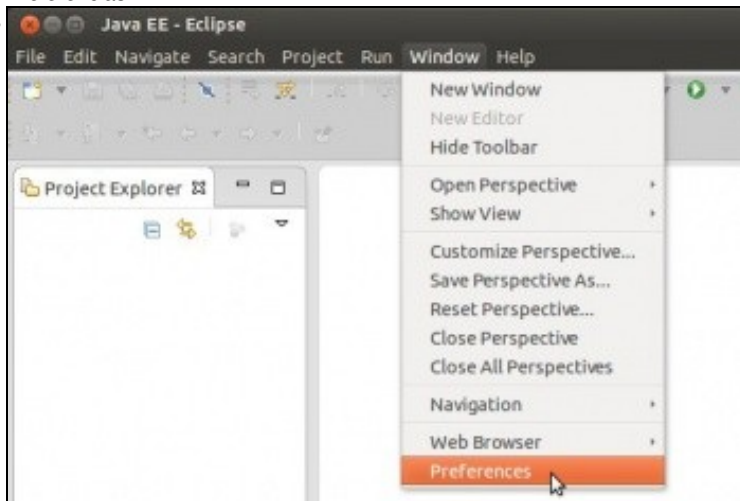
Executamos de novo o comando: `/opt/android-sdk-linux/platform-tools/adb` e xa amosa a axuda do comando.

- Agora xa podemos iniciar Eclipse que non se terá ningún problema co ADB nunha versión linux de 64 bits.
- Aínda hai que instalar algunha librería máis de 32 bits, pero imos esperar a que xurda o erro.

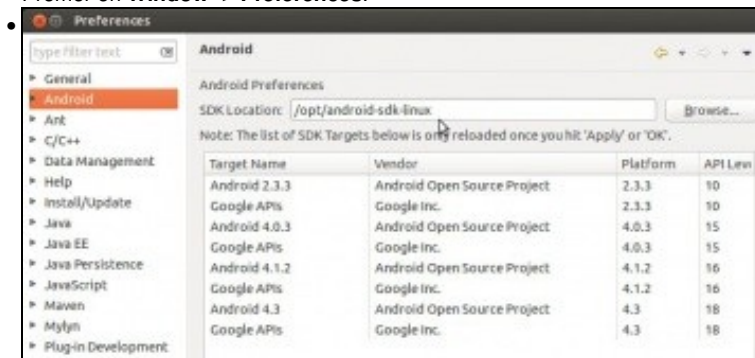
## Configuración Eclipse e plugin ADT. Actualizacións

- Agora xa podemos iniciar eclipse sen problemas e imos ver:
  - ◆ Parámetros de configuración de Eclipse.
  - ◆ Como actualizar Eclipse ou o SDK de Android.
  - ◆ Que elementos engadiu o plugin ADT á Perspectiva Java de Eclipse.
    - ◇ Unha **Perspectiva** é unha combinación de pantallas para facilitar o desenvolvemento de aplicacións de distintos tipos.
    - ◇ Aconséllase que se lle bote un ollo a: "**Mi primera hora con Eclipse**":  
[http://ubuntulife.files.wordpress.com/2008/03/intro\\_eclipse\\_espanol.pdf](http://ubuntulife.files.wordpress.com/2008/03/intro_eclipse_espanol.pdf)

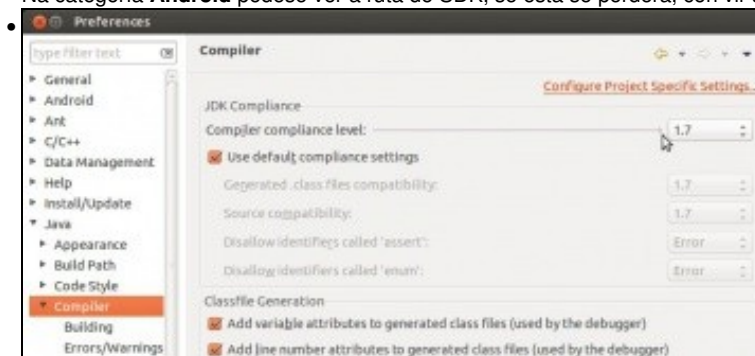
- Preferencias



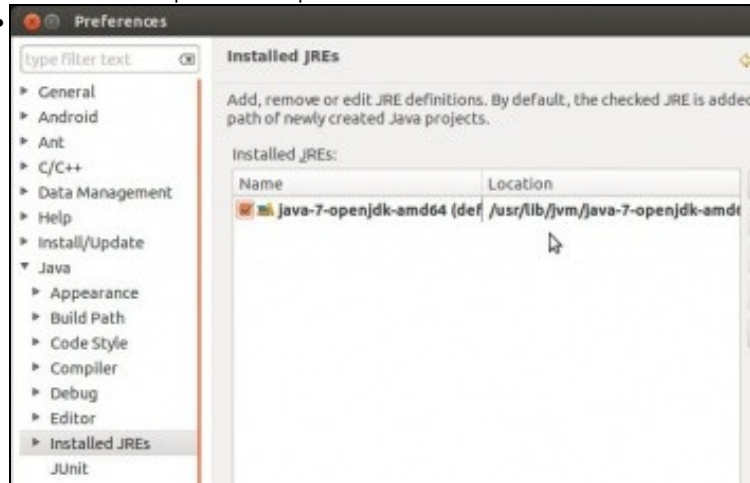
Premer en **Window -> Preferences**.



Na categoría **Android** pódese ver a ruta do SDK, se esta se perdera, con vir a esta preferencia xa se podería configurar.

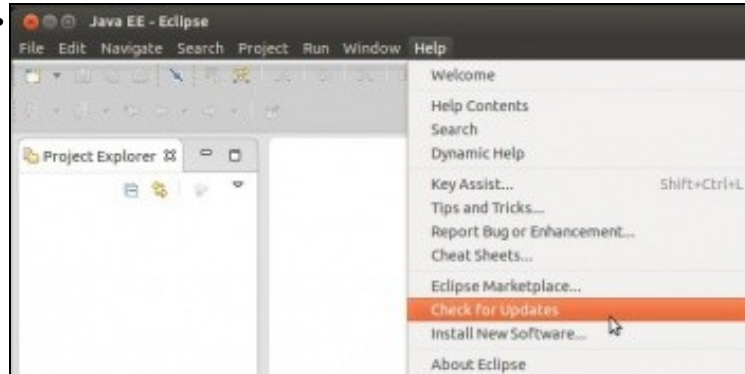


A versión do compilador JDK que vai usar o IDE.

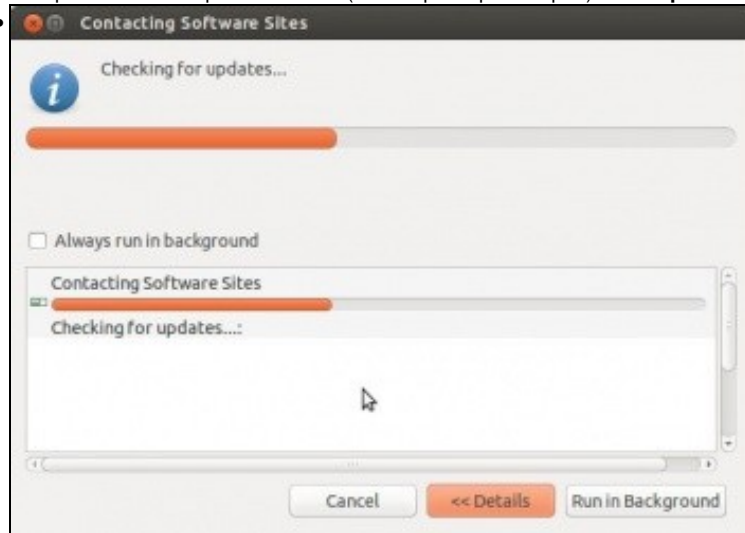


A versión do JRE e a súa ruta.

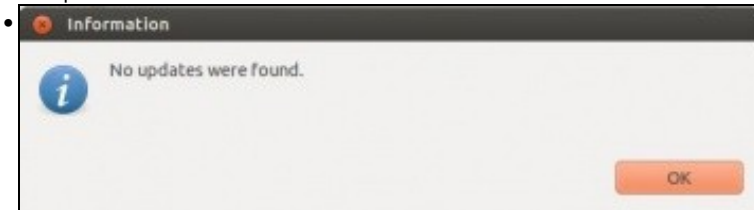
### Actualizacións



Hai que lanzar o Eclipse como root (sudo /opt/eclipse/eclipse). En **Help** -> **Check for Updates**.

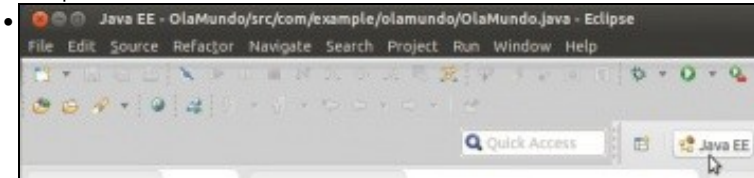


Comprobando se hai actualizacións do IDE.

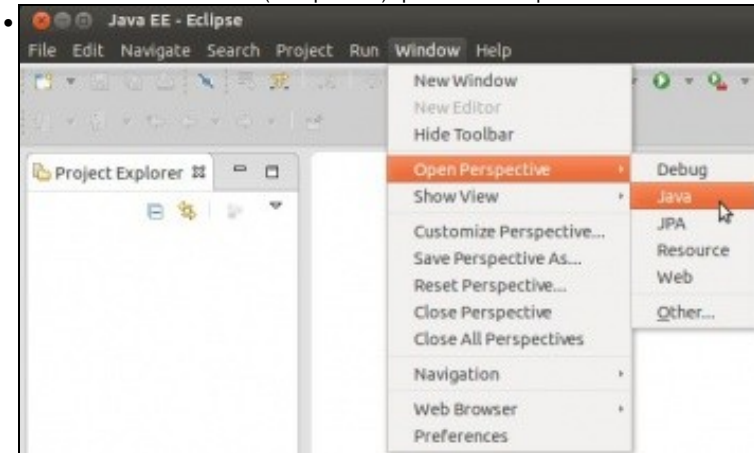


Non hai actualizacións.

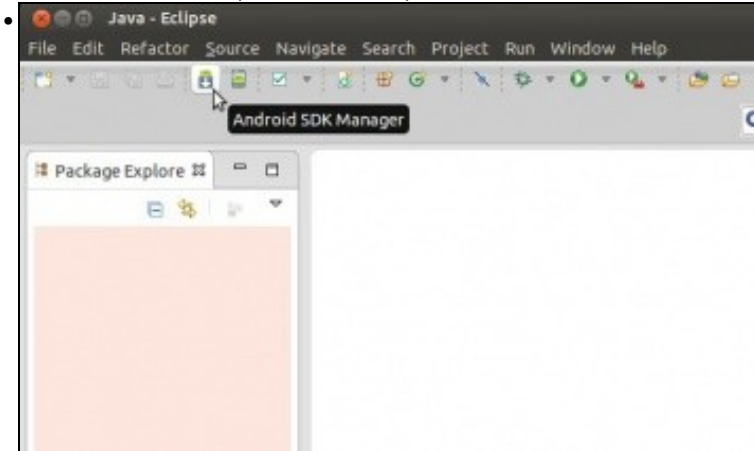
- Perspectiva Java



A combinación de ventás (Perspectiva) que se ve é a que se denomina Java EE

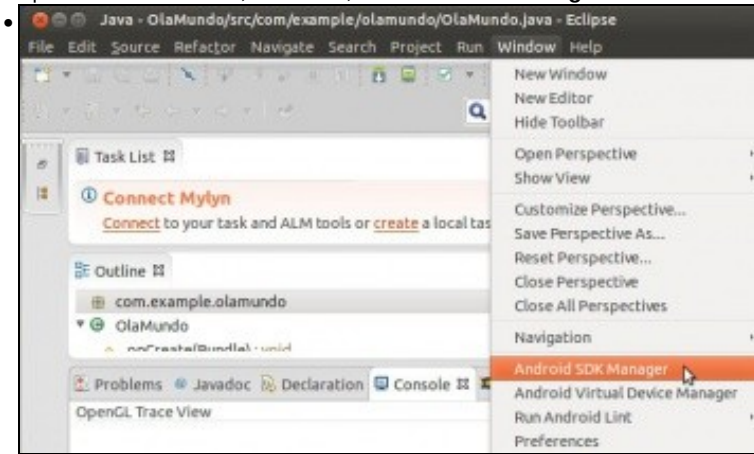


Pero ADT instalou compoñentes na Perspectiva Java. Premer en **Window -> Open Perspective -> Java**

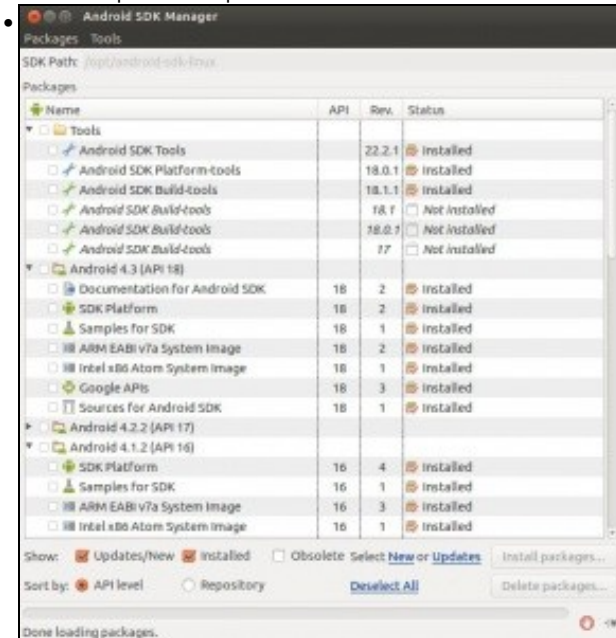




Aparecen novas iconas, entre elas, o **Android SDK Manager** e o Xestor de Dispositivos Virtuais.



O mesmo pasa se se preme en **Window**



Dende o propio Eclipse, o plugin proporciona un enlace ao comando que chama ao **Xestor de SDK de Android**, que estaba na ruta: **/opt/android-sdk-linux/tools/android sdk**. Dende aquí poderíanse instalar/desinstalar compoñentes e incluso actualizalos. Pero hai que iniciar o Eclipse como **root**: **sudo /opt/eclipse/eclipse**. Non o pode facer un usuario normal (ben, salvo que en /opt/android-sdk-linux se lle dean permisos de escritura a todo o mundo).

## Carpetas de traballo

• As carpetas que se obteñen finalmente pertencen a 2 tipos:

◆ **Carpetas de programas:**

- ◇ **/opt/eclipse**: Eclipse + plugin ADT
- ◇ **/opt/android-sdk-linux**: Android SDK

◆ **Carpetas de configuración de usuario**

- ◇ **/home usuario/.eclipse**: Carpeta oculta. Contén arquivos de configuración relativos a ese usuario e ao IDE Eclipse.
- ◇ **/ruta/workspace**: Carpeta indicada polo usuario onde se van gardar os proxectos creados no IDE Eclipse.
- ◇ **/home usuario/.android**: Carpeta oculta. Contén arquivos de configuración relativos ao SDK de Android e onde se van gardar os emuladores de dispositivos móbiles (AVDs) que cree o usuario.

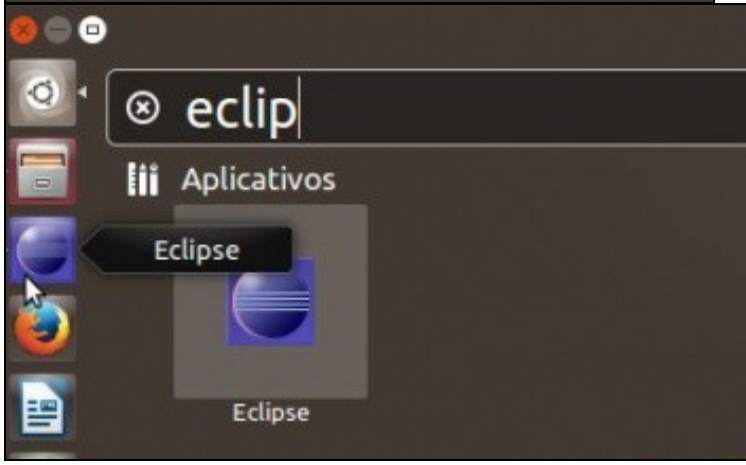
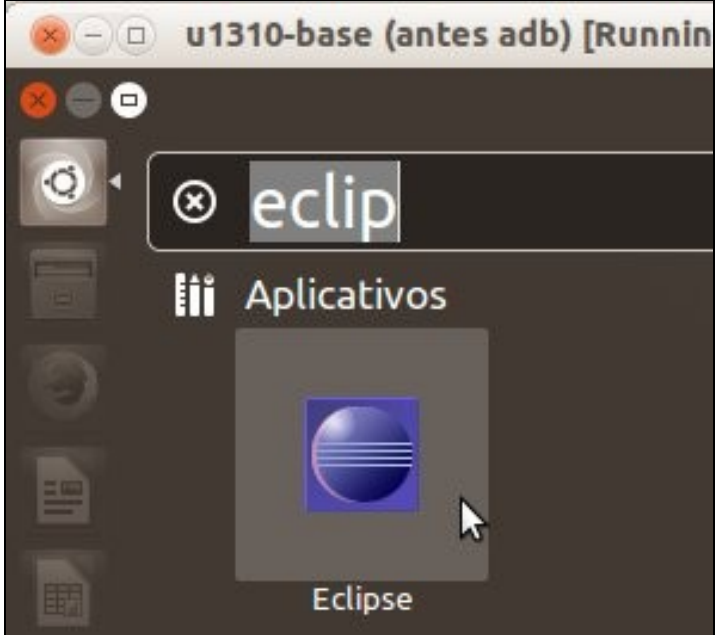
• Dende a sesión do usuario co que se estaba no sistema se estivo executando Eclipse e o SDK de Android como root, aconsellamos, para que non haxa problemas con permisos, que se borren as tres carpetas asociadas á configuración do usuario. Puidera ser o caso de que xa non existiran esa carpetas, pero é mellor asegurarse.

```
admin@ubase: ~
admin@ubase:~$ sudo rm .android -r
rm: non se pode eliminar ".android": Non hai tal ficheiro ou directorio
admin@ubase:~$
admin@ubase:~$ sudo rm .eclipse -r
rm: non se pode eliminar ".eclipse": Non hai tal ficheiro ou directorio
admin@ubase:~$
admin@ubase:~$ sudo rm workspace/ -r
admin@ubase:~$
```

### Fixar a icona ao Lanzador

- Unha forma de facilitar o acceso a eclipse pode ser fixando a icona ao Lanzador.
- Hai varias formas de facelo, unha delas é a seguinte, unha vez que se ten o localizado na Lente de Inicio, arrastralo ate a barra do Lanzador.

- Fixar icona ao Lanzador



-- Ángel D. Fernández González e Carlos Carrión Álvarez -- (2015).