

# Dispositivos Virtuais de Android: AVD

## Sumario

- 1 Introducción
- 2 Situación de partida
- 3 O Xestor de Dispositivos Virtuais de Android
- 4 Crear un AVD
- 5 Ficheiros e configuración dun AVD
- 6 Iniciar e manipular un AVD
  - ◆ 6.1 Cambio de parámetros do AVD
- 7 Intercambio de AVDs
- 8 Aceleración HW
  - ◆ 8.1 Uso en Linux procesador Intel
  - ◆ 8.2 Aceleración por HW en Windows
- 9 Lanzar un dispositivo dende a consola

## Introdución

- Android proporciona un emulador que permite executar e probar aplicacións sen ter que usar un dispositivo real.
- Para iso ofrece a posibilidade de crear **Dispositivos Virtuais de Android (AVD - Android Virtual Devices)**.
- Nos seguintes enlaces dispónse de máis información.
  - ◆ Xestor AVD: <http://developer.android.com/tools/help/avd-manager.html>
  - ◆ Xestión AVDs dende a liña de comandos: <http://developer.android.com/tools/devices/managing-avds-commandline.html>
  - ◆ Emulador Android: <http://developer.android.com/tools/help/emulator.html>
  - ◆ Usando o emulador: <http://developer.android.com/tools/devices/emulator.html>
- No presente caso dá igual traballar con calquera SO, por tanto realizaranse as capturas en Ubuntu e se hai diferencias significativas nos outros SOs, estas serán indicadas.

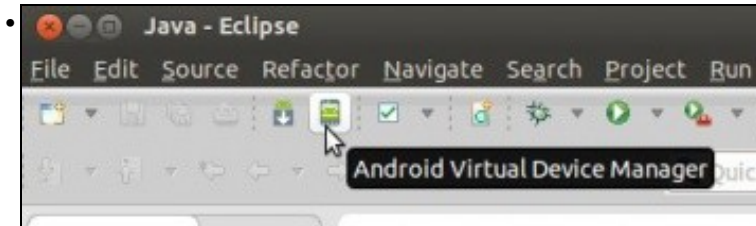
## Situación de partida

- Por se quedaron trazas da instalación ao executar como root Eclipse e a utilidade tools/android aconséllase que se sigan os seguintes pasos:
  - ◆ Eliminar carpetas de usuario en Ubuntu: [Instalación en Ubuntu#Carpetas de traballo](#)
  - ◆ Acceso rápido a Eclipse en Ubuntu: [Instalación en Ubuntu#Fixar a icona ao Lanzador](#)
  - ◆ Acceso rápido a Eclipse en Windows: [Instalación en Windows#Fixar a icona á barra de tarefas](#)

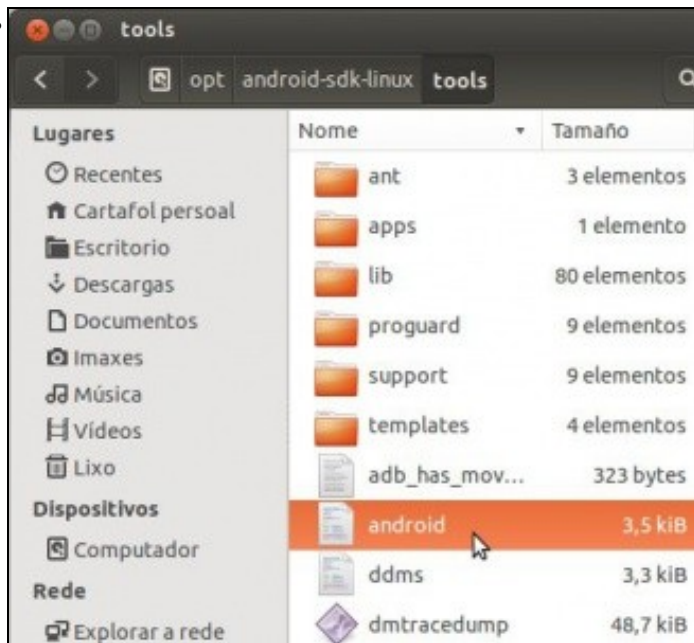
## O Xestor de Dispositivos Virtuais de Android

- O Xestor de Dispositivos Virtuais de Android pode ser iniciado de varias formas:

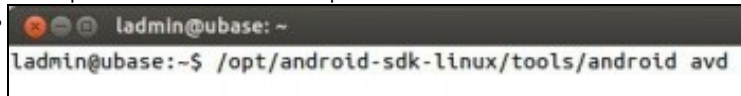
- O Xestor de AVDs



Dende a perspectiva de Java de Eclipse premendo no botón **Android Virtual Device Manager**. Tamén dende o menú **Window -> Android Virtual Device Manager**. *Buscar esta funcionalidade en Android Studio.*

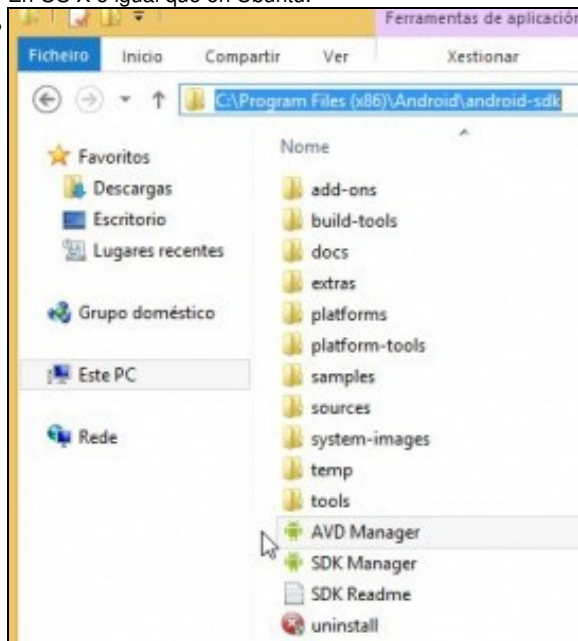


Na carpeta **tools** do SDK está a aplicación **android**.

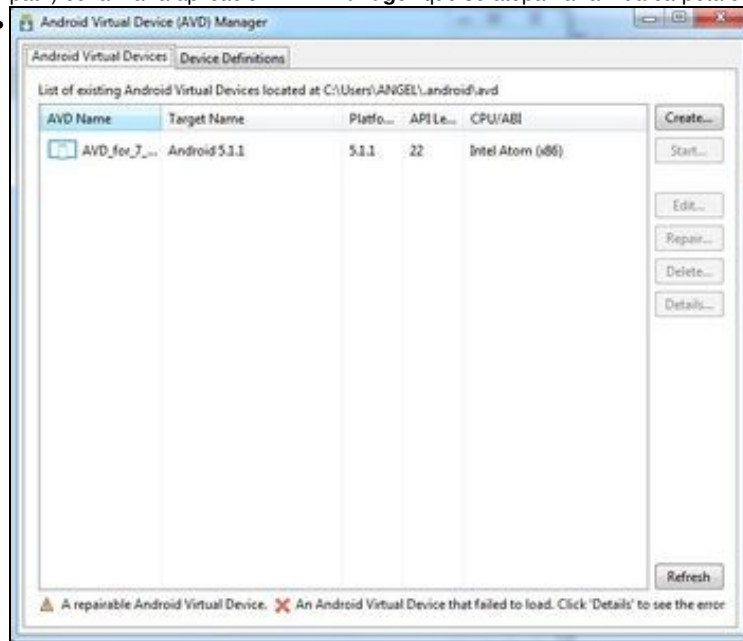


Dende calquera SO pódese lanzar **android avd**. Neste caso hai que poñer a ruta completa pois non está no path **/opt/android-sdk-linux/tools**.

En OS X é igual que en Ubuntu.



En **Windows** pódese lanzar dende Eclipse, dende a consola como no caso anterior (mellor situarse no directorio tools, ou meter entrada no path) ou lanzar a aplicación **AVD manager** que se atopa na raíz da carpeta do SDK.



Calquera dos métodos anteriores abrirá o **Xestor de Dispositivos Virtuais de Android (AVD)**. Observar como os dispositivos se van gardar en: **/home usuario/.android/avd**, isto en calquera dos 3 SOs.

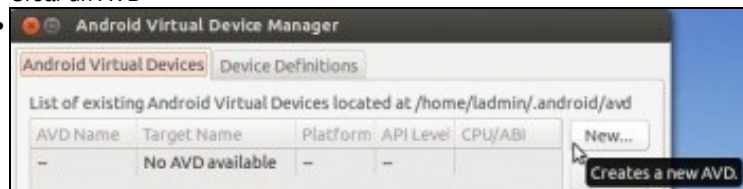
## Crear un AVD

- A continuación vaise crear un AVD:
  - ◆ Co seu dispositivo asociado
  - ◆ CPU
  - ◆ Versión de Android
  - ◆ Memoria
  - ◆ Tarxeta de memoria, etc.

**NOTA:** As seguintes imaxes, como xa indicamos, están baseadas na versión API 18, pero vos tedes que escoller a API 28 como está indicado.

**Edición 2020:** Nos traballaremos coa API 28. Isto estará indicado nas imaxes nas que se amosa outra API diferente.

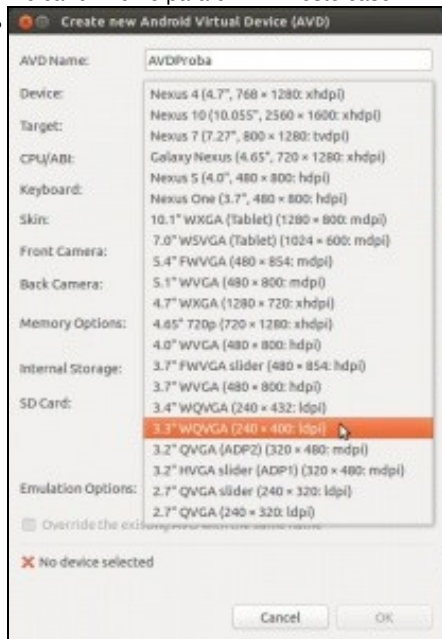
- Crear un AVD



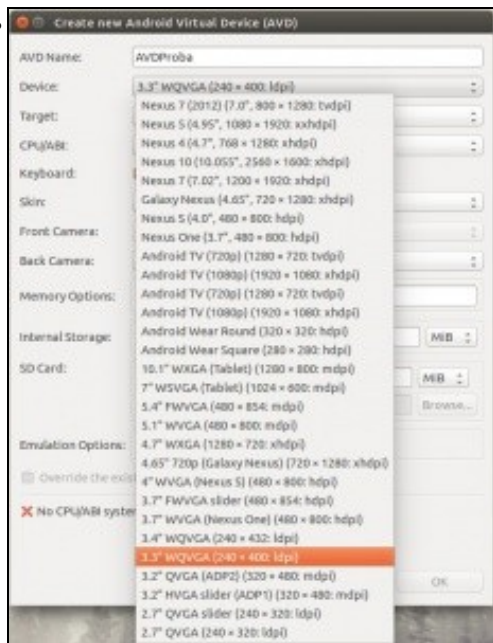
Premer en **New**.



Indicar un nombre para o AVD: neste caso **AVDProba**.



Seleccionar o dispositivo, neste caso seleccionouse un con tamaño de pantalla 3,3\"/>



No caso de ter APIs L e Wear podemos seleccionar dispositivos que traballan con esas versións do SO.



Seleccionar a versión de Android que correrá no AVD. Dependendo das opcións escollidas anteriormente aparecerán diferentes versións.

**Edición 2020:** Nos escolleremos: **API 28.**



Seleccionar a CPU, nesta caso unha Intel, que nas probas realizadas vai, como mínimo, o dobre de rápido que un procesador ARM. Instalar un procesador Intel permite que se instale un driver para acelerar por HW o AVD, aínda que ese Driver na última versión de MAC OS X (Maverick) da problemas. Isto verase máis adiante.

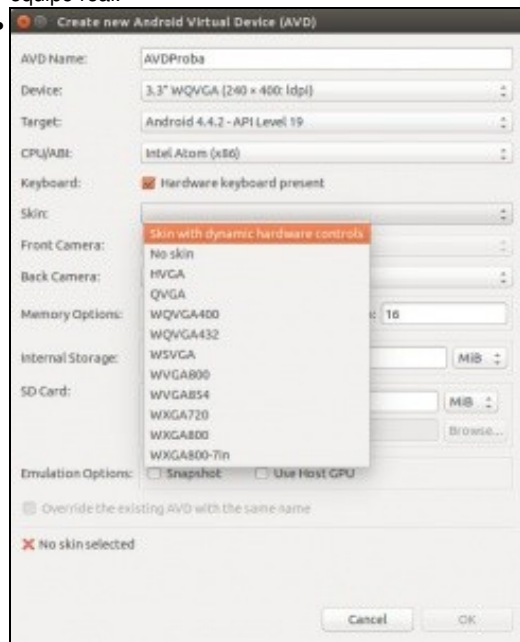


Se se escollera unha API 16 (Android 4.1.2) permitiríase traballar tamén cun procesador MIPS.

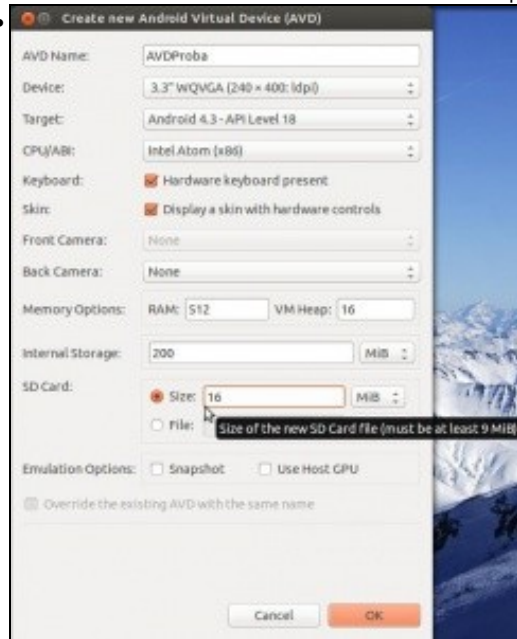


Volvemos á API 16 e ao procesador Intel.

Observar que está marcado que se use o teclado do ordenador, senón usaríase un teclado en pantalla ao igual que calquera dispositivo real. E observar que está marcado que o dispositivo amose unha **botonera** como calquera outro dispositivo real, para poder apagar, volume, etc. Hai dispositivos que permiten ter unha cámara frontal, non é o caso deste. Para a cámara traseira podemos simular unha ou que use a do equipo real.



No caso do novo SDK devemos escolher o **SKIN** co que se vai amosar o Hardware do teléfono (**Botonera**).



Configurar como mínimo 9 MiB (MebiByte) para a tarxeta SD (SD Card).

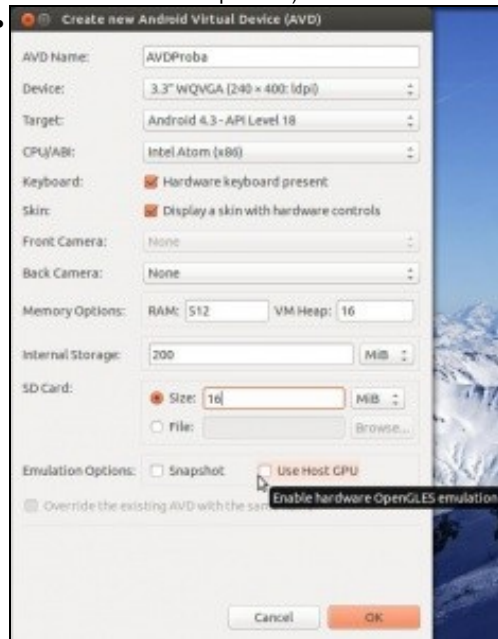




Esta tamén pode ser lida dende un ficheiro como se explica despois.



Indicar se se desexa que cando se apague o dispositivo faga unha instantánea (Snapshot), para que, cando se inicie, arranque dende a instantánea moito máis rápido. Pero esta opción consume moito máis espazo en disco. (Pódese ver o tamaño do dispositivo en `home/.android/avd/dispositivo`)



Indicar se se desexa que o AVD faga uso do OpenGL da tarxeta gráfica.



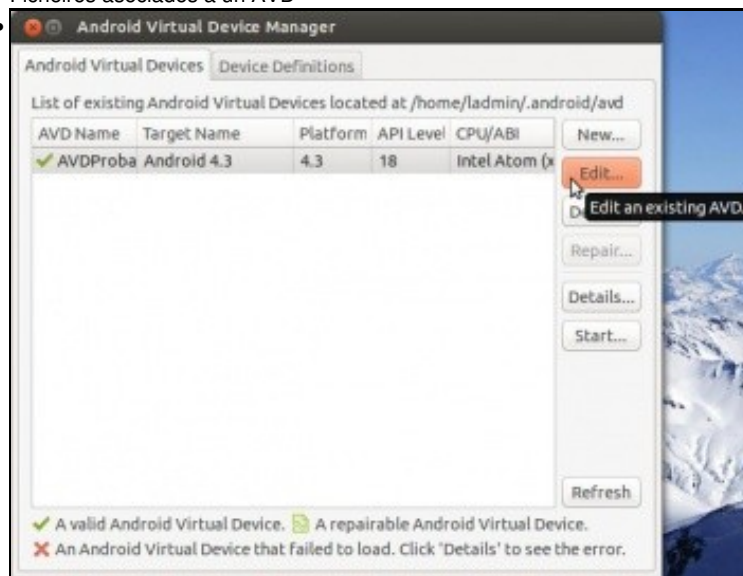
A configuración no noso AVD de proba.

- A continuación vanse realizar distintos inicios co este AVD.
- Como cada quen pode ter un equipo distinto ao equipo co que se elaborou o material, o lector pode cambiar/axustar os parámetros ás súas circunstancias para favorecer que o AVD vai o máis rápido posible.

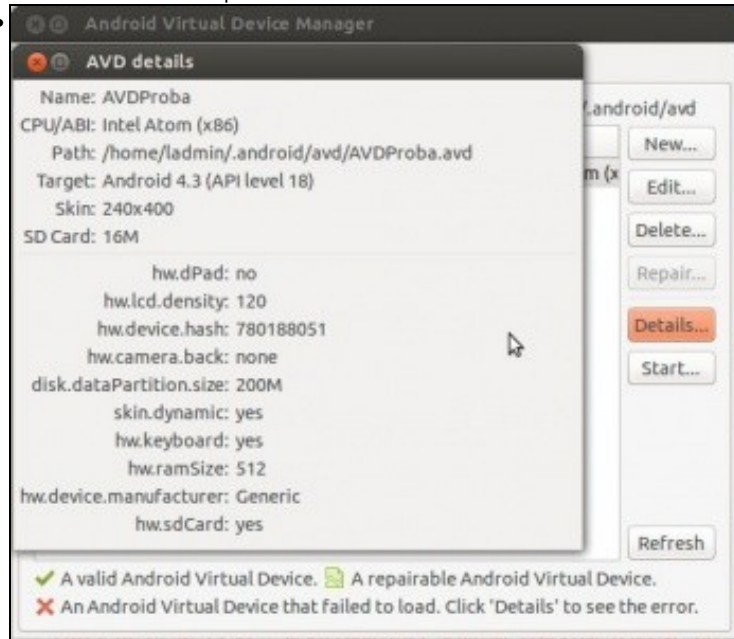
## Ficheiros e configuración dun AVD

- A continuación imos observar os ficheiros asociados a un AVD e o seu comportamento.
- No seguinte enlace pódese atopar máis información ao respecto: <http://developer.android.com/tools/devices/emulator.html#diskimages>

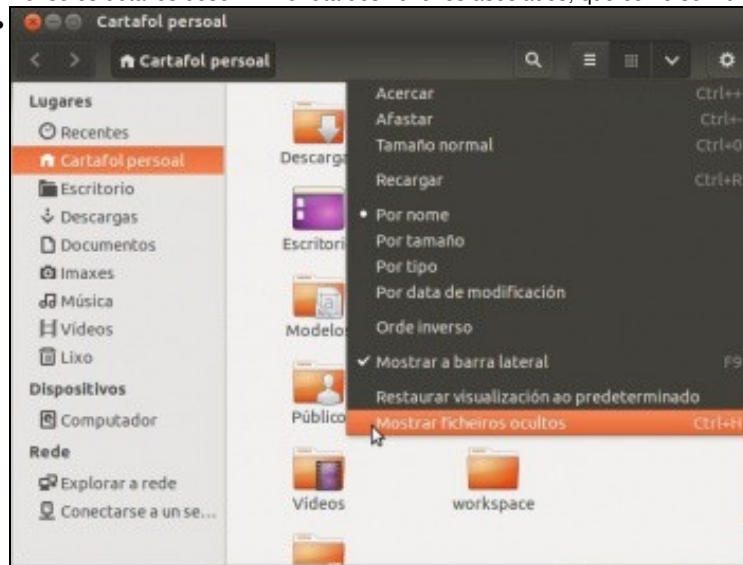
- Ficheiros asociados a un AVD



Seleccionar un AVD e premer en **Edit**.



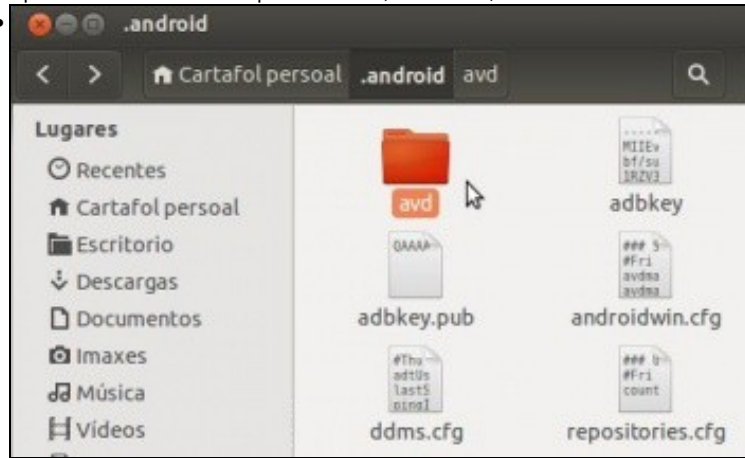
Vense os detalles dese AVD e ruta dos ficheiros asociados, que como se indicaba enriba está en: **/home usuario/.android/avd**.



Ir ao home do usuario. Lembrar que en Linux as carpetas que comezan por punto están ocultas. Premer **CTRL+H** ou premer en **Mostrar os ficheiros ocultos**.



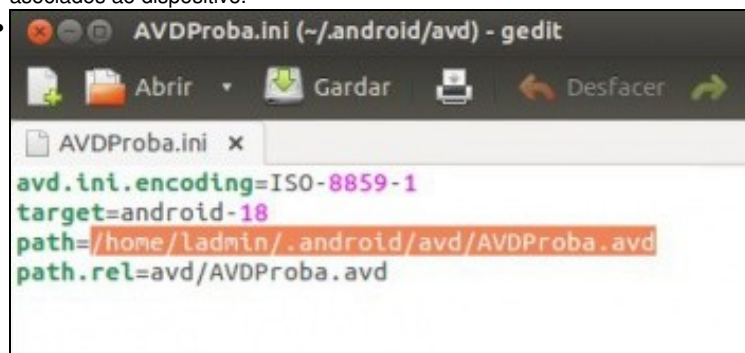
Aparecerá os ficheiros/carpetas ocultas, entre elas, **.android**.



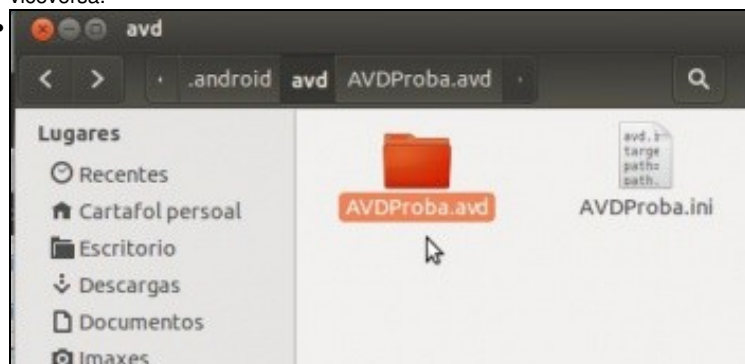
Dentro está a carpeta **avd** e ...



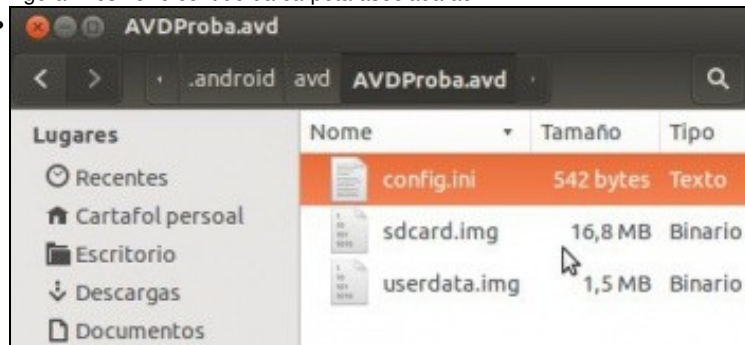
... dentro de **avd** o ficheiro de configuración asociado ao dispositivo AVDProba: **AVDProba.ini** e unha carpeta que contén os ficheiros asociados ao dispositivo.



Observar a ruta ao ficheiro AVD. Este é moi interesante porque deste xeito pódense intercambiar AVDs entre usuarios, e simplemente hai que modificar a ruta neste ficheiro, adaptándoa ao novo escenario de destino. Serve por exemplo para pasar unha MV a un alumno ou viceversa.



Agora imos ver o contido da carpeta asociada ao AVD.



Hai 3 ficheiros: un de configuración, outro asociado ao usuario (**userdata.img**) onde se vai gardar todo canto faga o usuario no AVD, de modo que como se verá despois este ficheiro pode *limparse* e un terceiro ficheiro que é a SD Card.

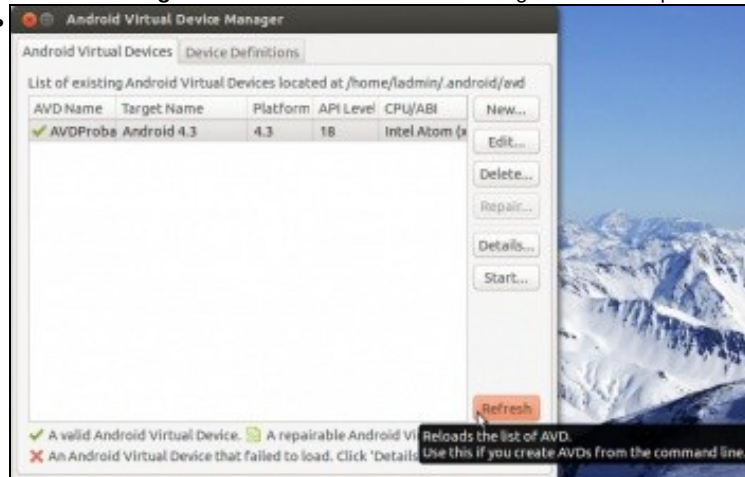
Esta SD Card podería ser usada como ficheiro para outro AVD.

Pódense crear ficheiros SD Card co comando **tools/mksdcard**, máis información en:

<http://developer.android.com/tools/devices/emulator.html#sdcard>



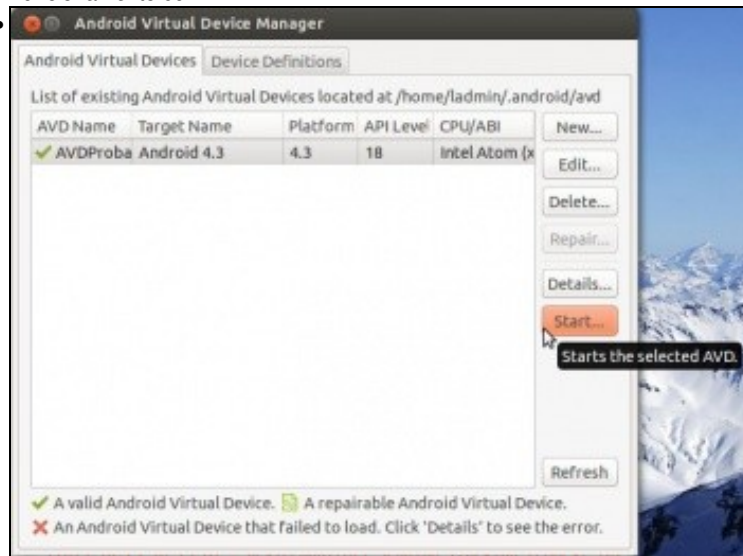
O ficheiro **config.ini** contén información sobre a configuración do dispositivo.



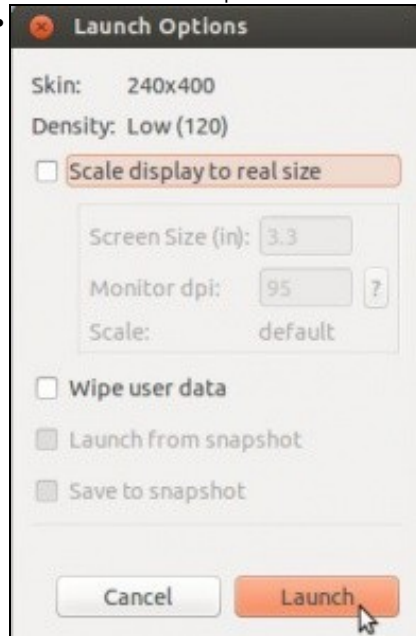
Un último botón **Refresh** no xestor de AVDs permite actualizar a lista de dispositivos por se este foron creados dende a consola ou copiados doutro ordenador ao directorio `.android/avd`.

## Iniciar e manipular un AVD

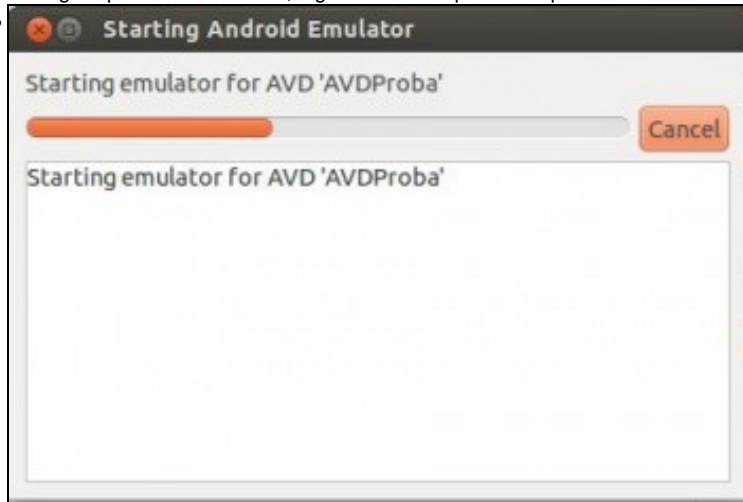
- Antes de instalar ningunha aplicación no dispositivo imos observar o funcionamento do mesmo.
- Funcionamento dun AVD



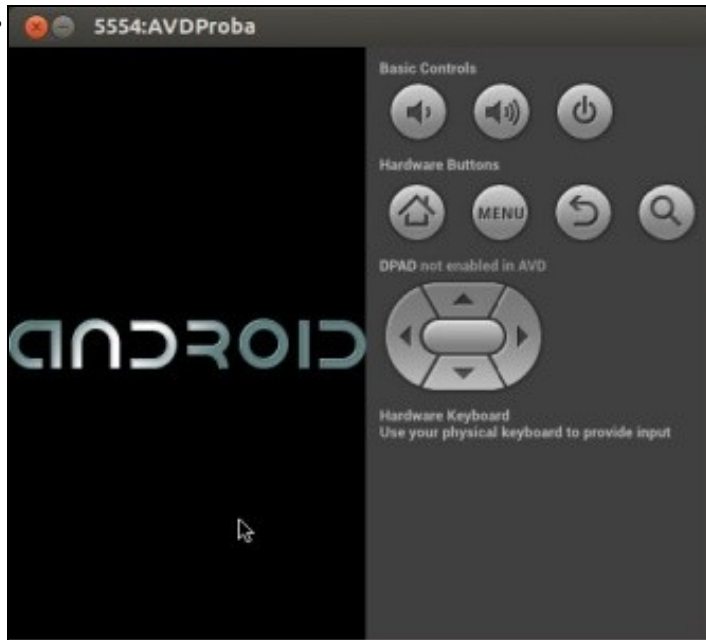
Seleccionar un AVD e premer en **Start**.



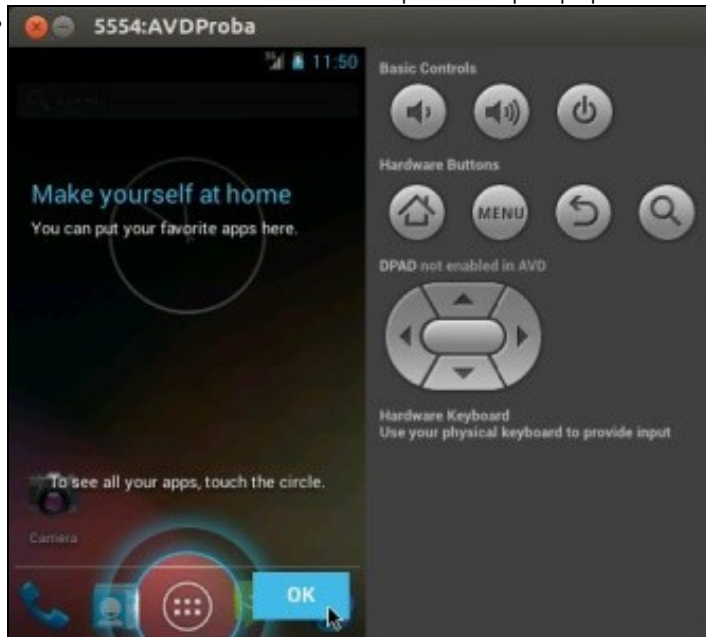
Por agora premer en **Launch**, logo veranse as posibles opcións de inicio.



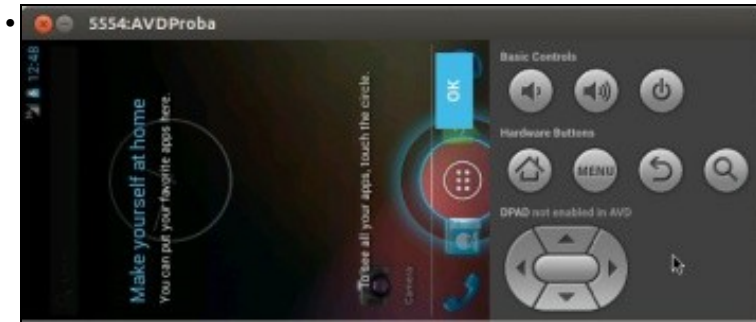
Inicio de AVD



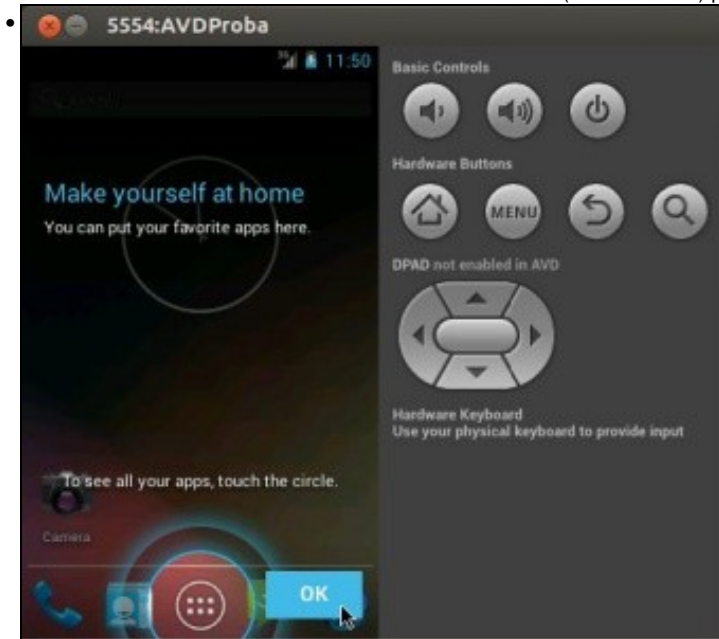
Dispositivo acendéndose e coa botonera HW como un dispositivo real: volume, apagado, Home, Menú, volver, etc. E fixarse que está activado o uso do teclado HW. Pode ser que dea tempo a preparar e tomar un café ... mentres carga.



O AVD iniciado.



Premendo CTRL + F11 ou a tecla 7 no teclado numérico (Desabilitado) pódese rotar o dispositivo a paisado. Volver a premer CTRL + F11.



Premer en OK e entrar.





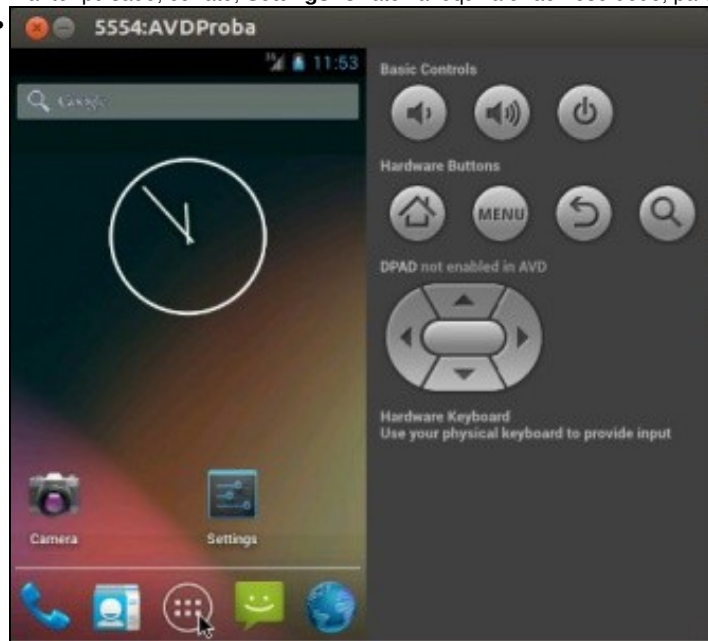
Premer en Aplicacions ...



Aviso que indica que se mantemos premda unha aplicación está pasará á pantalla de inicio.



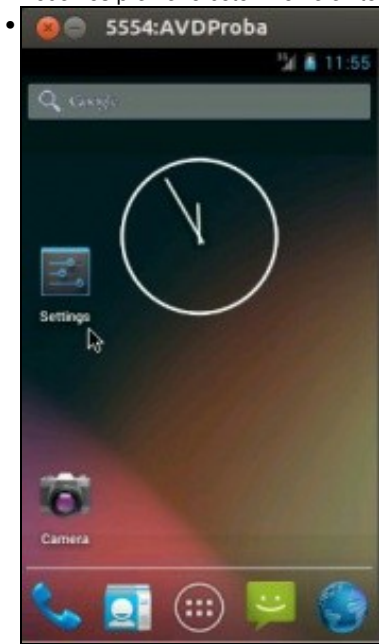
Manter pulsado, co rato, **Settings**. O rato vai equivaler ao noso dedo, para arrastrar, desprazar a pantalla, etc.



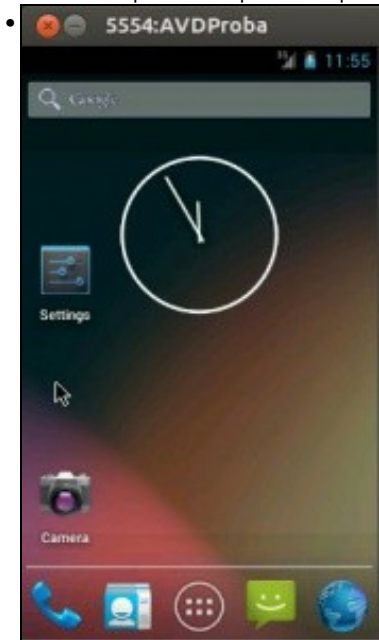
Settings na pantalla de inicio. Volver a premer en Aplicacións



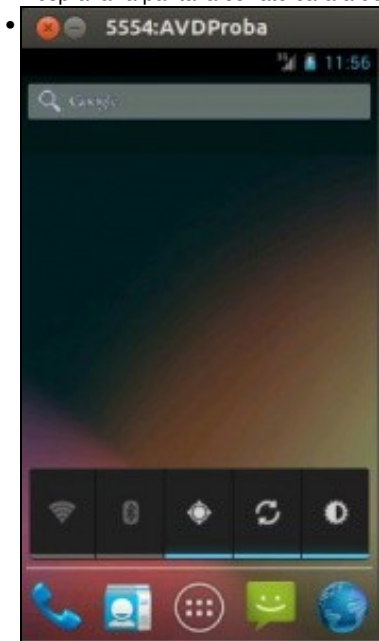
Podemos premer o botón Home en todo momento para regresar á pantalla de Inicio



Pódese desprazar as aplicacións pola pantalla.



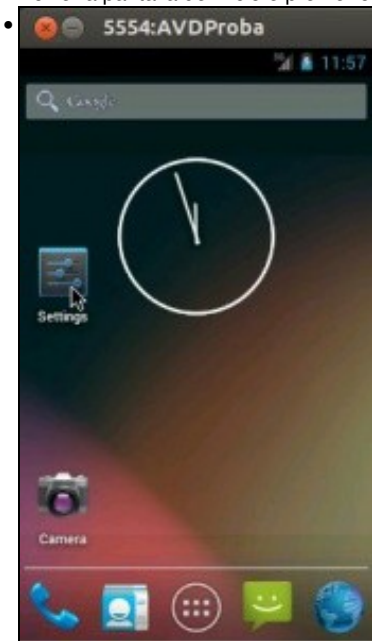
Desprazar a pantalla co rato cara á dereita



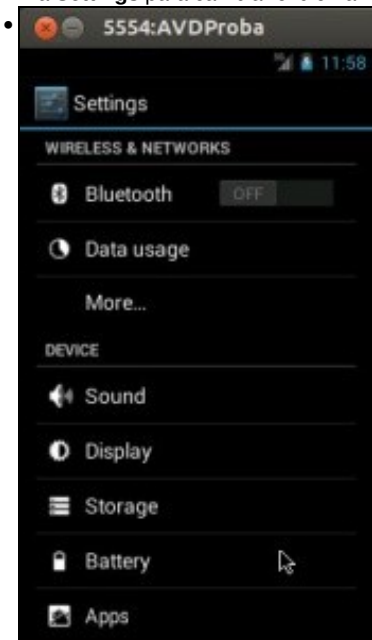
Vese outra das pantallas de inicio.



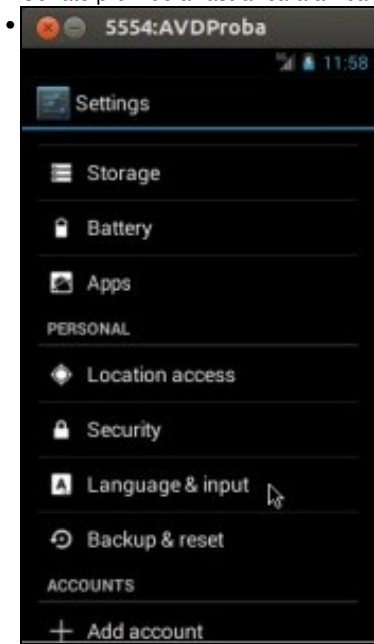
Volver á pantalla de inicio e premer o botón HW **Menú**, aparece o menú asociado á aplicación que estea aberta nese intre, se é o caso.



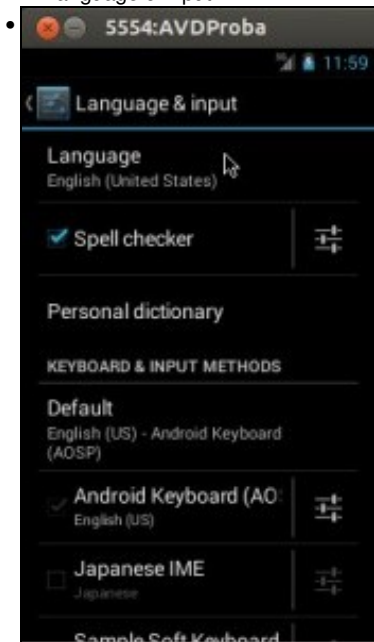
Ir a **Settings** para cambiar o idioma



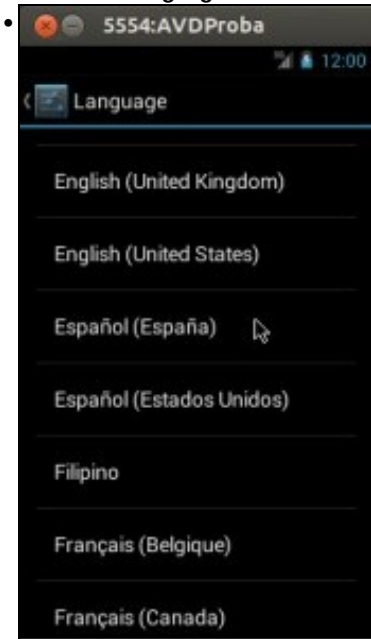
Co rato premido arrastrar cara arriba até que apareza ...



... Language & input



Premer en **Language**



Seleccionar **Español**



Volver á pantalla anterior co botón Hardware.

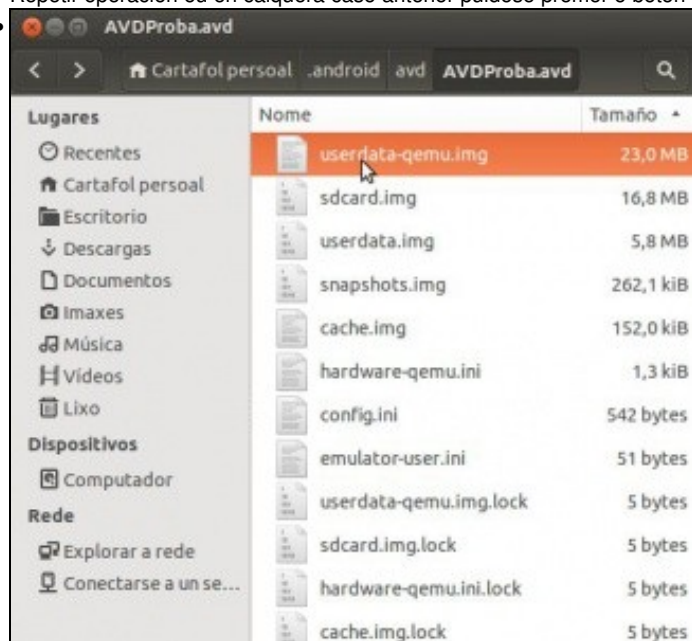


Repetir operación.

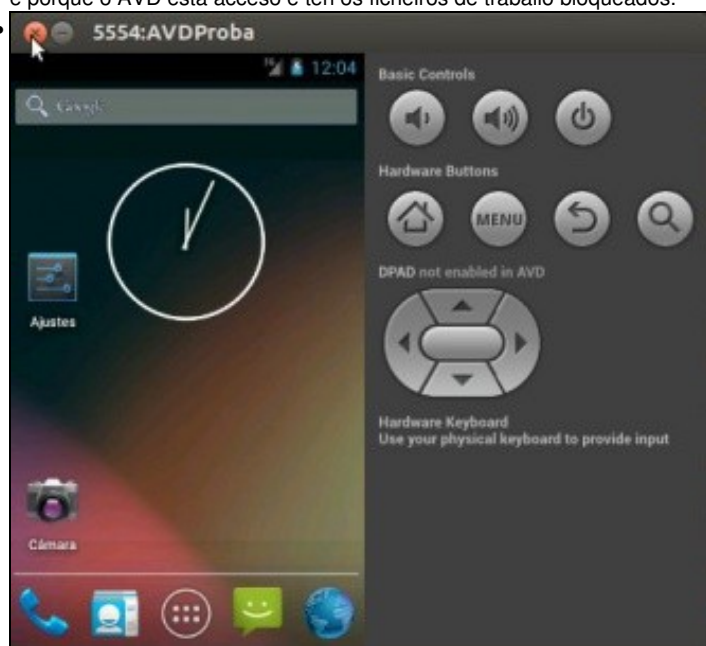




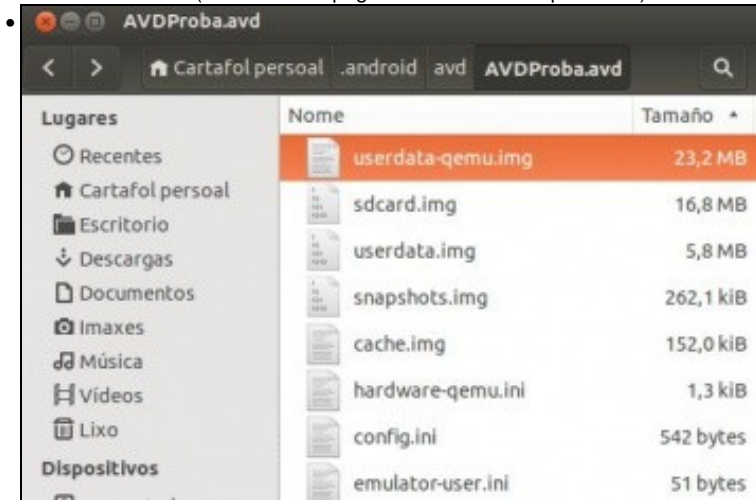
Repetir operación ou en calquera caso anterior púidose premer o botón Hardware Home.



Na carpeta asociada ao AVD pódense ver tódolos ficheiros que se crearon e os seus tamaños. Tamén se pode ver que se hai ficheiros .lock é porque o AVD está acceso e ten os ficheiros de traballo bloqueados.



Pechamos o AVD (Non fai falla apagalo no botón correspondente).



E agora vense os ficheiros creados perante o uso do AVD.

## Cambio de parámetros do AVD

- Agora vanse cambiar algúns parámetros do dispositivo e ver como afectan a este.
- O usuario pode cambiar todos aqueles parámetros que considere oportunos en orde a favorecer unha maior velocidade de funcionamento do AVD.

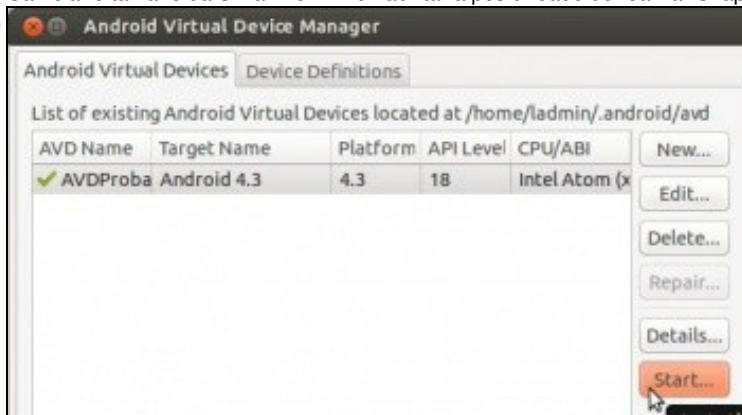
- Cambiar parámetros do AVD



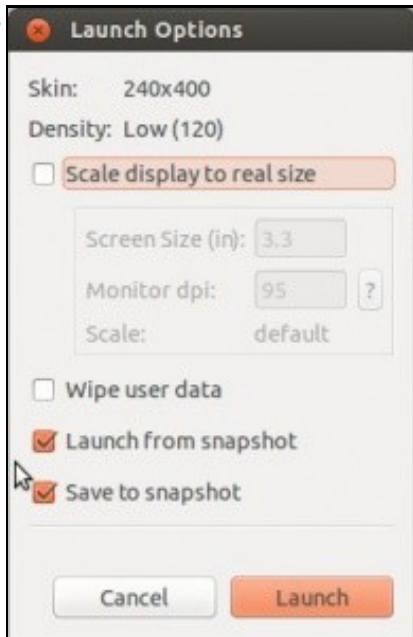
Editar o AVD.



Cambiar o tamanho da SD a 128 MiB e habilitar a possibilidade de realizar Snapshots, instantâneas.



Iniciar o AVD.



Marcar que se inicie o AVD dende un Snapshot e que cando se apague garde o seu estado.



Abrir o navegador web.



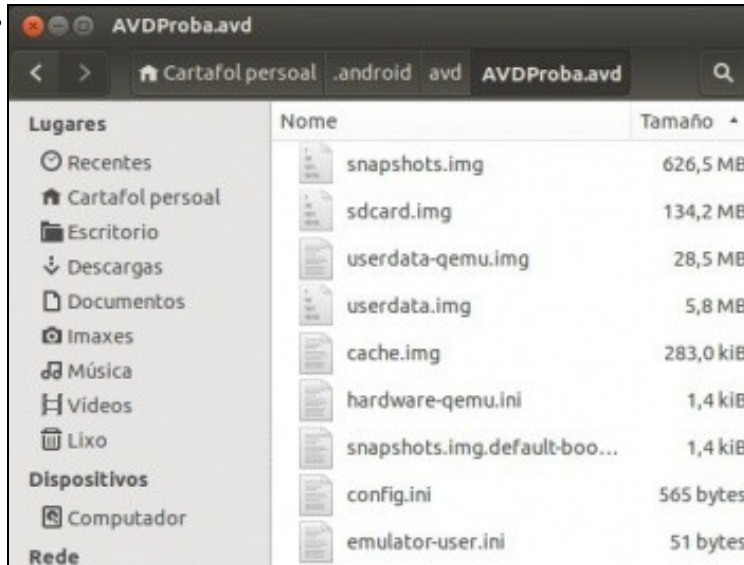
Facer dobre clic sobre el ...



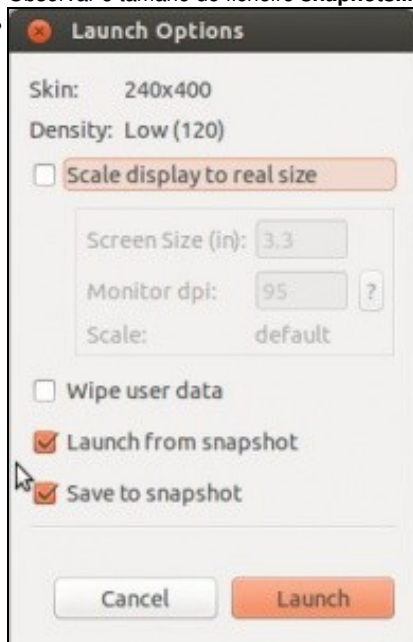
... amplíase a imaxe ... Volver a facer dobre clic para volver ao tamaño normal.



Pegar o AVD.



Observar o tamaño do ficheiro **snaphots.img**



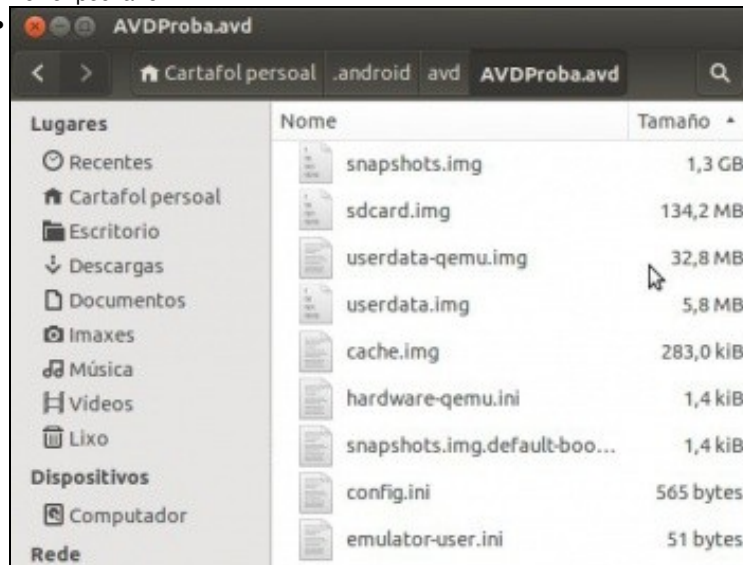
Volver lanzar o AVD.



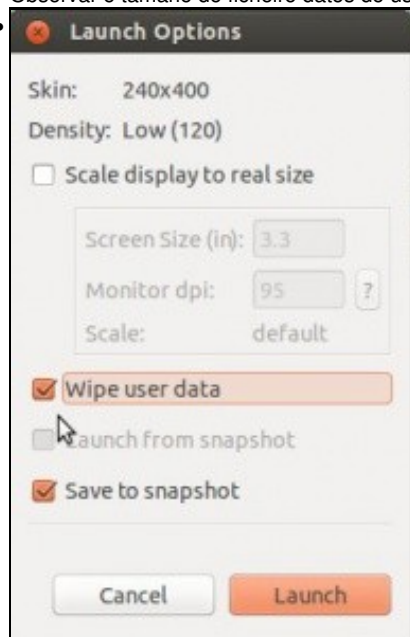
Carga no último lugar no que se estaba traballando no AVD e o inicio é inmediato. Esta vez non hai tempo para o café!!!



Volver pechar o AVD.



Observar o tamaño do ficheiro datos do usuario **userdata-qemu.img**.



• Volver lanzar, pero esta vez habilitar **Wiper user data** para que borre os datos de usuario e deixe o AVD como novo de fábrica.

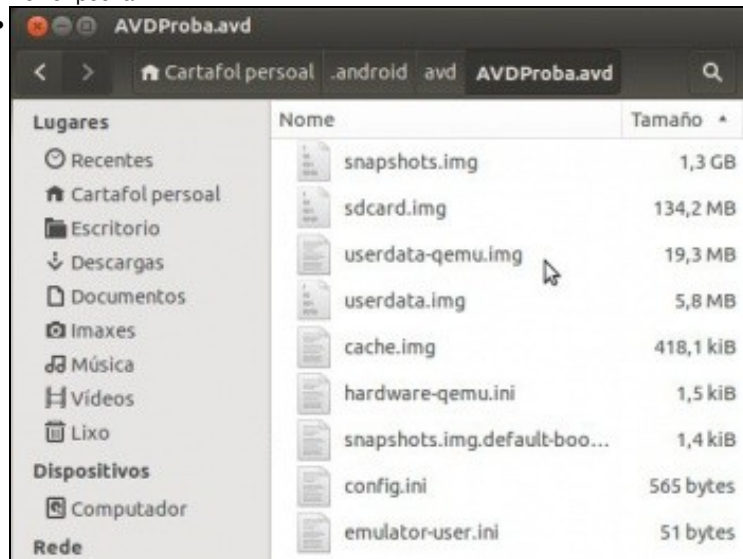


• Observar que **Settings** non está na pantalla de inicio e que o dispositivo está de novo en inglés.





Volver pechar

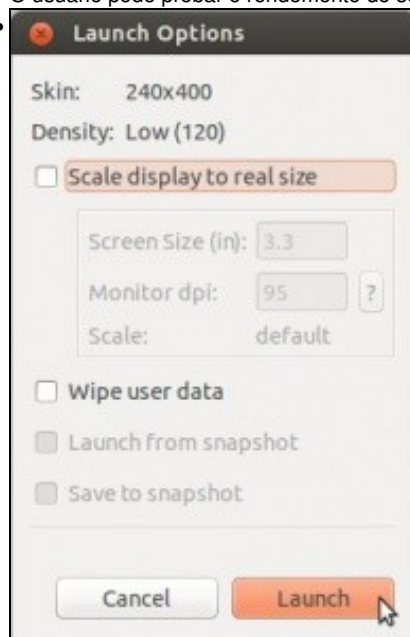


Observar como diminuíu o tamaño do ficheiro anterior.

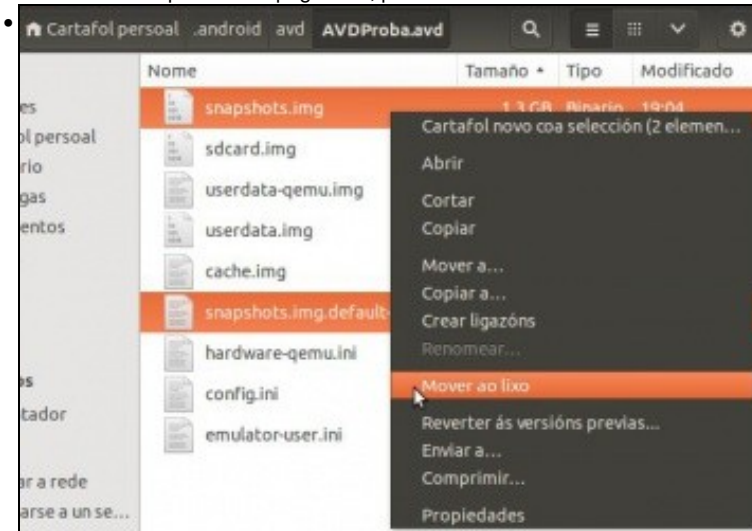


Volver editar o dispositivo e deshabilitar o Snapshot.

O usuario pode probar o rendemento do seu dispositivo facendo uso de GPU, se o desexa.



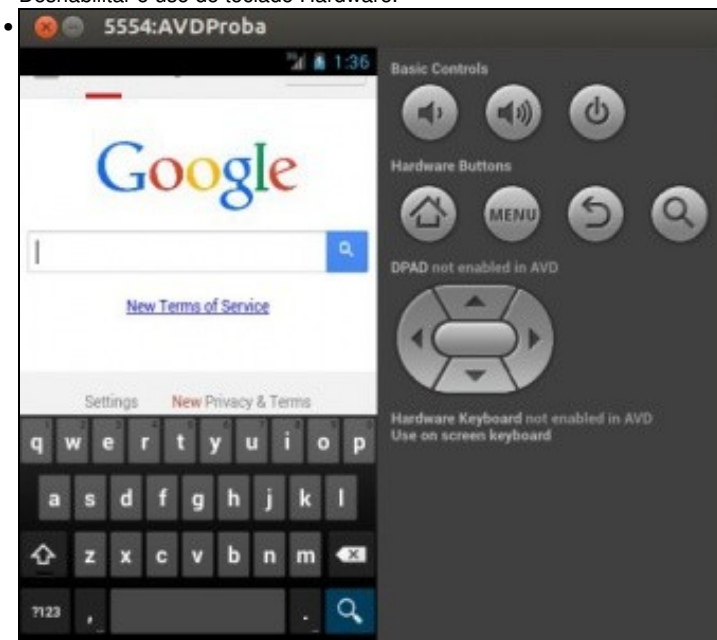
Lanzamos o dispositivo e apagámolo, pero ...



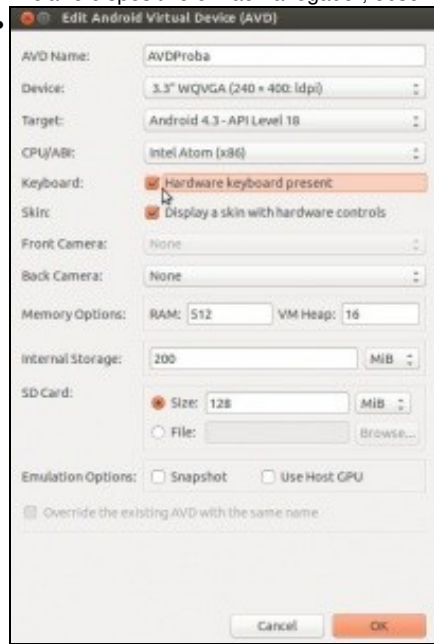
Non se eliminan os ficheiros de Snapshot, hai que borralos á man se se desexa, pois xa ocupan máis de 1 GiB.



Deshabilitar o uso de teclado Hardware.



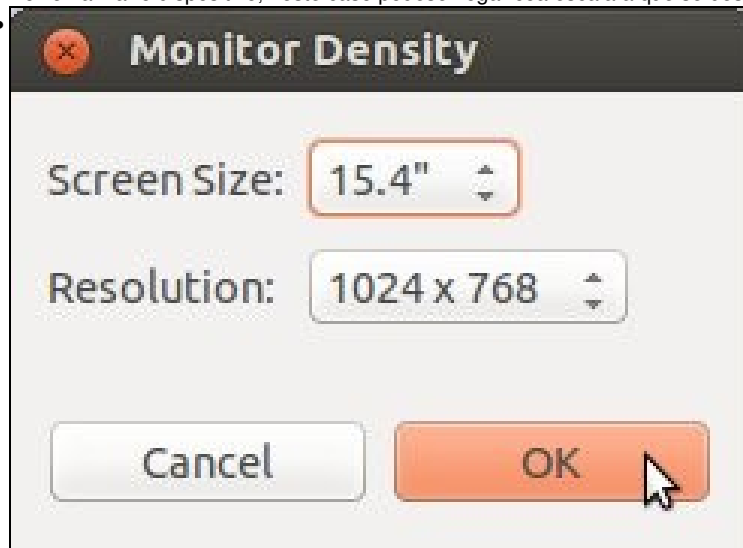
Iniciar o dispositivo e ir ao navegador, observar como para teclear hai que facer uso do teclado en pantalla do AVD.



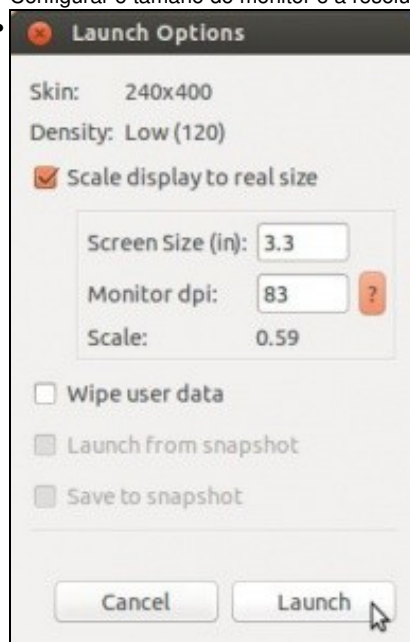
Volver habilitar o teclado Hardware.



Volver lanzar o dispositivo, neste caso pódese xogar coa escala á que se desexa representar o AVD no ordenador real.



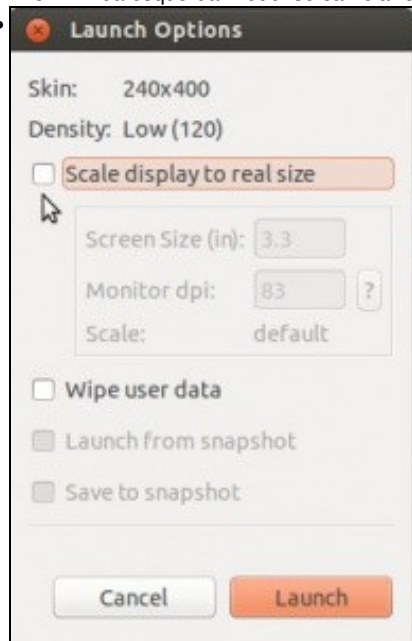
Configurar o tamaño do monitor e a resolución.



Iniciar o dispositivo, que a esta escala vese como ...



... o AVD da esquerda. Pódense cambiar os valores de Screen Size para obter o tamaño desexado.



Volver deixar por defecto.

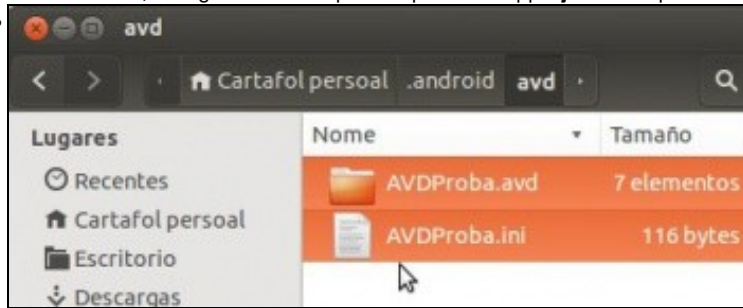
## Intercambio de AVDs

- Como se mencionou pódese crear un AVD nun ordenador e copialo a outro. Nesta ocasión vaise copiar o AVDProba creado e Ubuntu a un equipo Windows.

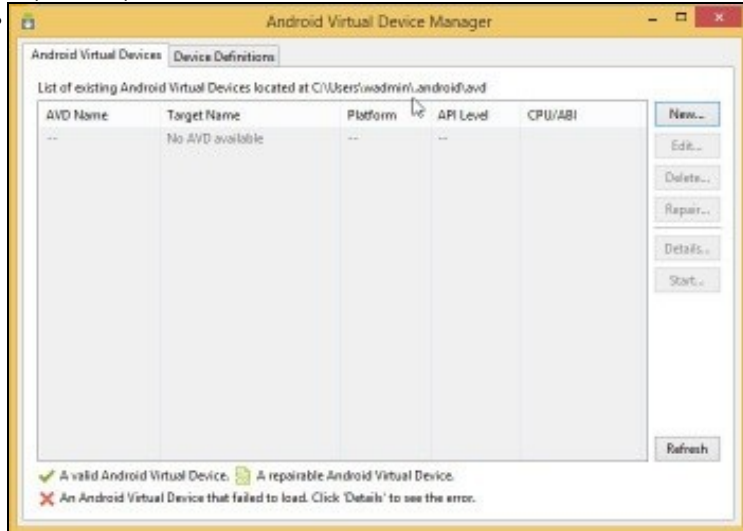
- Intercambio de AVDs



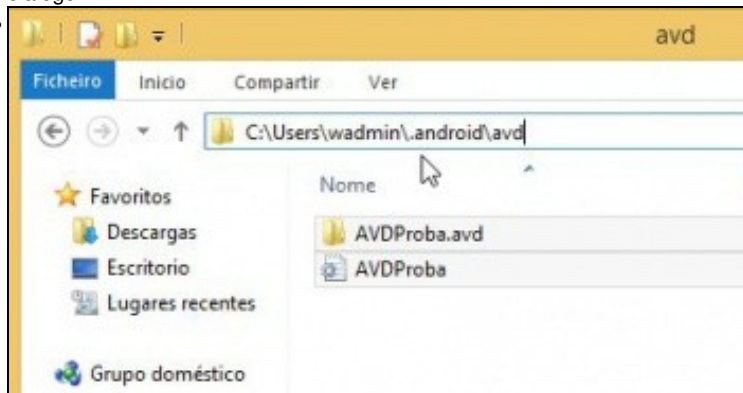
Iniciouse AVD, configurouse en Español e púxose a app **Ajustes** na pantalla de inicio.



Copiar a carpeta e ficheiro asociados ao AVD.



En Windows iniciamos o xestor de AVDs, por se nos esquecermos onde se gardaban estes. Podemos velo na parte superior do cadro de diálogo.



Pegar os ficheiros copiados de Ubuntu.



No Xestor de AVDs premer en **Refresh**.



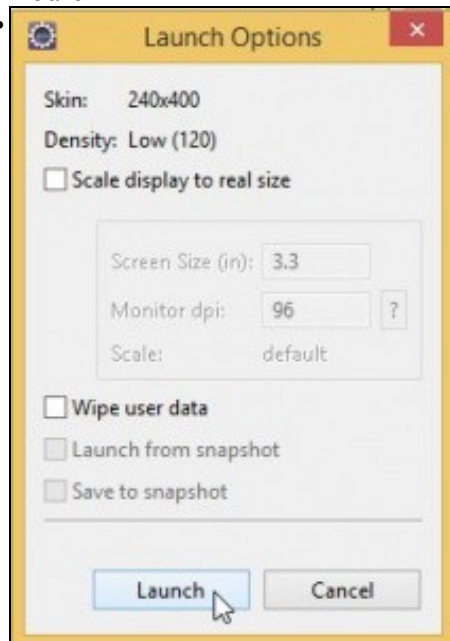
"E aí o está!".



Pódese editar para ver que mantén os parámetros de cando estaba en Ubuntu.



Iniciar o AVD.

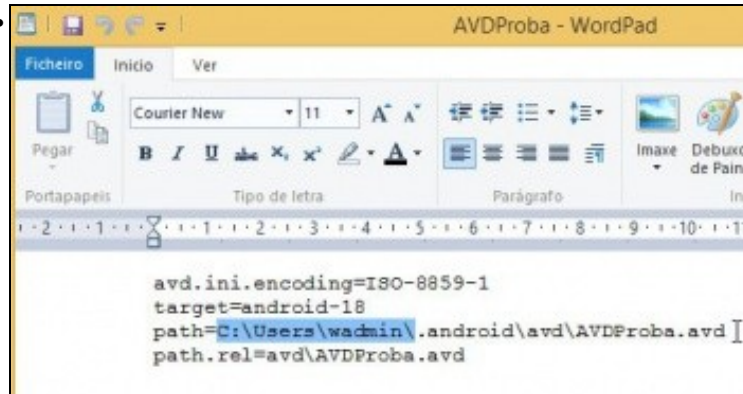


Lanzalo ...



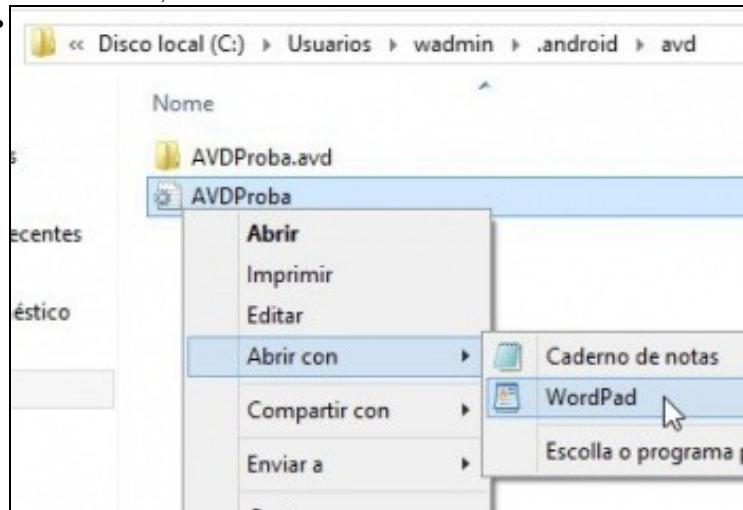


E tal cal se deixou en Ubuntu está no novo ordenador en Windows.

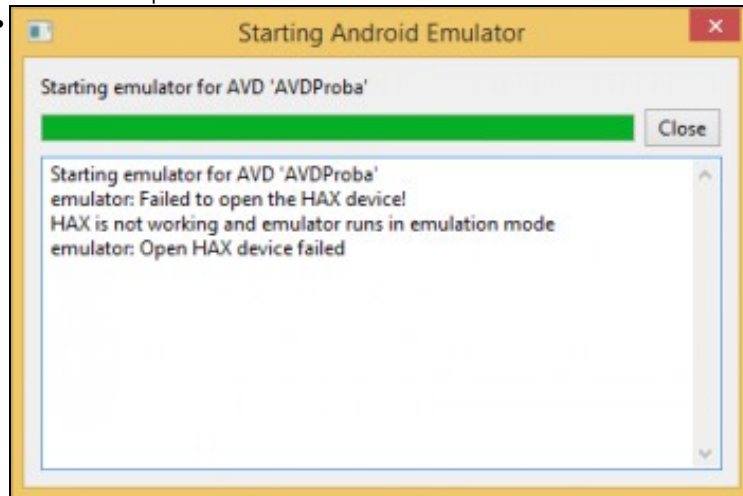


```
avd.ini.encoding=ISO-8859-1
target=android-18
path=C:\Users\wadmin\.android\avd\AVDProba.avd
path.rel=avd\AVDProba.avd
```

Se non se inicia, revisar o ficheiro AVDProba.ini



Abrir con Wordpad.



Pero mentres se lanza o dispositivo, obsérvase que o HAX (Hardware Accelerated Execution) non está funcionando. Ben!!! é que non está nin instalado ...

## Aceleración HW

- Só está dispoñible se a CPU proporciona as extensións de virtualización, que deben estar habilitadas na Bios.
- Esta funcionalidade foi probada en:
  - ◆ Windows e Mac OS X (Mavericks) con resultados satisfactorios.
  - ◆ Linux, non se apreciou melloría.

• Pódese atopar máis información en: <http://developer.android.com/tools/devices/emulator.html#accel-vm>

## Uso en Linux procesador Intel

- Para poder usar un dispositivo AVD con CPU Intel non pode estar en uso outra tecnoloxía de virtualización como o VirtualBox, senón obtense o seguinte erro ao iniciar o AVD.



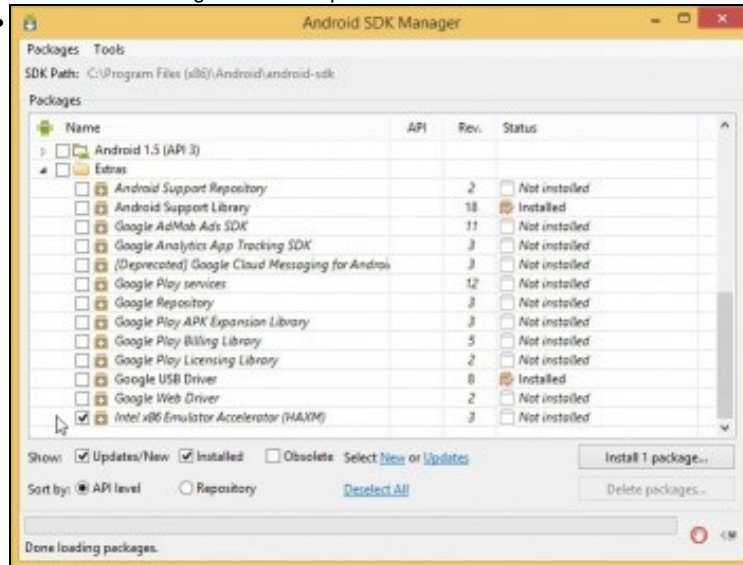
## Aceleración por HW en Windows

- Hai que instalar un compoñente do SDK Manager.

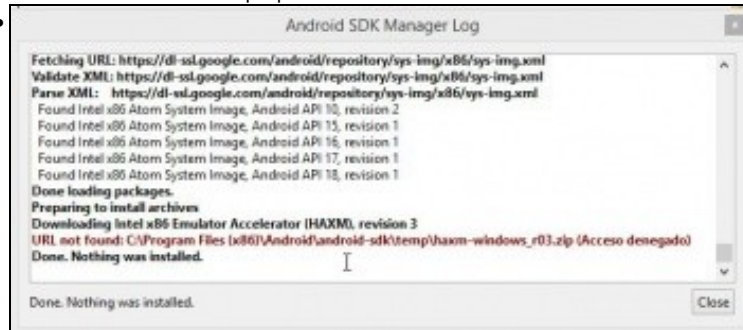
- Intercambio de AVDs



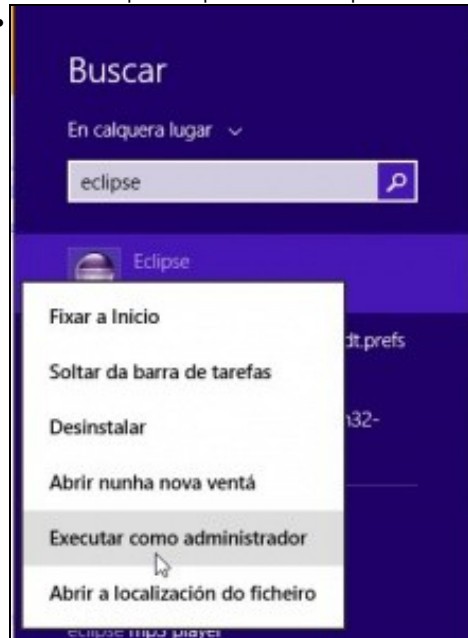
Iniciar o SDK Manager dende calquera dos métodos vistos.



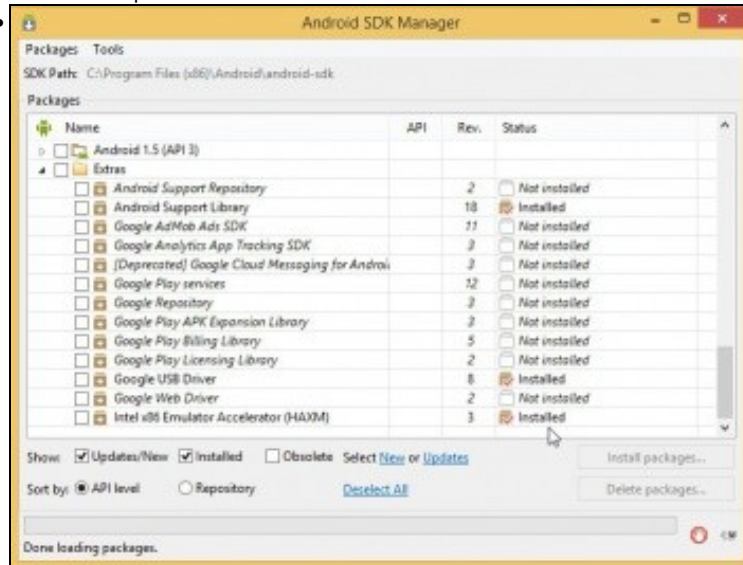
Seleccionar en **Extras** o paquete **Intel Hardware Accelerated Execution Manager**. Instalar ...



... Ostras!!! que hai que lanzar o eclipse ou Android SDK Manager como administrador.



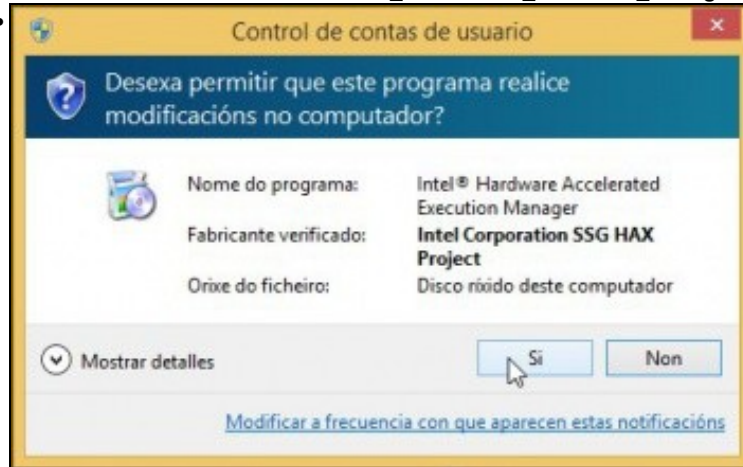
Executar Eclipse como administrador.



Instalar o paquete anterior.



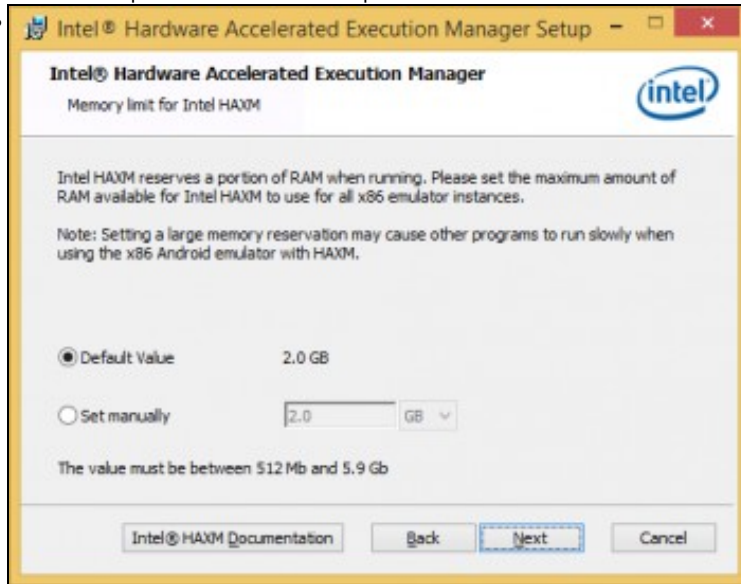
Ir á ruta do SDK /extras/intel/Hardware\_Accelerated\_Execution\_Manager e executar a aplicación IntelHAXM.exe



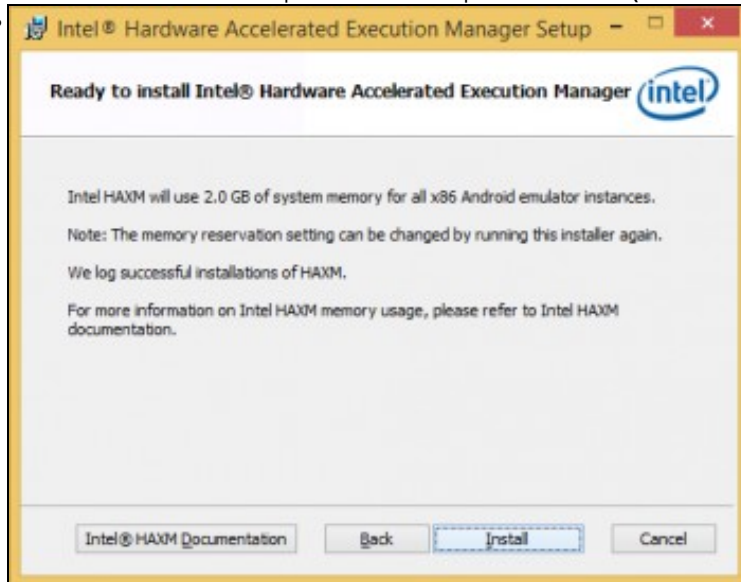
Aceptar a advertencia de seguridade.



Esta imaxe aparece cando non se dispón das extensións de virtualización ou estas non están habilitadas na BIOS.



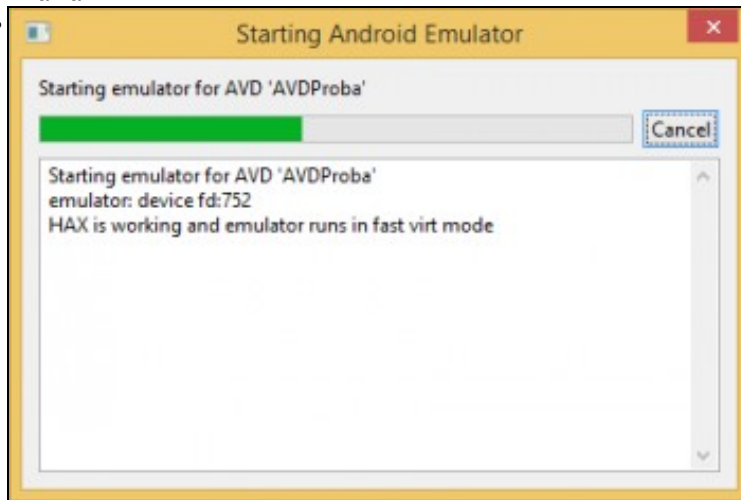
Indicar a cantidade de RAM que se vai reservar para Intel HAXM (Hardware Accelerated Execution Manager).



Instalar ... Aceptar as pantallas que poidan aparecer



Finalizar



Lanzar o AVD. Agora pódese comprobar que se está usando HAX. Agora o inicio do AVD debera estar como moito nos 20 segs.

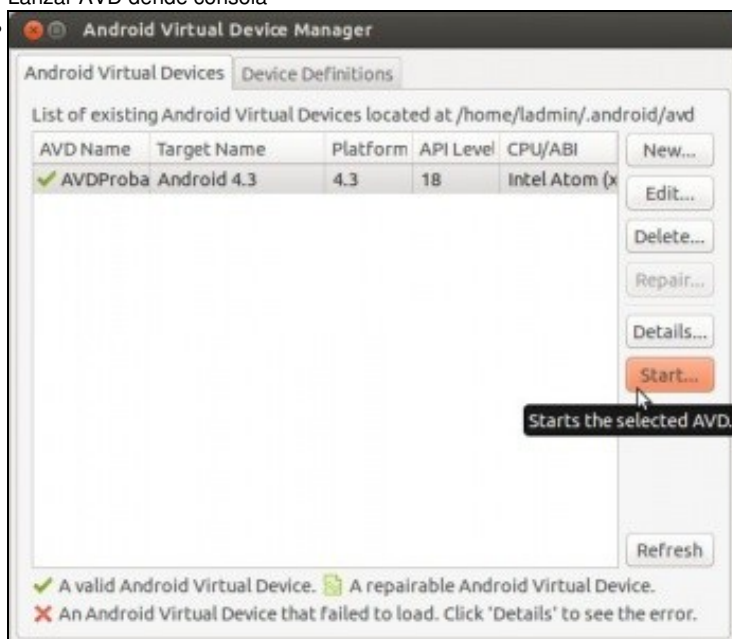


Tamén se pode comprobar que HAX está habilitado co seguinte comando: `sc query intelhax`

## Lanzar un dispositivo dende a consola

- Para iso faise uso do comando `tools/emulator`
- Isto vale para calquera dos tres SOs.
- Máis información en: <http://developer.android.com/tools/help/emulator.html>

- Lanzar AVD dende consola



Un AVD calquera ...

```
admin@ubase: ~
admin@ubase:~$ /opt/android-sdk-linux/tools/emulator -help
Android Emulator usage: emulator [options] [-qemu args]
options:
  -sysdir <dir>          search for system disk images in <dir>
  -system <file>         read initial system image from <file>
  -datadir <dir>        write user data into <dir>
  -kernel <file>        use specific emulated kernel
  -randisk <file>       randisk image (default <system>/randisk.img
  -image <file>         obsolete, use -system <file> instead
  -initdata <file>     same as '-init-data <file>'
  -data <file>         data image (default <datadir>/userdata-qemu.img
  -partition-size <size> system/data partition size in MBs
  -cache <file>        cache partition image (default is temporary file)
  -cache-size <size>  cache partition size in MBs
  -no-cache            disable the cache partition
  -nocache            same as -no-cache
  -sdcard <file>      SD card image (default <system>/sdcard.img
  -snapstorage <file> file that contains all state snapshots (default <dat
  <dir>/snapshots.img)
  -no-snapstorage     do not mount a snapshot storage file (this disables
  all snapshot functionality)
  -snapshot <name>   name of snapshot within storage file for auto-start
  and auto-save (default 'default-boot')
  -no-snapshot       perform a full boot and do not do not auto-save, but
  qemu vncload and vncsave operate on snapshot
  -no-snapshot-save  do not auto-save to snapshot on exit: abandon change
  d state
  -no-snapshot-load  do not auto-start from snapshot: perform a full boot
  -snapshot-list     show a list of available snapshots
  -no-snapshot-update-time do not do try to correct snapshot time on restore
  -wipe-data         reset the user data image (copy it from initdata)
  -avd <name>       use a specific android virtual device
```

Pode iniciarse dende a consola co comando **tools/emulator -avd**.

```
admin@ubase: ~
admin@ubase:~$ /opt/android-sdk-linux/tools/emulator -avd AVDProba &
```

Por exemplo **.../tools/emulator -avd AVDProba**. En Linux / OS X símbolo **&** é para que o lance en segundo plano e devolva o control á consola.

- Quen o desexe pode realizar fotos coa cámara do AVD, tanto conectada á cámara real do ordenador, se é o caso, como conectada a nada.