

Capítulo 1

Primeros pasos con Android

Resumen: En este capítulo daremos nuestros primeros pasos con Android, instalando el software necesario para desarrollar aplicaciones para él.

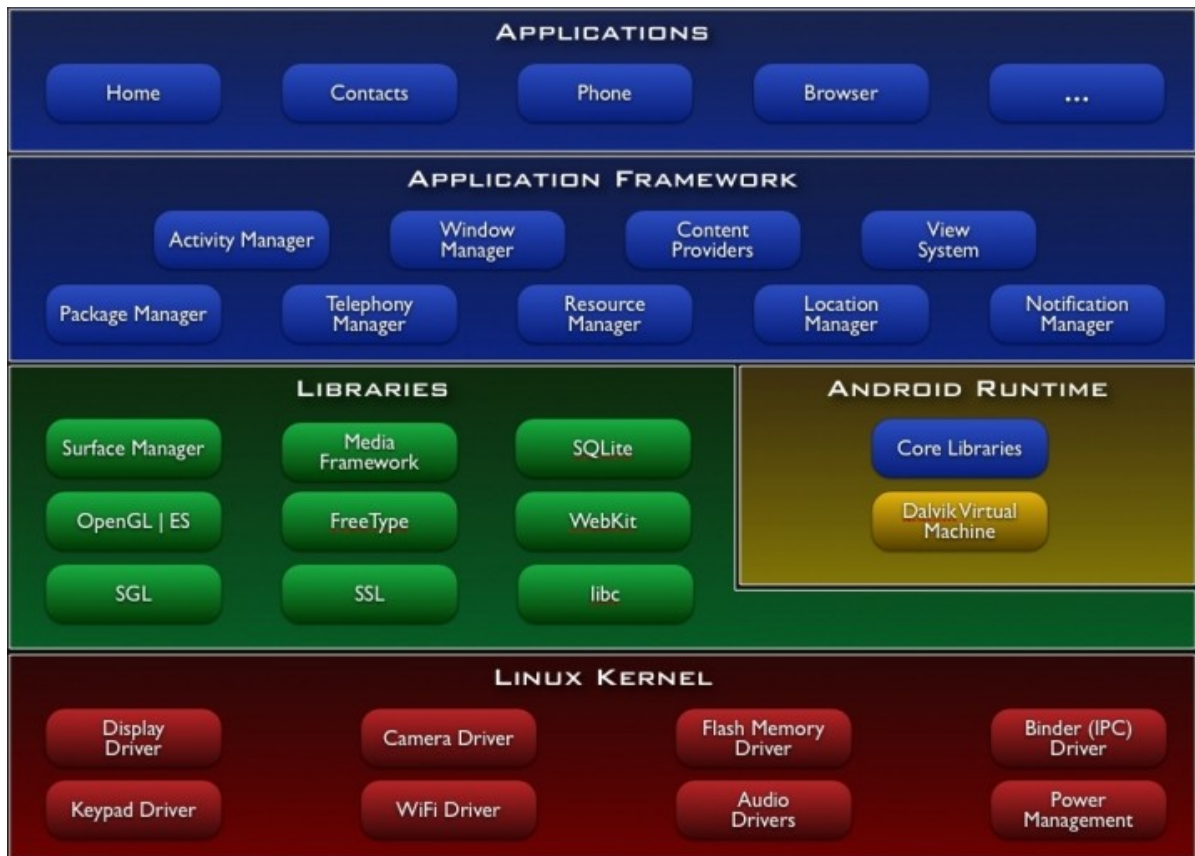
Android es una pila software para dispositivos móviles. Incluye el sistema operativo, un middleware, y una serie de aplicaciones básicas. Su origen se remonta a Octubre de 2003, año de la creación de Android Inc., que nació para desarrollar "dispositivos móviles más inteligentes, que fueran conscientes de la localización y preferencias de sus usuarios".

En Agosto de 2005 es comprada por Google, dejando ver su intención de entrar en el mercado de la telefonía móvil. En noviembre de 2007 se anuncia la creación de la Open Handset Alliance, un consorcio de empresas (Google, HTC, Intel, Samsung, Motorola...) que persigue el desarrollo de estándares abiertos para dispositivos móviles. El mismo día se anuncia el primer producto: la plataforma Android para dispositivos móviles basada en Linux. Aunque normalmente se dice que Android es de Google, formalmente Android es desarrollado por el Android Open Source Project que, eso sí, es liderado por Google.

Las características más destacadas son:

- Código abierto
- Gran cantidad de servicios: acceso al hardware (acelerómetro, GPS, ...), base de datos relacional, ...
- Aplicaciones hechas de componentes
- Capacidades multimedia
- Seguridad (entre aplicaciones y para el usuario)
- Gestión del ciclo de vida automático
- Ejecución en hardware variado
- Programación en Java (o en C/C++, aunque menos recomendable)
- Google mantiene el "Android Market" predefinido

La arquitectura de Android se resume en la siguiente figura:



Práctica 1.1: Instalación de las SDK de Android

Para poder desarrollar aplicaciones para Android necesitamos herramientas específicas que nos permitan compilar el código, empaquetar la aplicación, y visualizarla en un "teléfono software" (un emulador), o enviarla a un dispositivo físico.

Todas esas aplicaciones se proporcionan en las SDK (software development kit) de Android, disponibles en <http://developer.android.com/sdk/>.

La instalación es sencilla.

Se creará el directorio android-sdk-linux en la ruta actual, y se descomprimirá ahí todo el contenido.

En realidad lo que conseguiremos con esta instalación son únicamente las herramientas básicas de las SDK. Pero con ellas aún no podremos desarrollar nada. Necesitamos descargarnos lo que se denominan "Android platforms", que contiene la información específica de la versión concreta de Android para la que queremos desarrollar. Por tanto, lo que acabamos de instalar es prácticamente un gestor de descargas, y un emulador de terminales Android "vacío" al que le falta el propio Android que debe ejecutar.

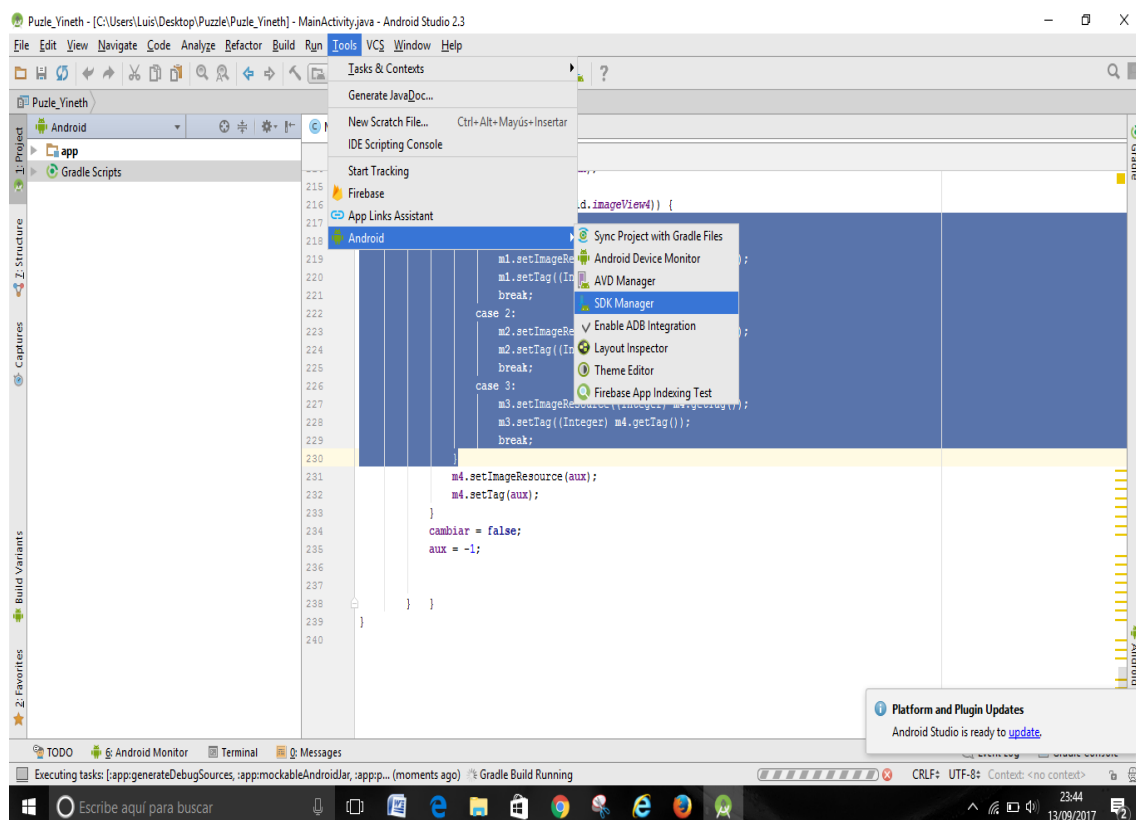
Si miramos el contenido de las carpetas.

- **add-ons:** de momento es un directorio vacío. En él se guardarán las ampliaciones a versiones concretas de Android (proporcionadas por terceros) que hayamos instalado.
- **platforms:** también vacío, contendrá el software de las diferentes versiones de Android que instalemos. El emulador de teléfonos Android ejecutará ese software.
- **tools:** contiene las herramientas básicas para la gestión de las SDK. El más importante es android.

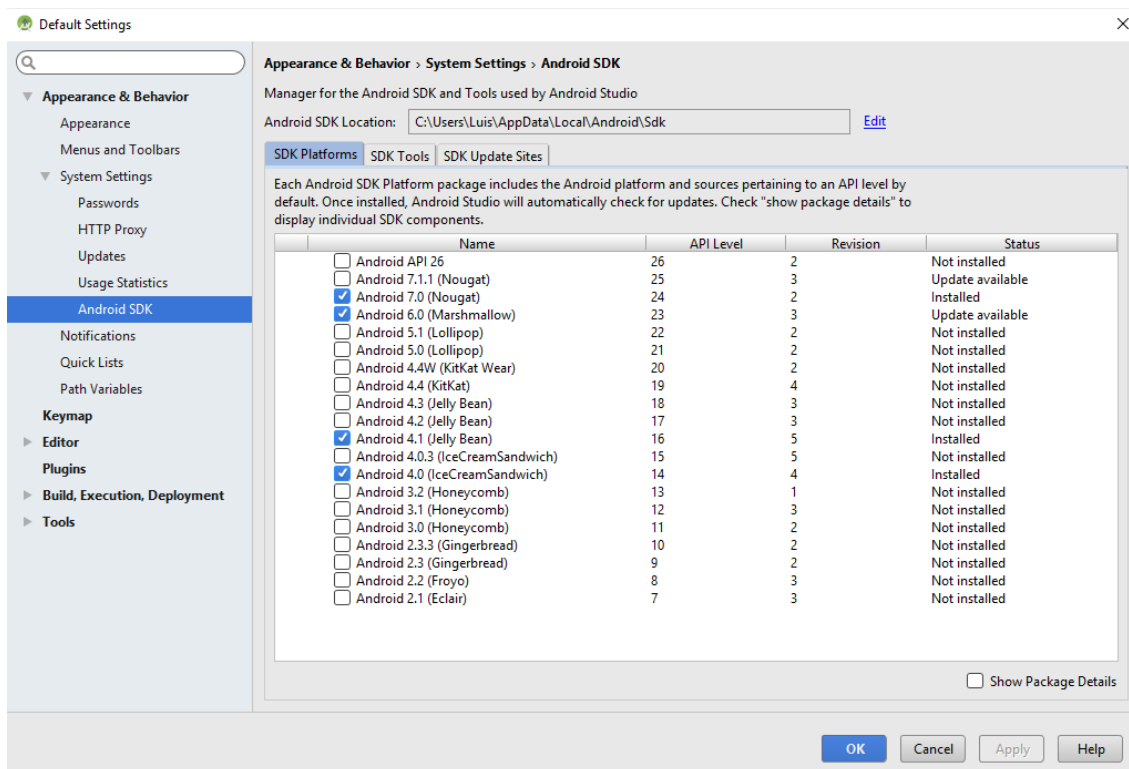
Práctica 1.2: Instalación de "plataformas Android"

- Después de instalar Android studio hay que comprobar si tenemos instalado SDK.

Tools/Android/SDK Manager



Comprobamos la ruta del SDK.



Práctica 1.3: Creación de Android Virtual Devices

Antes de comenzar a desarrollar nada, crearemos "emuladores de teléfonos" sobre los que ejecutar nuestros programas. También podremos utilizar dispositivos físicos, pero es más cómodo utilizar emuladores software, al menos en las primeras etapas del desarrollo.

En las SDK de Android los "teléfonos virtuales" se conocen como AVD (Android Virtual Device). Cada AVD está compuesto de:

- "Dispositivo físico" que emula: especifica el "perfil hardware" (sensores, tamaño de la pantalla, etcétera). No incluye el tipo de CPU ni la información del software.
- Imagen del sistema: qué software ejecuta (versión de Android).
- Procesador (ARM, Intel, MIPS) y ajustes de hardware adicionales.
- Almacenamiento dedicado en el disco duro anfitrión para guardar las aplicaciones instaladas en el terminal, la tarjeta de memoria (si tiene), etcétera.
- Opciones adicionales: si se puede hacer snapshot, o si se usa aceleración gráfica.

En esencia, es similar a una máquina virtual¹, pero ejecutándose sobre un hardware emulado particular (el de los teléfonos), y sobre él el software de Android en la versión que elijamos.

A nivel hardware, se emula:

- CPU y su unidad de gestión de memoria (MMU), ya sea ARMv5, Atom o MIPS.
- Pantalla LCD de 16 bits.
- Uno o más teclados.

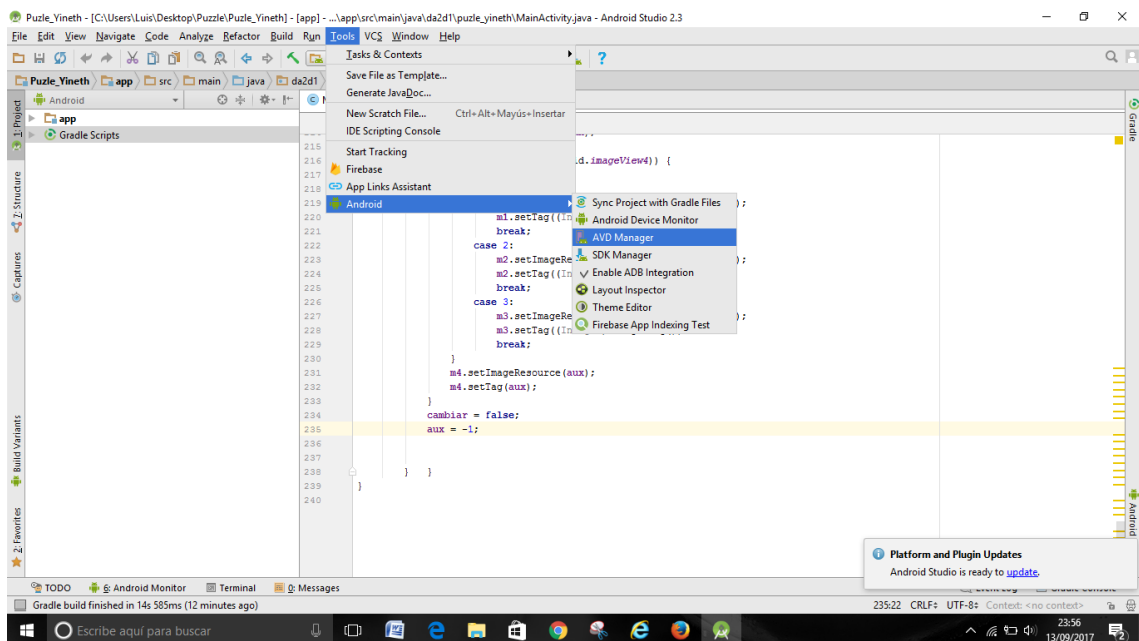
- Chip de sonido con entrada y salida.
- Particiones de la memoria flash (emulado con ficheros en el disco duro anfitrión).
- Módem GSM con su tarjeta SIM (falsa).
- Cámara
- Sensores

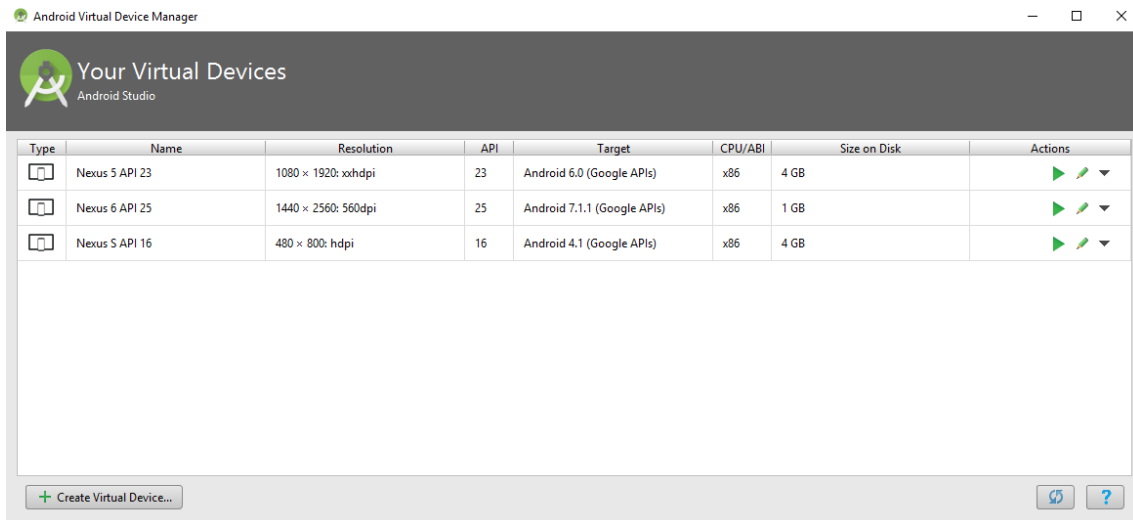
Aunque el emulador resulta increíblemente útil para el desarrollo (entre otras cosas, para poder probar la aplicación sobre diferentes plataformas Android sin necesidad de tener un dispositivo físico de cada una), tiene algunas limitaciones. Entre otras cosas, no es posible:

- Hacer o recibir llamadas reales, aunque se puede simular que se hacen y se reciben.
- Realizar conexiones USB.
- Detección de la conexión y desconexión de auriculares.
- Determinar el nivel de carga de la batería o si se está cargando.
- Detectar la inserción y retirada de tarjetas SD
- Comunicación por Bluetooth

Vamos a crear nuestro primer teléfono. Para eso, lanza el gestor de AVDs con:

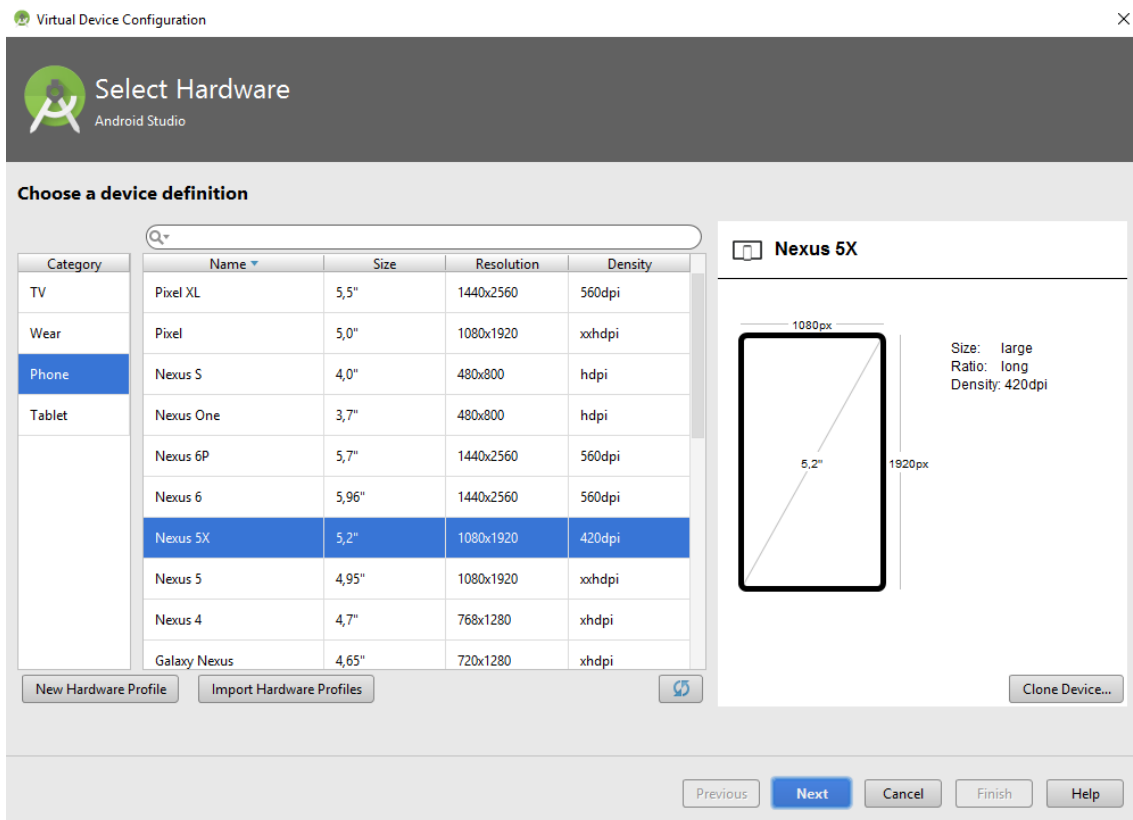
Tools/Android/AVD Manager



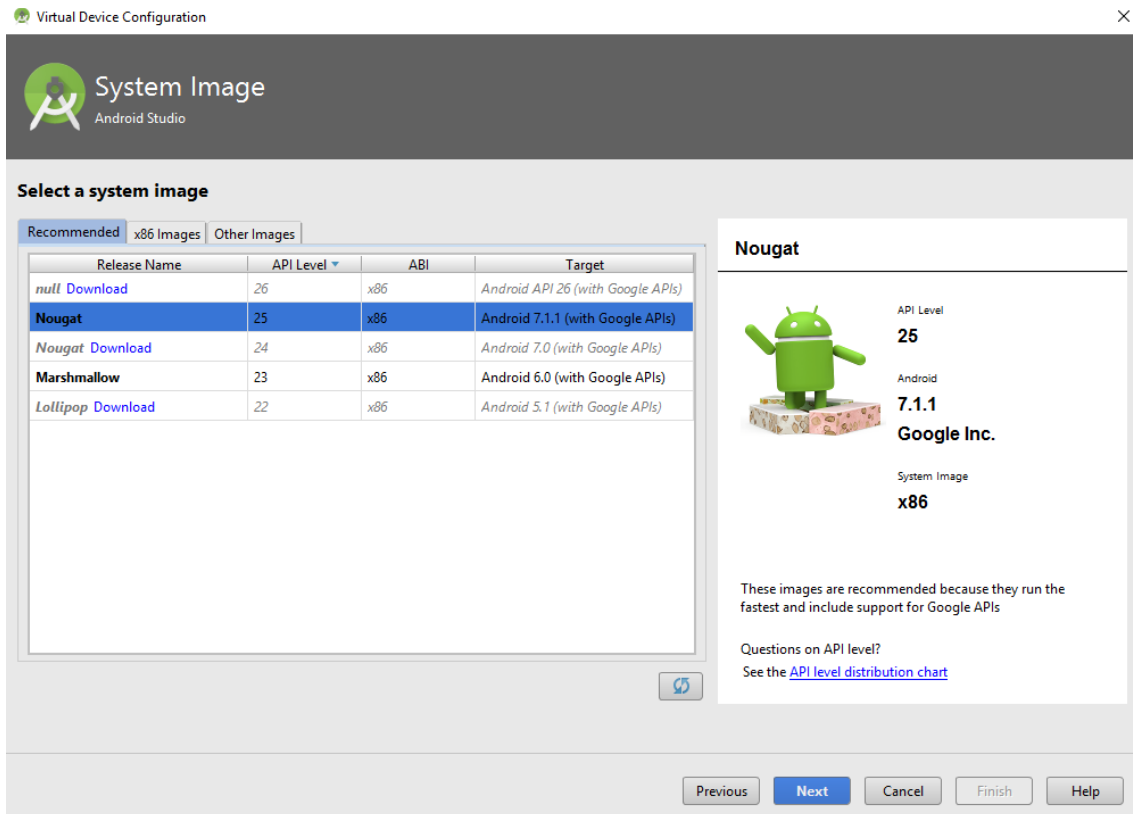


Pulsamos el botón

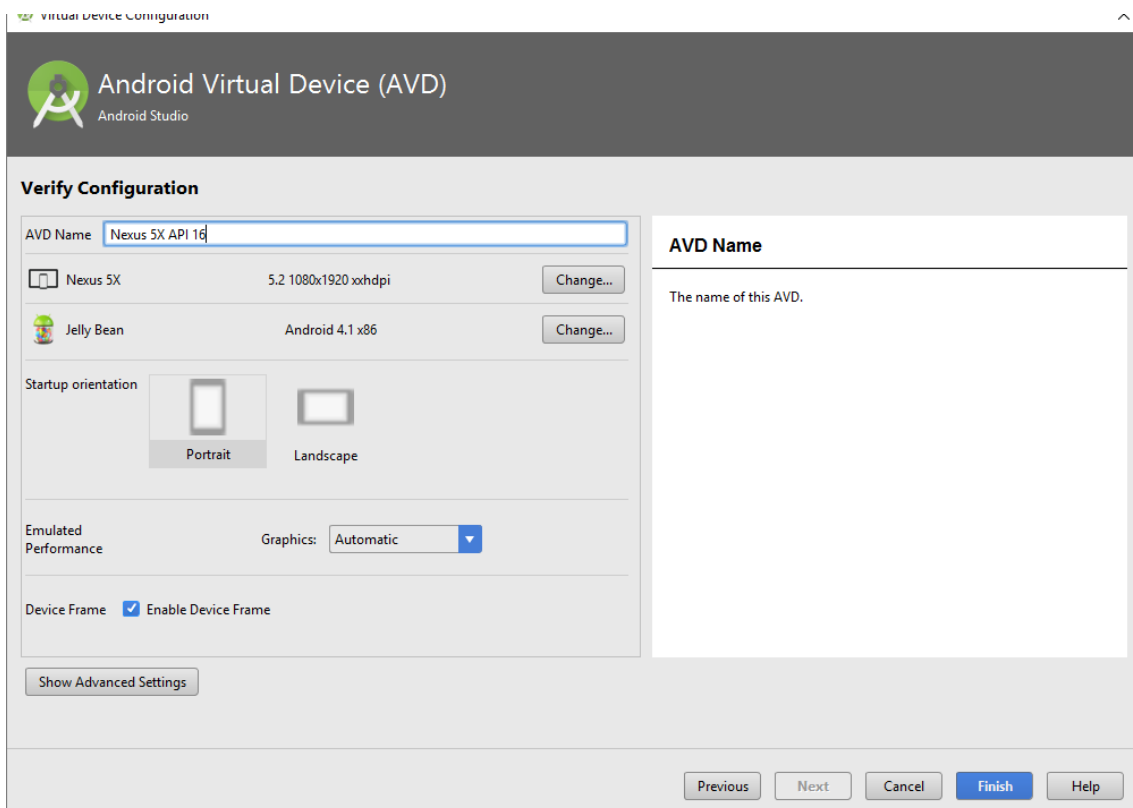
+ Create Virtual Device...



Seleccionamos el tipo de dispositivo físico y pulsamos Next.



Para poder elegir un sistema de imagen previamente hemos tenido que instalar el nivel de API, tarjeta, etc. en el SDK. Una vez seleccionado pulsamos Next.



Ponemos un nombre a la AVD y finalizamos.

Práctica 1.4: Explorando el AVD

Si no has tenido acceso previo a un dispositivo con Android, familiarízate con su uso. Lanza algunas aplicaciones y pruébalas. Fíjate que sólo puede haber una aplicación en primer plano, y que no hay botones para maximizar, minimizar o cerrar. A continuación se describen algunas de las teclas que puedes utilizar en el emulador:

- Escape: "volver" (atrás)
- Inicio: "home". Vuelve "al escritorio"
- Ctrl-F11/Ctrl-F12 o teclas 7 y 9 del teclado numérico (con el num-lock desactivado) rotan el dispositivo
- Ctrl-F5/Ctrl-F6 o teclas "+" y "-" para cambiar el volumen
- F8: des/habilitar GSM (llamadas)
- Alt-Enter: pantalla completa
- F3: "botón" de llamar

Práctica 1.5: Interacción entre dos AVDs

Si no los tienes ya, lanza ambos AVDs. Fíjate que en cada uno de ellos aparece un número como título de la ventana, junto al nombre. Por ejemplo nombre:5554, nombre2:5556. Podemos considerar ese número como el "identificador temporal" (en realidad es un número de puerto).

- Fíjate en el número asociado al AVD de un emulador que tengas lanzado.
- Ve al AVD ...5554: y pulsa F3 para entrar en las llamadas. También puedes utilizar el botón de llamada del lado derecho de la ventana, o simular el uso del teléfono y pulsar sobre el icono.
- Como número de teléfono, escribe el "identificador" del AVD del otro (por ejemplo, 5556) y llama.
- Ve al AVD :5554 y observa la llamada entrante (si el sistema tiene tarjeta de sonido, habrás oído la melodía de la llamada), junto con un número de teléfono, que termina con el identificador del AVD llamante. Descuelga (pero no esperes recibir sonido).
- Vuelve a 5556 y observa que se ha detectado que se ha contestado la llamada, y aparece el temporizador.
- "Cuelga" en cualquiera de los dos teléfonos, y observa que se detecta el fin de la "llamada" en el otro.

Por último, podemos hacer uso de la agenda. Añade en cualquiera de los dos teléfonos el contacto del identificador del otro, y repite la llamada.

Ten en cuenta que el identificador es temporal, pudiendo cambiar en ejecuciones posteriores, por lo que meterlo en la agenda no es especialmente útil. En cualquier caso, estos cambios desaparecerán si desmarcaste Save to snapshot al lanzar los AVD.

Práctica 1.6: La consola de los AVD

El identificador que veíamos antes es en realidad un puerto TCP (del anfitrión) donde se queda escuchando el AVD para recibir comandos. El primer AVD que se lanza se quedará escuchando en el puerto 5554, y se van incrementando de dos en dos. Como hemos visto, se puede conocer el puerto en el que está un AVD a la escucha mirando la barra de título de su ventana. El puerto se habilita únicamente para el localhost, por lo que no es posible la comunicación entre AVDs en diferentes máquinas. Puedes comprobarlo con `netstat -tln`. Verás que también están ocupados los puertos impares (5555 y 5557); un AVD ocupa dos puertos consecutivos, aunque por el momento sólo nos interesa el puerto par.

La comunicación con el AVD en su puerto es en texto, a través de un interfaz en línea de comandos relativamente intuitivo. Naturalmente, esto no podrá hacerse con los dispositivos físicos conectados por USB.

- Conéctate por telnet al AVD 5556

```
$ telnet localhost 5556
```

- Simula una llamada entrante de un número arbitrario:

```
gsm call 555666777
```

- Ve al AVD y descuelga.
- Vuelve al CLI y comprueba el estado de la llamada:

```
gsm list  
inbound from 555666777 : active
```

- Simula la recepción de una segunda llamada antes de haber terminado la primera:

```
gsm call 444333222
```

- Con `gsm list` observa que la segunda llamada está en estado incoming.
- Vuelve al AVD, y descuelga. La otra queda en espera.
- Vuelve a la consola y mira la lista de llamadas:

```
gsm list  
inbound from 555666777 : held  
inbound from 444333222 : active
```

- En el AVD, intercambia las llamadas, dejando en espera la segunda, y vuelve a listarlas.
- Desde el propio CLI podemos hacer muchas tareas. Por ejemplo, podemos terminar una llamada:

```
gsm cancel 444333222
```

- Con una llamada en marcha, simula la recepción de un mensaje:

`sms send 999888777 "Comunicas? Cuelga ya!!!"`

- Vuelve al AVD y observa la llegada del mensaje en la barra de notificación superior.
- Cierra la sesión de telnet con exit.