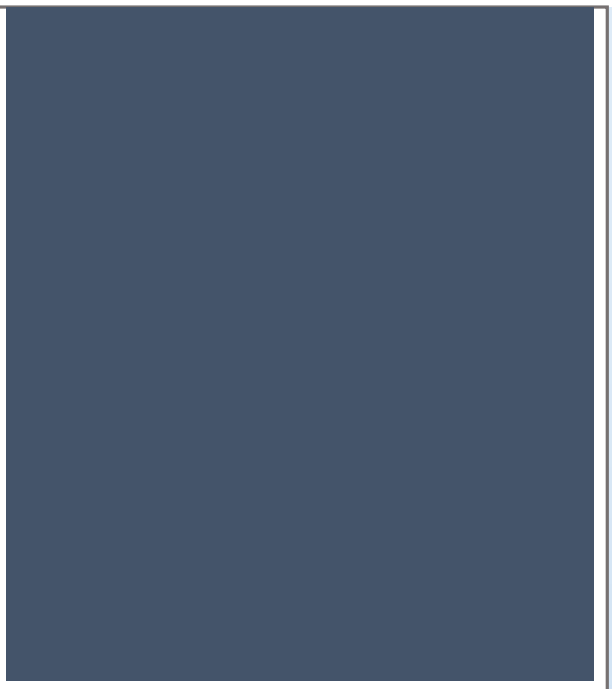


Programación multimedia y dispositivos móviles

Act1.2 El sistema operativo Android. Herramientas de desarrollo y gestión de dispositivos

Francisco José García Cutillas | 2FPGS_DAM

Índice

Ejercicio 1 3
Ejercicio 2 3
Ejercicio 3 10
Ejercicio 4 11

Ejercicio 1

¿Qué vamos a hacer?

- **Descargar el instalador de Android Studio desde la web.**
- **Realizar la instalación en nuestro equipo.**
- **Añadir un lanzador para Android Studio en el menú de aplicaciones.**
- **Utilizar el Device Manager para crear un dispositivo virtual en nuestro equipo, instalar el sistema operativo y realizar algunos ajustes.**
- **Comprobar que la aceleración por hardware está activada en nuestro equipo.**

Ejercicio 2

Probando nuestro Device Manager.

Si todo ha ido bien en la instalación de nuestro dispositivo virtual, es hora de ponerlo en marcha. Para ello, tal y como hemos visto en el caso práctico anterior, es posible lanzarlo tanto desde el Device Manager como desde la terminal.

Inicia tu dispositivo virtual y comprueba que funciona correctamente, así como la versión del sistema que tiene instalada. Personaliza también tu dispositivo configurando, por ejemplo, tu idioma preferido.

Device Manager

En este caso práctico vamos a ver cómo crear un dispositivo virtual sobre el que lanzar las aplicaciones que desarrollemos. Para ello, utilizaremos el componente Device Manager, que ya tendremos de la instalación de Android Studio.

Accediendo al Device Manager

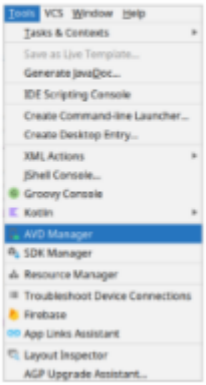
Una vez hemos instalado el IDE Android Studio, y cada vez que abramos el IDE sin tener ningún proyecto reciente, se presenta una ventana de bienvenida similar a la siguiente:

En ella se nos muestran tres opciones principales para crear un proyecto nuevo, abrirlo o importarlo directamente desde un sistema de control de versiones.

No obstante, desde la opción inferior *More Actions* podemos realizar algunas acciones más, tales como:

- Depurar aplicaciones.
- Importar proyectos creados con otros IDE.
- Importar códigos de ejemplo.
- Gestionar el SDK.
- Gestionar dispositivos con el Device Manager.

La opción que nos interesará para añadir un nuevo dispositivo virtual será el Device Manager. Si ya tuviésemos Android Studio iniciado con algún proyecto, también podemos acceder al Device Manager a través del menú de herramientas (*Tools*):



Creación de un dispositivo virtual

Cuando abrimos el Device Manager se presenta una ventana embebida dentro del propio Android Studio, con dos pestañas en la parte superior: *Virtual* y *Physical*. La primera de ellas nos servirá para crear dispositivos virtuales, mientras que la segunda nos permitirá configurar dispositivos físicos conectados. Vamos a ver cómo crear un nuevo dispositivo virtual.

Esta interfaz del Device Manager muestra una lista de los diferentes dispositivos configurados. Inicialmente, esta lista estará vacía.

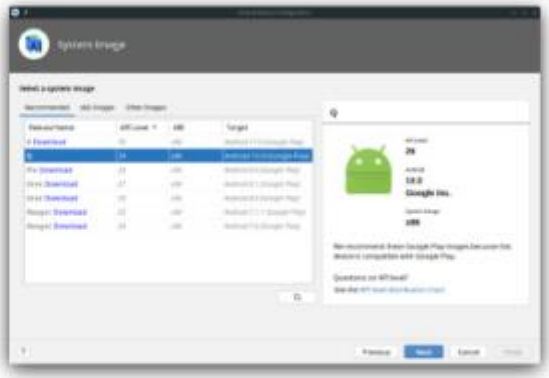
Paso 1. Selección de un nuevo dispositivo

Para añadir un nuevo dispositivo, hacemos clic en el botón *Create Virtual Device*, y con ello accederemos a la ventana de selección del dispositivo. Como vemos, es posible crear desde teléfonos y tabletas hasta televisores, pasando por dispositivos wereables o automóviles.

En la lista, seleccionamos un teléfono que se ajuste a las capacidades de vuestro equipo. Pensad que cuanto más potente sea el dispositivo que deseéis emular más carga le supondrá al sistema. Lo que sí es conveniente es que se trate de un dispositivo que tenga soporte para la Play Store. Cualquiera de los Pixel del propio Google puede ser una buena opción. En este caso, se ha escogido un Pixel 3, pero podríais utilizar el 4 si disponéis de un buen equipo.

Paso 2. Selección de la imagen del sistema

Una vez hemos escogido el dispositivo, podemos instalar en él diferentes versiones de Android. El sistema recomienda una versión concreta para el dispositivo, pero podemos elegir cualquiera.

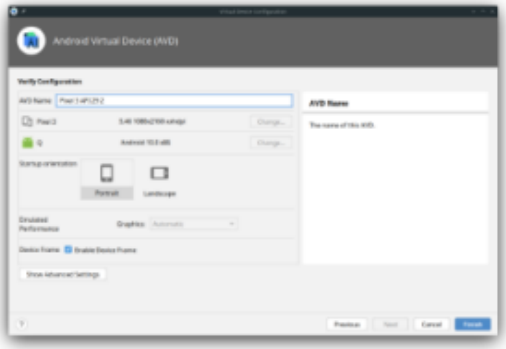


Recordad que las versiones más nuevas ofrecen más posibilidades, a través de API de mayor nivel. No obstante, hay que tener en cuenta los dispositivos que queremos que sean compatibles con nuestras aplicaciones. Para ayudarnos a ello, tenemos el enlace *API level distribution chart*, un gráfico donde se muestra información más o menos actualizada sobre el porcentaje de dispositivos que soportan una u otra versión. En nuestro caso, hemos escogido la versión 10 [Nivel API 29].

Cuando hayamos seleccionado una versión, en la ventana anterior deberemos hacer clic en *Download* para descargar la imagen completa del sistema. Este proceso puede tomar cierto tiempo.

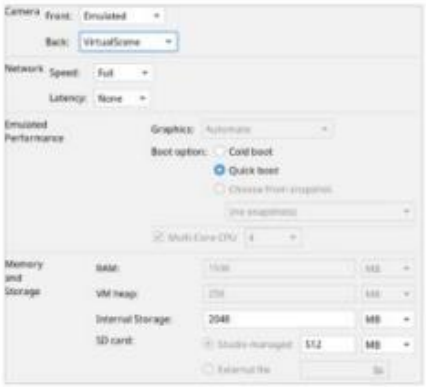
Paso 3. Últimos ajustes

Finalmente, verificamos los ajustes seleccionados, el nombre del dispositivo y cómo queremos que este se muestre, en vertical u horizontal, así como si deseamos que el marco del dispositivo se muestre o no.



En los ajustes avanzados [*Show Advanced Settings*] podemos configurar algunos más, como:

- Las cámaras, frontal y trasera, donde podemos utilizar imágenes y entornos emulados, o directamente enlazar a la cámara de nuestro equipo.
- Las capacidades de la red.
- El tipo de arranque, en frío [iniciar cada vez el dispositivo] o rápido [iniciar el dispositivo en el estado en que se dejó en el último uso].
- Las capacidades de almacenamiento.



Una vez comprobado todo, haremos clic en el botón *Finish* para finalizar la configuración e iniciar la creación del dispositivo.

Opciones sobre el dispositivo

Una vez creado el dispositivo, este aparecerá en la lista de dispositivos virtuales disponibles y se mostrarán sus principales características.

En la parte derecha tenemos la columna de acciones sobre el dispositivo (*Actions*). Disponemos de tres iconos: para abrir el dispositivo, para abrir los archivos de almacenamiento interno del dispositivo (si este está en funcionamiento) y un menú desplegable con varias opciones, al que también podremos acceder pulsando el botón derecho del ratón sobre la fila del dispositivo.

Desde este menú vamos a poder realizar varias opciones:

- Editar (*Edit*) la configuración del dispositivo, por si deseamos aumentar el almacenamiento, el nombre, las cámaras o cualquier otra opción de este.
- Duplicar (*Duplicate*) el dispositivo.
- Limpiar (*Wipe*) el dispositivo, para dejarlo en el estado original.
- Realizar un arranque en frío (*Cold Boot Now*), reiniciando el dispositivo.
- Mostrar los ficheros del dispositivo en nuestro disco (*Show on Disk*).
- Mostrar una ventana con un resumen de las características del dispositivo (*View Details*).

- Eliminar (*Delete*) el dispositivo.
- Parar (*Stop*) el dispositivo si este está en funcionamiento.

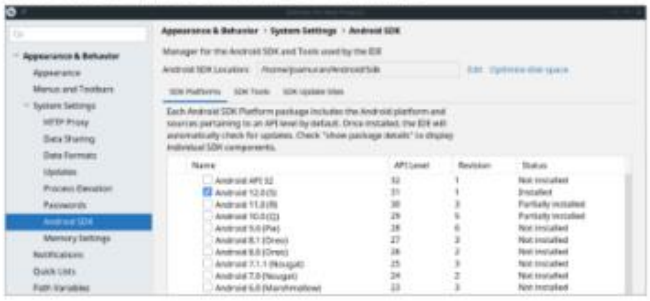
Disponéis de más información sobre la creación de dispositivos y su gestión en la documentación oficial de Google: **https://developer.android.com/studio/run/managing-avds**

Activando la aceleración de hardware para el emulador

Como se sugiere en la instalación del dispositivo virtual en sistemas GNU/Linux, es posible activar la aceleración de hardware para el emulador, para mejorar su rendimiento.

Con el fin de poder utilizar la aceleración, debemos tener en cuenta que necesitamos una versión mínima del SDK (la 17, aunque la recomendable es la 26.1.1 u otra posterior, según la documentación).

Podemos consultar la versión del SDK accediendo al gestor de este (bien desde la ventana de inicio, donde lanzamos el Device Manager, bien desde el menú de herramientas).



Como vemos, la versión del SDK no tiene que ser la misma que configuremos para nuestros dispositivos.

Por otra parte, la versión del dispositivo al que queramos añadirle aceleración deberá tener un nivel de API 10 o superior (Android 2.3.3).

Además, se requiere que nuestro equipo soporte extensiones de virtualización (VT, VT-x, vmx en el caso de Intel, o AMD-V o SVM en el caso de AMD). La mayoría de procesadores modernos soportan estas tecnologías.

En la documentación también se comentan un par de restricciones, y es que no podemos utilizar la virtualización si ya nos encontramos dentro de una máquina virtual. Asimismo, dado que hipervisores como VirtualBox, VMWare o Docker utilizan esta tecnología, no vamos a poder virtualizar en estos entornos de forma simultánea a nuestro emulador Android.

Con todo esto en cuenta, podemos proceder a la instalación de KVM. Para ello, desde la línea de comandos de Ubuntu, con una versión posterior a la 18.04, lanzaremos la orden siguiente para instalar los paquetes necesarios:

```
sudo apt-get install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils
```

Una vez hecho esto, podemos comprobar si la aceleración puede usarse en nuestros dispositivos de dos modos:

Opción 1. Mediante el propio emulador, con la opción *-accel-check* desde la línea de comandos. Para hacerlo, suponemos que nuestro emulador se instaló en la carpeta *~/Android/Sdk/emulator* e invocamos el comando siguiente:

```
$ cd ~/Android/Sdk/emulator
$ ./emulator -accel-check
    accel:
    0
    KVM (version 12) is installed and usable.
    accel
```

Observamos que se indica que KVM está instalado y es funcional.

Opción 2. Mediante el programa *kvm-ok* del paquete *cpu-checker*. Para ello, en primer lugar debemos instalar dicho paquete:

```
$ sudo apt install cpu-checker
```

Y ejecutar el comando kvm-ok:

```
$ kvm-ok
    INFO: /dev/kvm exists
    KVM acceleration can be used
```

Que nos indica que la aceleración está también disponible.

Cuando lo hayas comprobado, puedes probar a lanzar el emulador desde la línea de comandos, con y sin aceleración, y ver la diferencia entre ambas. Desde el directorio donde tenemos el emulador instalado listamos en primer lugar los dispositivos disponibles:

```
$./emulator -list-avds
    Pixel_3_API_29
```

Como vemos, se muestra el dispositivo Pixel 3 que configuramos. Para lanzarlo, sin nada, podemos utilizar el parámetro -avd [`./emulator -avd Pixel_3_API_29`] o el símbolo @ [`./emulator @Pixel_3_API_29`].

Para lanzar el emulador sin aceleración, probaríamos:

```
$ ./emulator @Pixel_3_API_29 -accel off
```

Y para lanzarlo con aceleración:

```
$ ./emulator @Pixel_3_API_29 -accel on
```

Por defecto, cuando lanzamos el emulador desde el Device Manager o mediante la ejecución de una aplicación, la aceleración se configura de forma automática.

Podéis encontrar más información sobre estos temas y cómo trabajar la aceleración en otros sistemas en la documentación oficial de Android:

- Artículo *Configura la aceleración de hardware para Android Emulator* [https://developer.android.com/studio/run/emulator-acceleration]
- Artículo *Cómo iniciar el emulador desde la línea de comandos* [**https://developer.android.com/studio/run/emulator-commandline**].

Ejercicio 3

Indica si las afirmaciones siguientes acerca del SO Android son ciertas o falsas.

a) Se adapta a una gran cantidad de pantallas y resoluciones. **V**
b) No soporta pantallas multitáctiles de forma nativa. **F**
c) Ofrece almacenamiento local a través de una base de datos SQLite. **V**
d) Las aplicaciones se programan en Java, Kotlin o Swift, y se requiere una licencia de desarrollador. **F**
e) Ofrece un extenso catálogo de aplicaciones a través de Google Play. **V**

Ejercicio 4

Selecciona la opción correcta:

¿Qué es Material Design?

1. Una versión inicial de Android.
2. Una característica introducida en Ice Cream Sandwich para unificar los diseños de tabletas y móviles.
3. Un conjunto de especificaciones de diseño, lanzadas con Android 5.0 Lollipop.
4. Una característica de Android 7.0 que permite dividir la pantalla.
5. Una actualización de diseño que permite al sistema ajustarse a la configuración del usuario.

¿Cuáles de los elementos siguientes integran la arquitectura en capas de Android?

1. Las aplicaciones, la API, las librerías nativas de C y C++, el Runtime, la capa de abstracción y el kernel de Linux.
2. Las aplicaciones, la interfaz gráfica, las librerías nativas de C y C++, el Runtime, la capa de abstracción y el kernel de Linux.
3. Las aplicaciones, la interfaz gráfica, las librerías nativas de Java y Kotlin, el Runtime, la capa de abstracción y el kernel de Linux.
4. Las aplicaciones, la API, las librerías nativas de Java y Kotlin, el ART, la capa de abstracción y el kernel de Linux.
5. Las aplicaciones, la API, las librerías nativas de C y C++, el Runtime, el ART y el kernel de Linux.

¿Qué componente de Android, actualmente integrado en Jetpack, permite utilizar funcionalidades nuevas o equivalentes en versiones anteriores de Android?

1. Material Design.
2. Material You.
3. La capa HAL.
4. Las bibliotecas de compatibilidad.
5. La propia API.

¿Qué componente nos permite gestionar dispositivos virtuales?

1. Android Emulator.
2. Android Debug Bridge (ADB).
3. SDK Manager.
4. Device Manager.
5. VirtualBox.