Android

Explicación de práctica 6

Sistemas Operativos

Facultad de Informática
Universidad Nacional de La Plata

2021

Principales características

- Android utiliza el kernel Linux (Android 11 - Linux 5.4.x)
- Android aprovecha Linux para:
  - Abstracción de hardware.
  - Administración de memoria
  - Administración de CPU.
  - Networking.
  - Seguridad
- El usuario no nota el Linux subyacente.

- El nombre proviene de la novela: ¿Sueñan los androides con ovejas eléctricas? de Philip K. Dick.
- 2003 - Android Inc. idea inicial.
- 2005 - Android Inc. es adquirida por Google.
- 2007 - Se forma la Open Handset Alliance (Google, HTC, Motorola, Samsung, Qualcomm, Texas Instruments y otros).
- 2008 - Versión 1.0

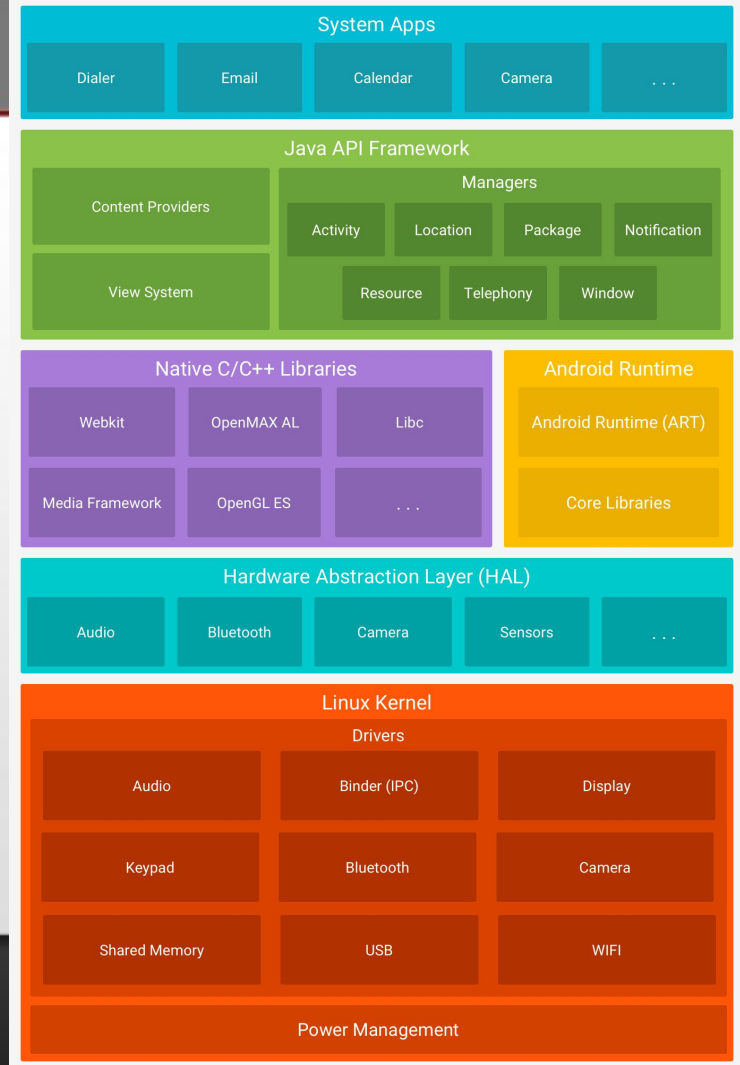
HTC Dream
Primer dispositivo
con Android

- Nombres de dulces desde v1.5

- Android 9.0 Pie
- Android 10 – la privacidad por bandera
  - Es el primero en no llevar un nombre postre.
  - Protege la actividad, datos y ubicación de los usuarios.
- Android 11
  - Versión “actual”: Android 11 (2020)

Stack

Pila de software de código abierto basado en Linux



Procesos e Hilos - Componentes

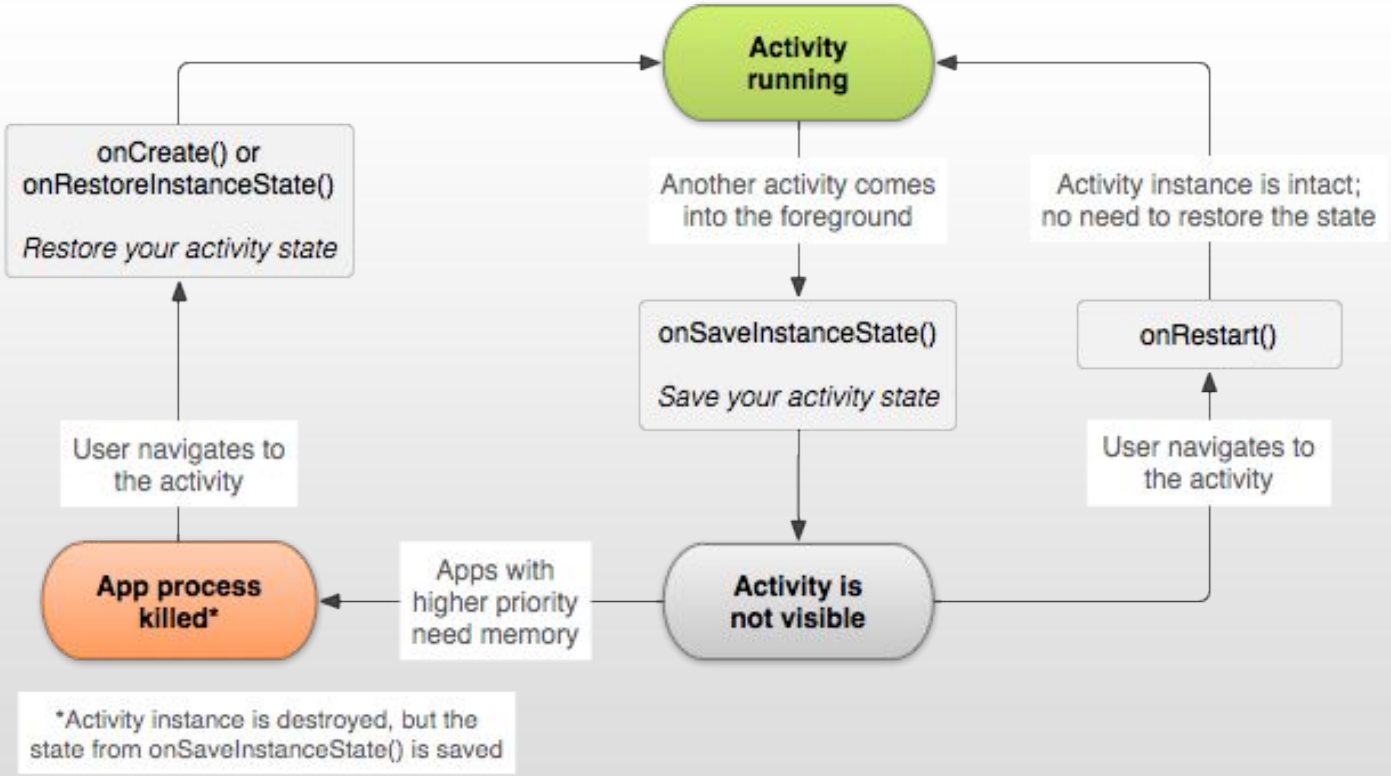
- Componentes: *activity*, *service*, *receiver*, *provider*.
- Cuando se invoca el primer componente de una aplicación se crea un proceso Linux con un único thread (*main*/*UI* thread).
- Por defecto todos los componentes de una aplicación se ejecutan sobre ese mismo thread del proceso que represente a la aplicación.
- Los componentes pueden ser configurados para que ejecuten sobre diferentes procesos (*android:process*).
- Es posible crear threads dentro de cada proceso → importante para el acceso a la *UI* (*UI toolkit thread unsafe*).

Procesos e Hilos - Clasificación

- Android elimina procesos bajo demanda ante la ausencia de memoria → no cuenta con área de intercambio [1].
- Decisión en base a jerarquía de procesos clasificados en:
  - *Foreground process*
  - *Visible process → onPause()*
  - *Service process → startService()*
  - *Background process (LRU) → onStop()*
  - *Empty process*

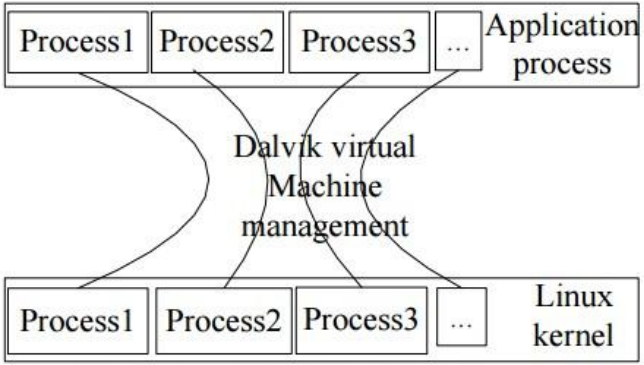
[1] https://zerocredibility.wordpress.com/2009/08/24/why-android-swap-doesnt-make-sense/

Procesos e Hilos - Seudo swapping



Procesos e Hilos - DVM/ART

- Individualizada por cada aplicación.
- Cada proceso es un proceso Linux.
- Cada thread es un thread Linux.



The relationship between Dalvik virtual process and Linux process

DVM/ART vs. JVM

DVM/ART	JVM
Máquina basada en registros	Máquina basada en stack
Diseñada para ejecutar sobre poca memoria	Consume más memoria
Ejecuta sus propios bytes codes (*.dex*) [2]	Ejecuta bytes codes Java (*.class*)
Uni-plataforma → Android	Multi-plataforma
Ejecutable → *.apk*	Ejecutable → *.jar*

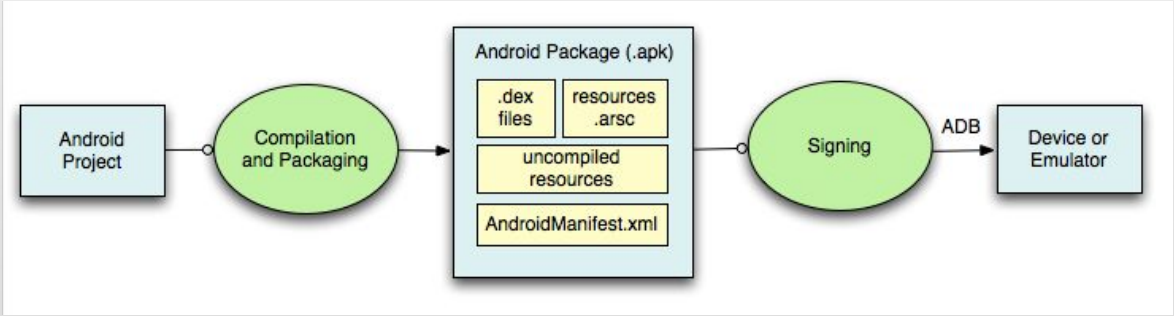
Mejoras de *ART* [ref]:

- Incorporación de compilación *Ahead-of-time* (AOT) utilizando **dex2oat** además de la compilación *just-in-time* (JIT) existente.
- Optimización del garbage collection (GC).

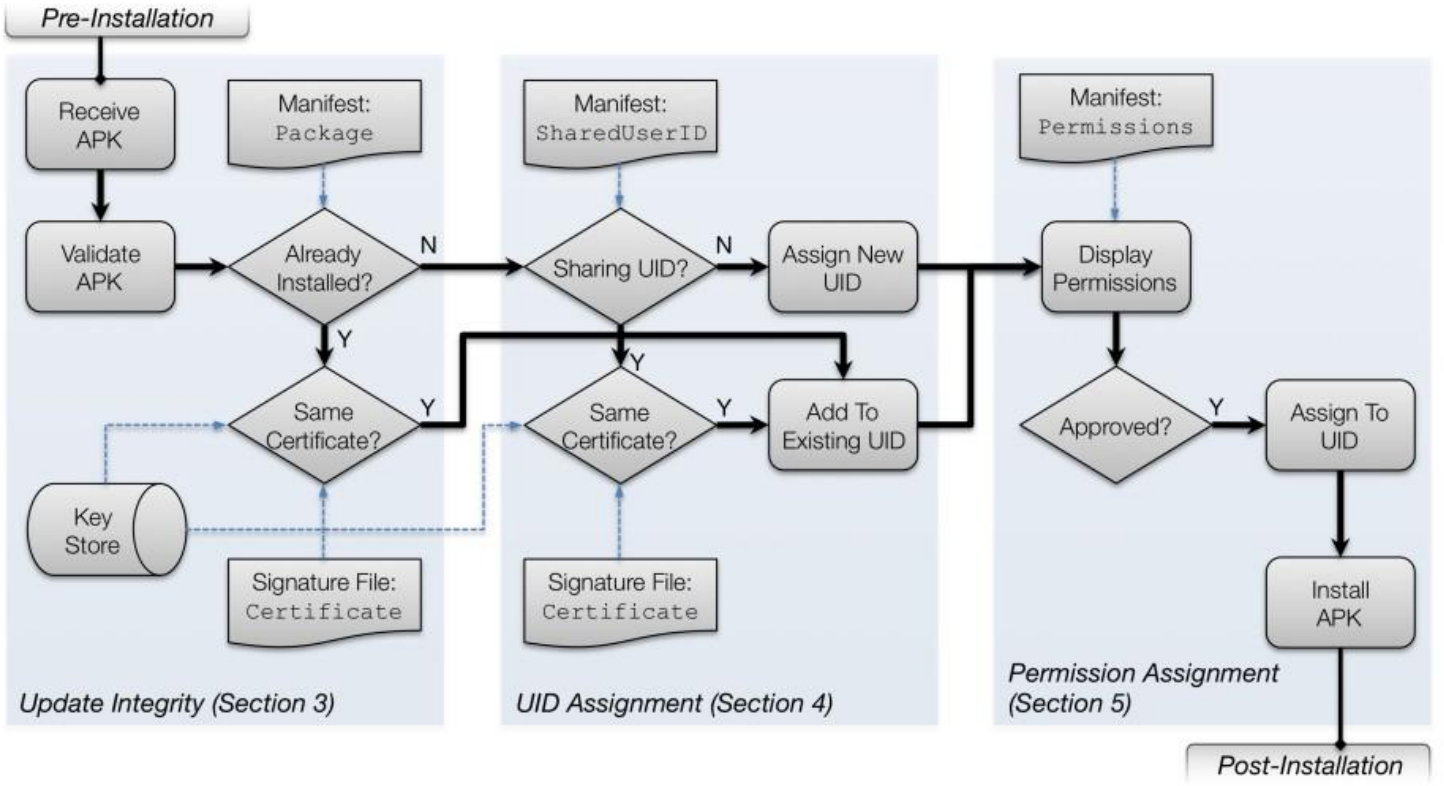
[2] https://source.android.com/devices/tech/dalvik/dex-format

Aplicaciones - Building [ref]

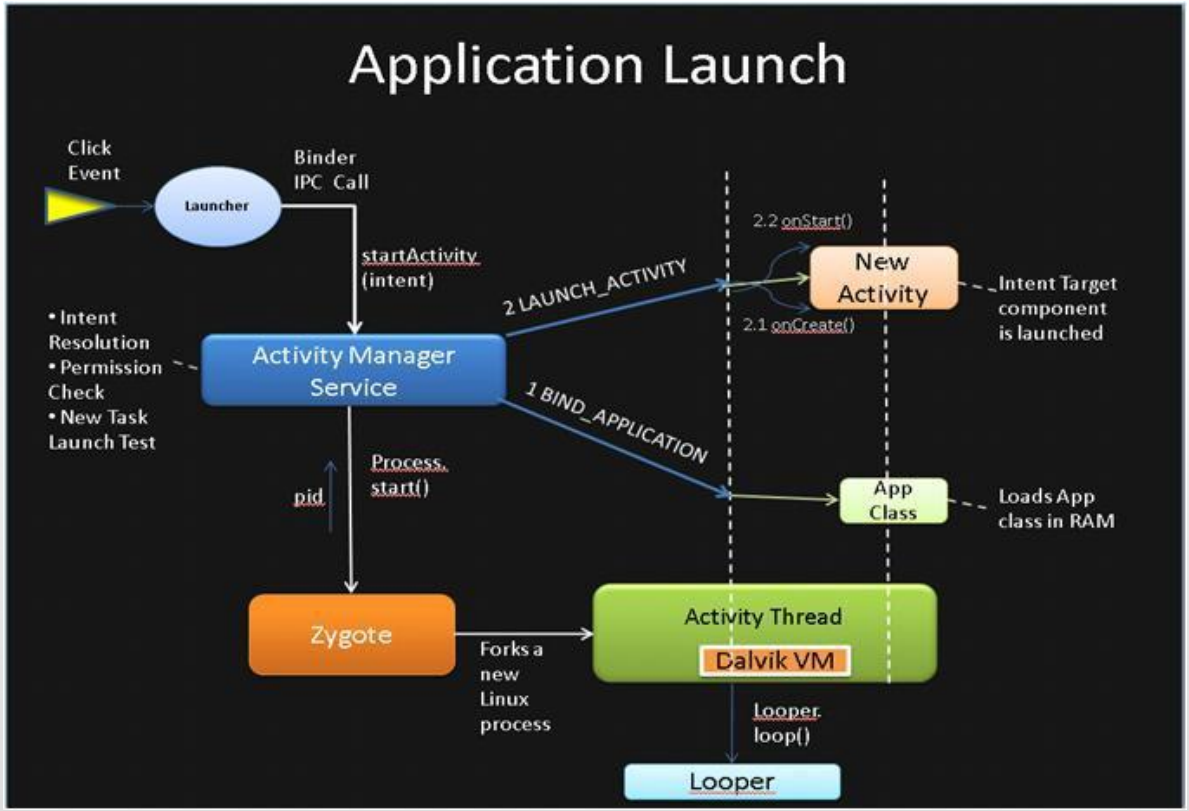
- Un *android package* contiene todo lo necesario para ejecutar la aplicación en un dispositivo.
- **Gradle** es el *application builder* oficial del *SDK* de Android (*build.gradle*).
  - Debug mode → automático.
  - Release mode.



Aplicaciones - Installing



Aplicaciones - Launching



Seguridad - Datos de aplicación

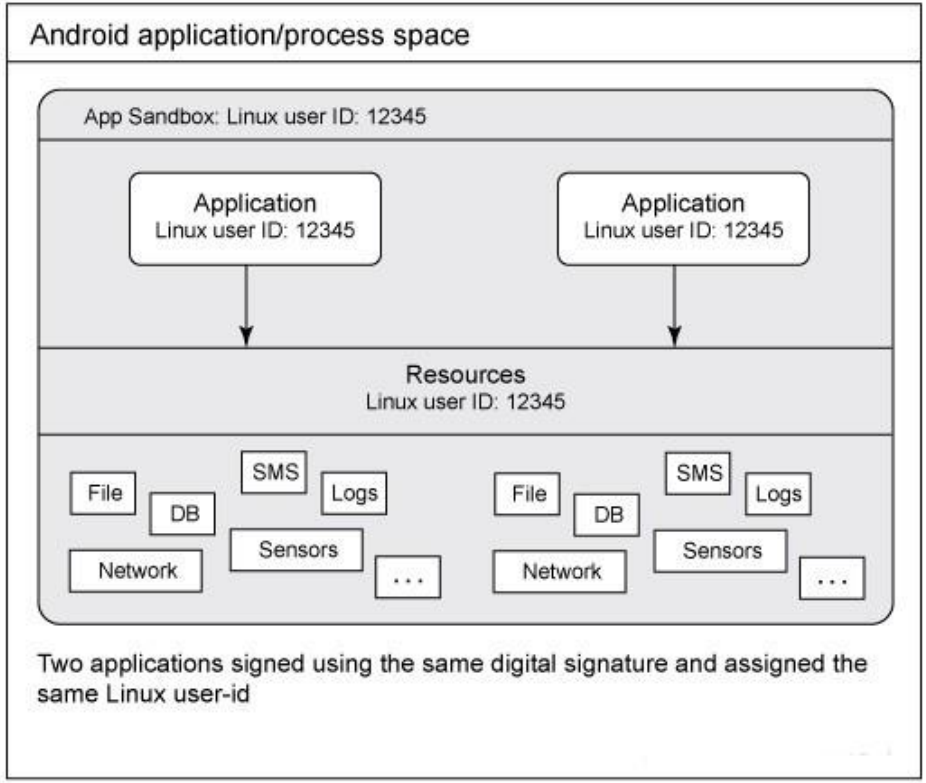
- Ninguna aplicación puede ejecutar operaciones que afecten a las demás.
- Solo pueden escribir y leer datos privados de la aplicación.
- Las aplicaciones comparten datos de manera explícita.

Seguridad - Id. de usuario y permisos sobre archivos

- Cuando se instala un *.apk*, Android le otorga un *userId* de Linux definitivo.
- En otro dispositivo el mismo paquete podría tener otro *userId*.
- Dos aplicaciones con el mismo *userId* son tratadas como la misma aplicación.
- Mismo *userId* = Mismo usuario Linux = Misma aplicación = Mismos permisos (*UGO*) sobre los archivos de la aplicación.
- Las aplicaciones que comparten el *userId* tienen que compartir la firma. Es decir que deben ser firmados por la misma clave privada.

```
# ls -l data/data/ | grep brow
# drwxr-x--x u0_a14 u0_a14 2016-05-01 17:55 com.
    android.browse
```

Seguridad - Ejemplo shared user id



Two applications signed using the same digital signature and assigned the same Linux user-id

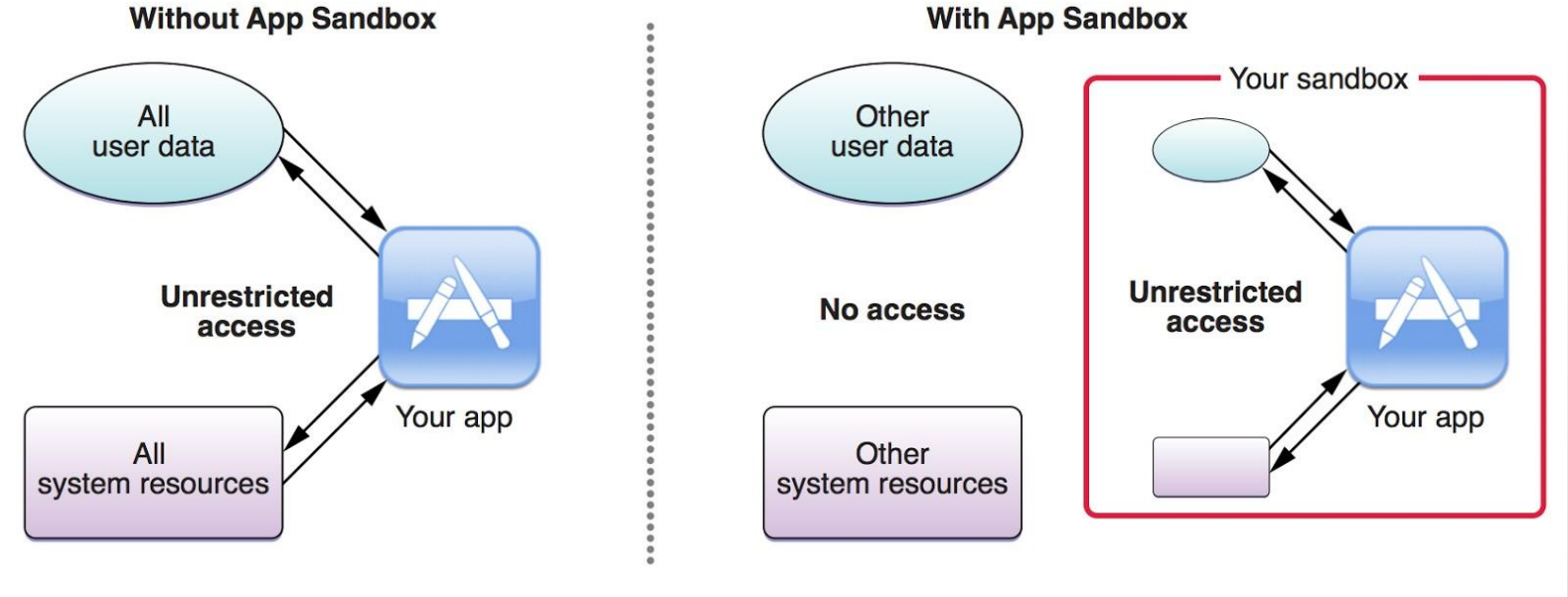
Seguridad - Recursos

- Se debe declarar el acceso a los recursos de manera estática (*Manifest.xml*) → dinámico a partir de v6.0
- Cuando la aplicación es instalada el usuario debe dar su consentimiento.
- *SecurityException*

```
<manifest xmlns:android="http://schemas.android.com/
     apk/res/android"
      package="com.android.app.myapp" >

      <uses-permission android:name="android.
           permission.RECEIVE_SMS" />
      <uses-permission android:name="android.
           permission.INTERNET" />
     ...
</manifest>
```

Seguridad - Application sandbox



- Solo una porción de código Android corre como *root*.
- ¿Cómo hacen diferentes reproductores para acceder a la música?

Seguridad - Signing [ref]

- Todas las aplicaciones (*.apk*) tienen que ser firmados digitalmente.
- Se emiten certificados auto-firmados (*KeyTool*):
  - Debug mode → *debug key* → el *keystore* se crea automáticamente (*$HOME/.android/debug.keystore*).
  - Release mode → *developer's private key* → manualmente.

Seguridad - Repaso y Overview

- A nivel de *AndroidManifest* → permisos a recursos que termina de conceder o configurar el usuario (aplicado por el *sandbox*).
- A nivel de las aplicaciones y sus archivos → certificado, *userId* y permisos *UGO* (aplicado por el *sandbox*).
- A nivel de file system → particiones *read-only* (/system).
- A nivel de *Linux* y *ART* (aislamiento de aplicaciones).

Almacenamiento [ref]

- Almacenamiento interno
  - *MODE PRIVATE*, *MODE WORLD READABLE* o *MODE WORLD WRITEABLE* → desde v7.0 *FileProvider*.
- Shared preferences [ref]
  - Se puede manejar una colección de ellas.
  - Se almacenan a través de archivos XML.
  - Aplican como almacenamiento interno.
  - */data/data/<package name>/shared prefs/*
- Almacenamiento externo → READ EXTERNAL STORAGE y WRITE EXTERNAL STORAGE
- SQLite:
  - Bajo licencia *GPL*.
  - Actúa sobre archivos ordinarios.
  - Cumple las propiedades ACID.
  - */data/data/<package name>/databases/*

File system - Tipos de memorias

- Raw NAND flash:
  - Subsistema *MTD* (Memory Technology Device).
  - */dev/block/mtdblockN*

```
$ cat /proc/mtd
dev: size erasesize name
 mtd0:       05660000   00020000   "system"
 mtd1:       04000000   00020000   "userdata"
 mtd2:       04000000   00020000   "cache"
```

- SD, MicroSD y eMMC (Flash Translation Layer):
  - Driver *mmcblock* (Multimedia Card Block).
  - */dev/block/mmcblk[chip number]p[partition number]*

```
$ ls -l /dev/block/platform/msm_sdcc.1/by-name
cache -> /dev/block/mmcblk0p36  system ->
/dev/block/mmcblk0p38  userdata ->
/dev/block/mmcblk0p40
```

File systems - Tipos

- YAFFS
  - Consume menos memoria.
  - Divide los archivos en páginas.
  - GPL.
  - v1 y v2.
  - Single-theading.
  - Por lo general, hasta Froyo (v2.2).
  - Especializado para funcionar en memorias de tipo raw NAND.
- Ext4
  - Multi-theading.
  - Desde Gingerbread (v2.3).
  - Utilizado generalmente en memorias SD, MicroSD o eMMC.

File systems - Puntos de montaje principales

- /system
- /system/bin
- /data/app
- /data/data/<package_identifier>
- /recovery
- /boot
- /cache

File systems - Puntos de montaje principales

- /system
- /system/bin
- /data/app
- /data/data/<package_identifier>
- /recovery
- /boot
- /cache

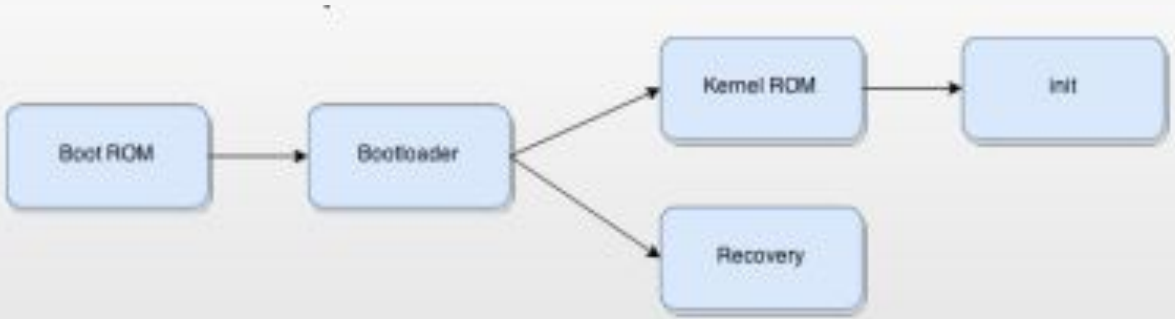
Rooting - Definición

- Rooting Android device → darle permisos de root al dispositivo

- Jailbreaking iOS device → también necesario para instalar aplicaciones no autorizadas.
  - Elimina las restricciones

Bootloader - Definición

- Inicializa el hardware y carga el initramfs.
- Permite bootear una imagen → Android o recovery.
- Está fuera de lo que es Android.



- ¿Por qué está bloqueado si Android es open source?.
- Desbloquearlo implica perder la garantía del dispositivo → impuesto por Digital Rights Management.

Bootloader - Fastboot

- Protocolo USB.
- Permite bootear customs ROMs.
- Permite flashear una partición.

  fastboot flash boot out/target/product/hikey/boot.

  img

- Permite desbloquear el bootloader (solo si el dispositivo lo soporta).

  fastboot oem get_unlock_data

  fastboot oem unlock [unlock code]

Bootloader - Tipos

- Open bootloader → las compañías no lo desean.

- Locked bootloader Carrier ID → pérdida de garantía y reseteo de fábrica.

Boot.img & initramfs - Boot.img

- Obtención:
  - Extracci´on de una imagen original → de un dispositivo o del SDK de Android.
  - Descarga desde un sitio confiable (MIUI/LineageOS) → zip.
- División:
  - A través de unmkbootimg.
  - Estructura:

```
+-----------------+
| boot header     | 1 page
+-----------------+
| kernel          | n pages
+-----------------+
| ramdisk         | m pages
+-----------------+
| second stage    | o pages
+-----------------+

n = (kernel_size + page_size - 1) / page_size
m = (ramdisk_size + page_size - 1) / page_size
o = (second_size + page_size - 1) / page_size

0. all entities are page_size aligned in flash
1. kernel and ramdisk are required (size != 0)
2. second is optional (second_size == 0 -> no second)
```

Boot.img & initramfs - Initramfs

- Desempaquetado y descompresión de initramfs (cpio + gzip):

```
gunzip -c ../your-ramdisk-file | cpio -i
```

- Archivo default.prop → contiene propiedades de configuración para la inicialización del sistema.
- Habilitar modo inseguro: propiedad ro.secure debe estar en falso → demonio de adb (adbd) con permisos de root.
- Empaquetar y comprimir initramfs inseguro:

```
find . | cpio -o -H newc | gzip > ../newramdisk.cpio.gz
```

Boot.img & initramfs - Reconstrucci´on de boot.img

- A través de mkbootimg

```
mkbootimg --kernel zImage --ramdisk
    insecure_initramfs.cpio.gz --base 0x80200000 --
    cmdline 'androidboot.hardware=qcom user_debug=31
    zcache' -o new_boot.img
```

Booteo/flasheo y testeo

- Booteo (método volátil):

$ fastboot boot new_boot.img

- Flasheo (método no volátil):

$ fastboot flash boot new_boot.img

- Acceso al dispositivo mediante adb como usuario root en lugar de shell.

$ adb shell

¿Preguntas?