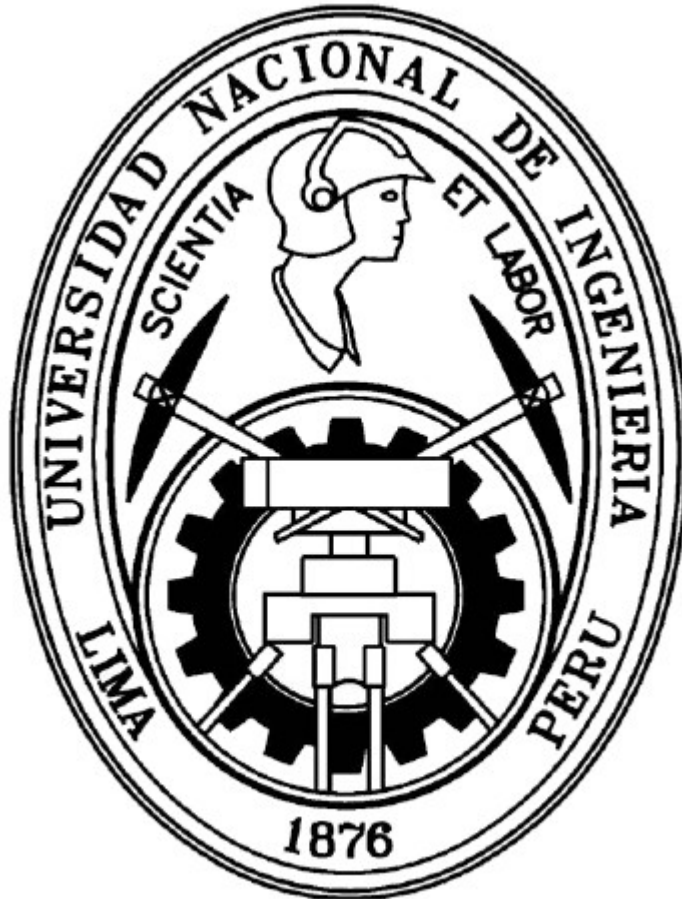


Laboratorio 1



Apellidos: Moreno Vera

Nombres: Felipe Adrian

Código: 20120354I

**Asignatura: Programación en Dispositivos Móviles
(CC481)**

2016 - I

Indice

| | |
|--------------------------|-------------|
| Actividad 1 | (3) |
| Actividad 2 | (7) |
| Actividad 3 | (7) |
| Actividad 4 | (8) |
| Actividad 5 | (10) |
| Actividad 6 | (13) |

Actividad 1

1. Buscar a qué se llama API en google.

Una application programming interface (API) es un conjunto de rutinas, protocolos y herramientas para la construcción de software y aplicaciones.

2. Que API Level corresponde con cada versión de Android y el por qué de esta numeración y los saltos que se producen. Comentar cuáles son de desarrollo netamente y cuáles funcionales a los dispositivos.

3. Exponer las novedades más importantes en cada API, en cuanto a métodos, sensores, etc.

Haciendo el ejercicio 2 y 3 a la vez ...

La numeración corresponde a plataformas lanzadas como estables, sus nombres y numeración son conforme van siendo lanzadas. Aparte cada versión o API tiene nombres comerciales que los identifican, por ejemplo, Cupcake (v 2.3) y curiosamente han sido en orden alfabético.

Android 1.0 (API 1) Primera versión de Android. Nunca se utilizó comercialmente, por lo que no tiene mucho sentido desarrollar para esta plataforma **(funcional con dispositivos)**.

Android 1.1 (API 2) Se corrigieron errores que habían en la primera versión, **Se puede desarrollar una aplicación con esta versión (netamente desarrollo), pues esta versión solo sirvió para el teléfono HTC Dream.**

Android 1.5 (API 3) Esta versión mantiene el 0.1% hasta el enero del 2013, se incluyó la posibilidad del teclado en pantalla con predicción de texto, grabación de audio y video, bluetooth, autorotación, **funcional con dispositivos y desarrollo.**

Android 1.6 (API 4) Se lanzo el SDK manager basado en núcleo linux 2.6.29, permite la conversión de texto a audio aparece el atributo XML, onClick, que describe y especifica una vista, Soporte resoluciones pantalla WVGA, **desarrollo con mejoras.**

Android 2.0 (API 5) se lanzo con el sdk 2.0. incluyen cambios, bluetooth 2.1, nueva funcionalidad que permite sincronizar adaptadores y conectarse a cualquier dispositivo, aumento o reducción de pantalla, soporte para HTML5, y la clase MotionEvent que soporta eventos en pantallas multitáctiles, **netamente desarrollo(los fabricantes saltaron del 1.6 al 2.1).**

Android 2.1 (API 7) Reconocimiento de voz y desarrollo de fondos de pantalla animados, el paquete webKit que incluye nuevos métodos para manipular base de datos almacenadas en

internet, obtener información sobre la red, **netamente desarrollo (esta versión incluye una mínima actualización, así que se considera como API de desarrollo).**

Android 2.2 (API 8) Esta versión hace un salto en la mejora y gestionamiento de procesos (aumenta la velocidad de ejecución de aplicaciones de 2 a 5 veces más rápida), añade la pregunta de “ desea instalar X complemento”, se añade soporte a javascript v8 utilizado en chrome, incluyendo soporte de Adobe Flash10.1, proporciona una copia de seguridad de datos que se puede realizar desde una aplicación, mensaje y actualización automática de paquetes y nuevas versiones de apps, Ahora podemos poner nuestros dispositivos como anclaje de zona wifi (ser un wouter andante :D), soporte para OpenGL, y nuevos modos de “automovil” y “noche”, conexión 3G, **funcional con dispositivos y netamente desarrollo.**

Android 2.3 (API 9) Actualizado y diseño de la interfaz usuario con incrementos en velocidad y simpleza. Entrada de texto del teclado virtual más rápida, entrada por voz, funcionalidad copiar pegar, nuevo gestor de descargas, soporte de multiples cámaras, como la normal y la frontal. **Funcional con dispositivos.**

Android 2.3.3 (API 10) Sdk 2.3, se baso en el núcleo linux 2.6.35 mejora de la funcionalidad copiar pegar y cortar, reconocimiento facial en la segunda cámara, la máquina virtual Dalvik introduce un nuevo recolector de basura que minimiza las pausas de aplicación, mejorando la mejor animación y el aumento de la capacidad de respuesta en juegos y similares, **funcional y desarrollo**

Android 3.0 (API 11) Da soporte a dispositivos con pantallas amplias como las tablets, se hacen cambios en la barra de acciones, donde las aplicaciones pueden mostrar un menú siempre visible, las teclas físicas son reemplazadas por teclas de pantalla, se mejoran gráficos 2D/3D gracias al renderizador OpenGL, acelerado por hardware, se incorporan mejoras multimedia, como M3U a través de HTTP Live Streaming, Transferencia de archivos a través de USB con protocolos MTP y PTP, esta versión es compatible con aplicaciones anteriores, **funcional con dispositivos.**

Android 3.1 (API 12) Maneja dispositivos conectados por USB, protocolo de transferencia y video PTP/MTP y tiempo real RTP, soporte de teclados externos y dispositivos punteros, incrementada la capacidad de las aplicaciones para acceder a archivos de las tarjetas SD, por ejemplo para sincronización. **Funcional con dispositivos.**

Android 3.2 (API 13) Incrementada la capacidad de las aplicaciones para acceder a archivos de las tarjetas SD, por ejemplo para sincronización. Nuevas funciones de soporte de pantalla, dando a los desarrolladores un mayor control sobre la apariencia de la pantalla en diferentes dispositivos Android. **Desarrollo.**

Android 4.0 (API 14) Separación de widgets en una nueva pestaña, listados de forma similar a las aplicaciones, Nuevo navegador web con pestañas bajo la marca de Google [Chrome](#),

permitiendo hasta 15 pestañas, Grabación de vídeo a 1080P para dispositivos con Android de serie, **funcionalidad con dispositivos.**

Android 4.0.3 (API 15) Mejoras en el calendario, Mejoras en gráficos, bases de datos, corrección ortográfica y funcionalidades Bluetooth, desbloqueo por reconocimiento facial, **desarrollo.**

Android 4.1.2 (API 16) Se incorporan varias técnicas para corregir la fluidez del interfaz usuario, como: sincronismo vertical, triple búfer y aumentar la velocidad del procesador al tocar la pantalla, a 60fps, Se incorporan nuevo soporte para usuarios internacionales: como texto bidireccional y teclados instalables. Para mejorar la seguridad las aplicaciones son cifradas. También se permite actualizaciones parciales de aplicaciones, Google Now, **funcional y desarrollo.**

Android 4.2.2 (API 17) Los Widgets de escritorio pueden aparecer en la pantalla de bloqueo. Se incorpora un nuevo teclado predictivo deslizante al estilo Swype. Posibilidad de conectar dispositivo y TVHD mediante wifi (Miracast). Mejoras menores en las notificaciones. Nueva aplicación de cámara que incorpora la funcionalidad Photo Sphere para hacer fotos panorámicas inmersivas (en 360°), **funcionales.**

Android 4.3.1 (API 18) Se da soporte para OpenGL ES 3.0. Se mejora la seguridad para gestionar y ocultar las claves privadas y credenciales, Se da soporte para Bluetooth Low Energy (BLE) que permite a los dispositivos Android comunicarse con los periféricos con bajo consumo de energía, Soporte para Hebreo y Árabe, Locación de WiFi en segundo plano, Auto-completar en el marcado, Añadido el soporte para más de 5 idiomas, Opciones para creadores de Apps, Mejoras en el modo de conexión externa y de desarrollador (para actualizaciones via cable USB). **funcional**

Android 4.4.2 (API 19) Se ha arreglado un problema de batería que provoca el desgaste de está muy rápidamente, este arreglo solo fue en los celulares de Motorola ya que fueron los únicos que presentaron este problema, Se hizo posible que Android esté disponible en una gama aún más amplia de dispositivos, incluyendo aquellos con tamaños de memoria RAM de solo 512 MB. Para ello, todos los componentes principales de Android han sido recortados para reducir sus requerimientos de memoria, y se ha creado una nueva API que permite adaptar el comportamiento de la aplicación en dispositivos con poca memoria, El modo de inmersión en pantalla completa oculta todas las interfaces del sistema (barras de navegación y de estado) de tal manera que una aplicación use el tamaño completo de la pantalla, webViews (componentes de la interfaz de usuario para mostrarlas páginas web) ahora como se basa en Google Chrome, se puede mostrar contenido basado en HTML5, se añade la clase provider, con un método Content provider para gestionar los SMS, se incluye una máquina virtual ART, que es mucho más rápida ejecutando que Dalvik, pero solo era experimental. **Desarrollo y funcional.**

Android 4.4w.2 (API 20) Es la versión de android para relojes, AndroidTV u otros artefactos, hacen so de GPS, vienen con diferenteas apps para hacer deporte como Endomondo, Strava o Runstatic, conexión con dispositivos blueetooth, como auriculares inalámbricos o altavoces, como un reproductor de música, además cuenta con software de gestor de canciones como Spotify, Play Music, **Desarrollo.**

4. Explicar nuevas APIs y versiones de Android después de la 4.4.

Continuando con as descripciones ... ahora para versiones superiores a 4.4:

Android 5.0.1 (API 21) Es la extensión de Android a nuevas plataformas, incluyendo Google Wear, Google TV y Google Card. Hay un cambio significativo en la arquitectura, al utilizar la máquina virtual ART en lugar de Dalvik. Esta novedad ya había sido incorporada en la versión anterior a modo de prueba. ART mejora de forma considerable el tiempo de ejecución del código escrito en Java. Además se soporta dispositivos de 64 bits en procesadores ARM, x86, y MIPS. Muchas aplicaciones del sistema (Chrome, Gmail,...) se han incorporado en código nativo para una ejecución más rápida, también el modo de ahorro de batería se activa por defecto, incorpora soporte para OpenGL ES 3.1, se cambiaron los botones de pantalla a triangulo, circulo y cuadrado. Se incorporan nuevos sensores como el de pulso cardiaco, el de inclinación (para reconocer el tipo de actividad del usuario), y sensores de interacción compuestos para detectar ciertos gestos, se incluye WEREABLES y Google TV, y además introduce un modo de bloqueo que impide al usuario salir de una aplicación y bloquea las notificaciones. Esto podría utilizarse, por ejemplo, para que mientras un usuario realiza un examen, no pueda ver las notificaciones, acceder a otras aplicaciones, o volver a la pantalla de inicio, Soporte para CPUs de 64 de bits, La adición de 15 nuevos idiomas: Vasco, bengalí, birmano, chino (Hong Kong), gallego, islandés, kannada, Kirguistán, Macedonia, Malayo, marathi, nepalí, singalés, tamil y telugu, Se incluye una aplicación de linterna, que funciona en los dispositivos compatibles con un flash de cámara. **Funcional**

Android 5.1.1 (API 22) Se incluye tecnología Tap and Go permite a los usuarios migrar rápidamente a un nuevo dispositivo Android, el uso de NFC y Bluetooth para transferir Detalles de la cuenta Google, ajustes de configuración de datos del usuario y las aplicaciones instaladas, Soporte para múltiples tarjetas SIM, Protección de dispositivos, si un dispositivo se pierde o es robado permanecerá bloqueado hasta que se inicie sesión con una cuenta de Google - incluso si el dispositivo se restablece a la configuración de fábrica, **Desarrollo.**

Android 6.0 (API 23) Administrador de permisos: Podemos decidir a qué permisos se puede acceder cada aplicación, como los permisos de calendario, contactos, cámara, micrófono, SMS, sensores, teléfono y ubicación, Google Now on Tap: es la expansión de Google Now a todo nuestro dispositivo. Con una pulsación prolongada nos aparecerá una tarjeta con información referente a lo que está apareciendo en pantalla. Por ejemplo, si estamos leyendo un correo de un amigo que nos propone ir al cine a ver una película, al pulsar "Now on Tap" nos aparecerá la ficha de esa película, Android Pay, Soporte de huellas dactilares, Android hará restauraciones y copias de seguridad de la data completa y automáticas de nuestras

aplicaciones tras cambiar de dispositivo o tras restablecerlo de fábrica para continuar con todos nuestros datos y partidas. Soporte oficial de tarjetas SD y USB, Compatibilidad con lápices bluetooth, pantalla de bloqueo mejorada, incorpora los sensores de Acelerómetro, para indicar vibraciones o de movimientos, Magnetómetro, GPS, giroscopio, Barómetro, termómetro, fotómetro, sensor de proximidad y sensor de presión dactilar. **Desarrollo y funcional**

Actividad 2

1. Realice el punto primero de tener un google ID.

Simplemente cuando tenemos nuestra cuenta gmail, buscamos la opción de Google +, una vez ahí, vamos a nuestro perfil y la dirección que aparece <https://plus.google.com/u/0/108256644602669790044> es nuestro google ID. Que es usado cuando descargamos aplicaciones y necesitan de nuestra información de nuestra cuenta google.

Actividad 3

1. Lo primero que tenemos que realizar es verificar si se encuentra el JDK instalado con la versión recomendada para Android. Que diferencia existe entre JDK y JRE ?

Verificando nuestra versión de JDK en nuestro ordenador:

```
jbot@jLap:~$ dpkg --get-architecture | grep -i jdk
ii oracle-java7-installer 7u80+7u60arm-0~webupd8~1 por lo qu
all Oracle Java(TM) Development Kit (JDK) 7
ii oracle-java7-set-default 7u80+7u60arm-0~webupd8~1 audio. Es
all Set Oracle JDK 7 as default Java
jbot@jLap:~$
```

En la página de Android Studio, se recomienda que se tenga una version de JDK 6 o superior y JDK 7 cuando desarrollas android 5.0 y superior.

Que diferencia hay entre JRE y JDK ?

JRE: Java Runtime Environment, Entorno de ejecución de Java. Es, básicamente, la Máquina Virtual de Java donde tus programas Java. También incluye los plugins de navegador, para la ejecución del Applet.

JDK: Java Development Kit, Es el completo Kit de Desarrollo de Software para Java, incluyendo la **JRE**, y los compiladores y las herramientas (como JavaDoc, y el Depurador de Java) para crear y compilar programas.

Entonces se diferencian en que para ejecutar aplicaciones, o plugins del sistema o del navegador, solo se necesita el JRE, pero si vas a desarrollar aplicaciones en java, necesitas todo el entorno, JDK.

2. Verificar e instalar el JDK.

Como tenemos instalado el Jdk 7, no es necesario instalar, pero en linux ubuntu (mi sistema host), se instala de la siguiente manera.

```
$ sudo add-apt-repository -y ppa:webupd8team/java  
$ sudo apt-get update  
$ sudo apt-get install oracle-java7-installer oracle-java7-set-default
```

3. Descargar Android Studio e instalarlo.

Repito, al ser usuario de linux ubuntu, hacer estos comandos una vez instalado el java JDK. Instalamos algunas dependencias antes ...

```
$ sudo apt-get install lib32z1 lib32ncurses5 lib32bz2-1.0 lib32stdc++6z
```

Y luego seguimos con AndroidStudio.

```
$ sudo add-apt-repository ppa:paolorotolo/android-studio  
$ sudo apt-get update  
$ sudo apt-get install android-studio
```

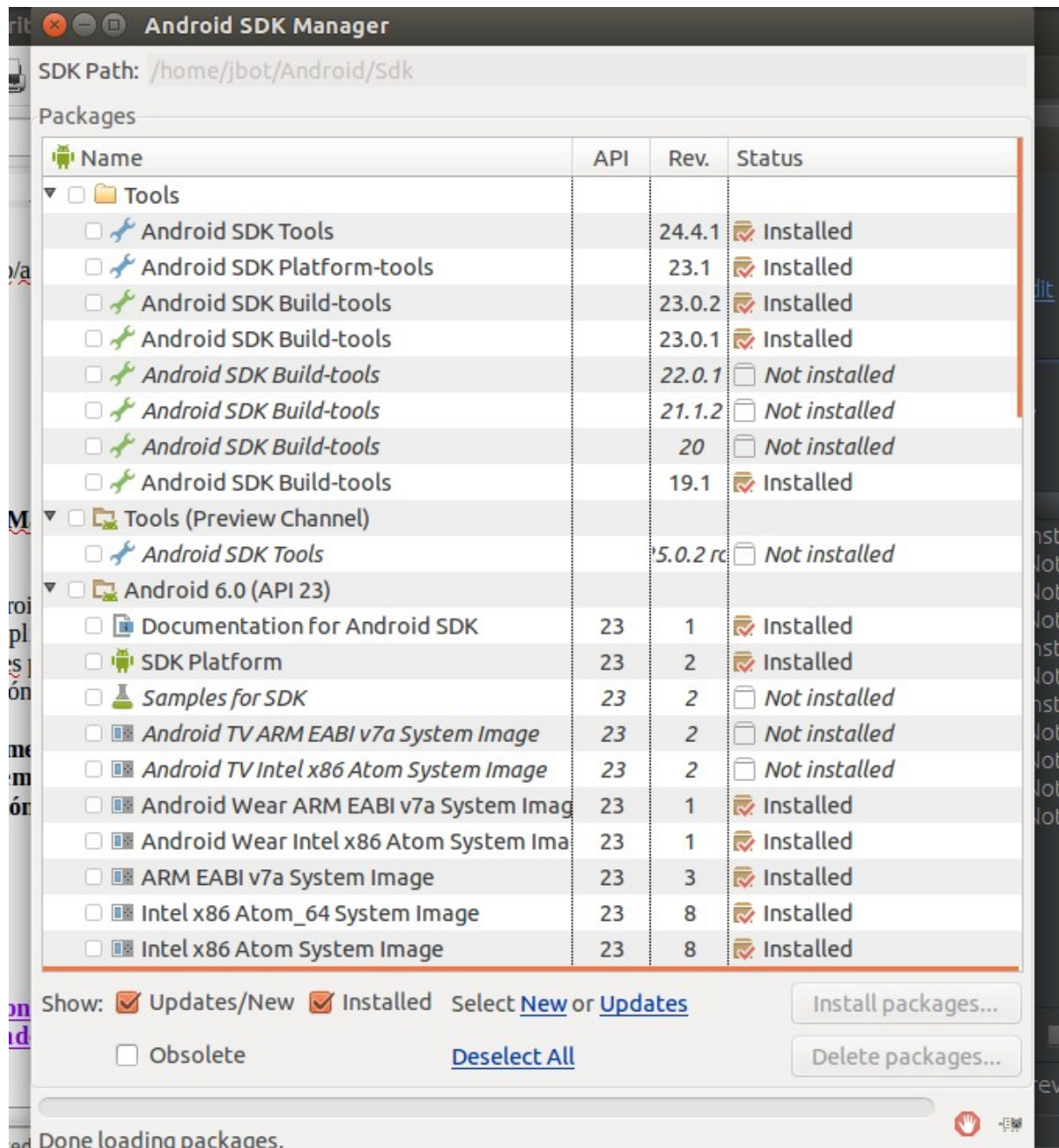
Actividad 4

1. Lo primero que abrimos es el “SDK Manager”. Antes de verlo, explique en no más de 3 líneas su funcionalidad.

SDK Manager: con nombre completo Android SDK Manager ... que vendría a ser: Android Software Development Kit, es el gestor de aplicaciones, que nos muestra mediante una conexión a internet todos las API y/o packages utiles para tener un buen entorno de desarrollo según el tipo de aplicación y según el tipo de versión en la que queramos hacer aplicaciones Android.

2. Descargar las APIs versión más recomendable para nuestras prácticas. La versión que se utilizará será según el estudio que hemos realizado en las actividades anteriores por el mayor número de personas con la versión indicada.

Se descargaron las API 17, 19 y 23.



3. Descargar mínimo 3 componentes más que crea necesario y explicar su importancia.

Android Support repository: Es un conjunto de librerías de código abierto que proporcionan versiones compatibles con versiones anteriores de la API framework para Android, cada soporte brinda poder usar niveles de API anteriores en una versión actual.

Google Play Services: Contiene las APIs que le permiten resolver cualquier problema en

tiempo de ejecución, tales como una falta de paquetes o fuera de la fecha de servicios de Google Play APK, nos brinda la manera de poder subirlo a google play y además que al correrlo en máquina, nos aparezca ícono de app.

Google repository: Repositorios de google para android.

Actividad 5

1. Busque información y explique brevemente cada uno de estos aspectos al realizar nuestra máquina virtual en el entorno.

1. Dispositivo: un dispositivo es un aparato o mecanismo que determina acciones, en este caso es en el cual se hacen todas las operaciones y se aloja el sistema Android.

2. Dimensiones de la Pantalla: Son el ancho y alto que tiene determinado dispositivo para proyectar o mostrar frames.

3. Memoria RAM: Memoria principal de la computadora, donde residen programas y datos, sobre la que se pueden efectuar operaciones de lectura y escritura.

4. Memoria Interna(almacenamiento): Memoria de almacenamiento de datos, que se mide en Mb o Gb (actualmente en móviles) y Tb en computadoras, que guardan cantidades de archivos y es la cantidad de memoria con la que viene dicho dispositivos (móvil o computadora).

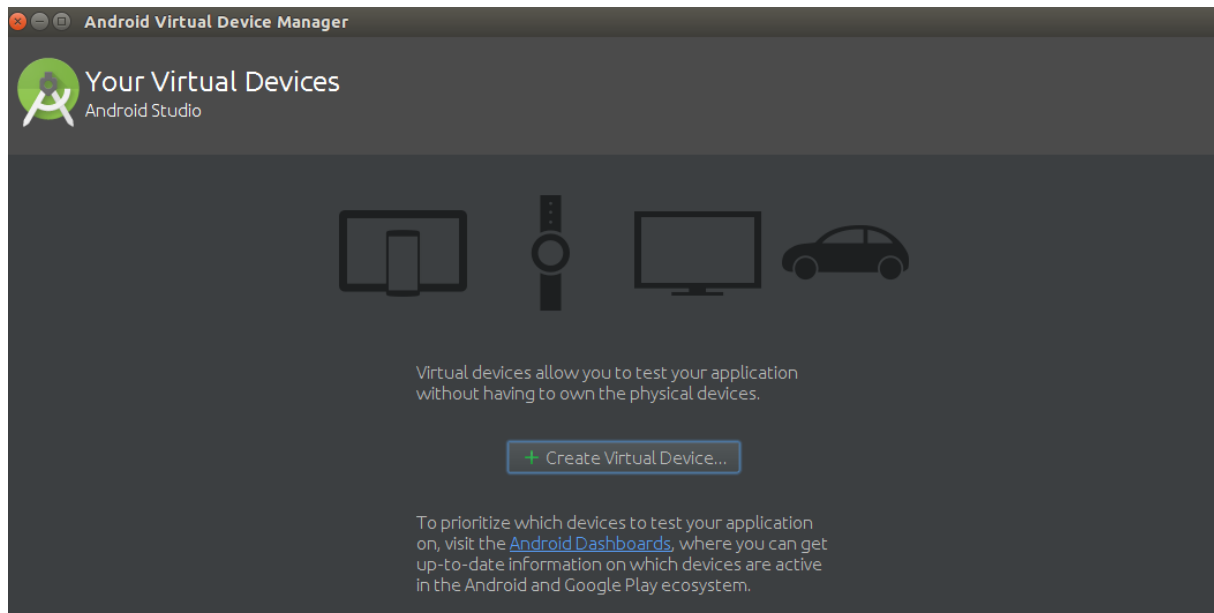
5. Soporte de SDCard: Viene a significar que tiene compatibilidad con memorias de tipo Security Digital, que son cards de almacenamiento pequeño.

6. Métricas de la máquina virtual: Se refiere a los recursos que consume nuestra máquina virtual, ya sea como uso de GPU, memoria, tráfico de red, etc.

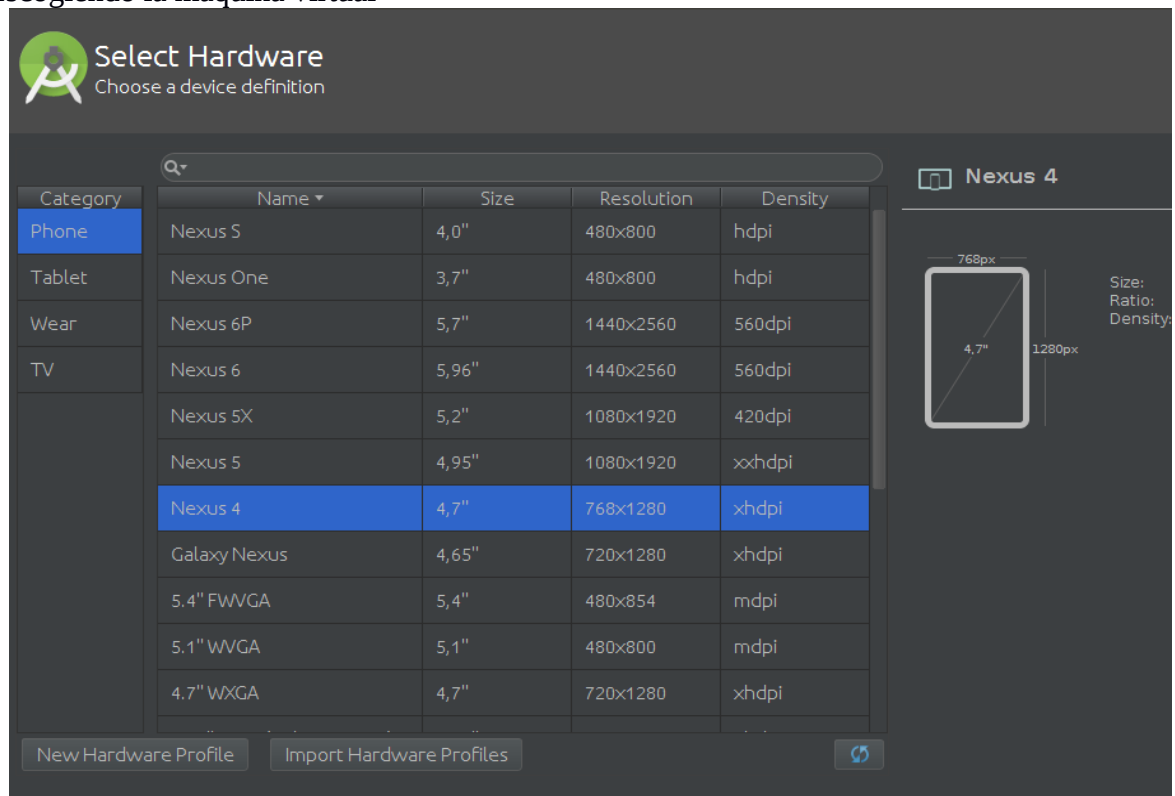
7. Integración con la virtualización: Cuando se genera la máquina virtual, interactuas y ves tus aplicaciones desarrolladas como si fuera un dispositivo android.

2. Una vez estudiado, creamos nuestra primera máquina virtual con los parámetros adecuados y la ejecutamos.


Creando la máquina virtual con AVD ...




Escogiendo la máquina virtual



Escogiendo la imagen del sistema ...


System Image
 Select a system image


| Release Name | API Level ▾ | ABI | Target |
|---------------|-------------|-------------|--------------------------------|
| Marshmallow | 23 | armeabi-v7a | Android 6.0 (with Google APIs) |
| Marshmallow | 23 | x86 | Android 6.0 (with Google APIs) |
| Marshmallow | 23 | x86_64 | Android 6.0 (with Google APIs) |
| Marshmallow | 23 | armeabi-v7a | Android 6.0 |
| Marshmallow | 23 | x86 | Android 6.0 |
| Marshmallow | 23 | x86_64 | Android 6.0 |
| KitKat | 19 | armeabi-v7a | Android 4.4 |
| KitKat | 19 | x86 | Android 4.4 |
| KitKat | 19 | armeabi-v7a | Android 4.4 |
| KitKat | 19 | x86 | Android 4.4 |
| Jelly Bean | 17 | armeabi-v7a | Android 4.2 |
| Jelly Bean | 17 | mips | Android 4.2 |
| Jelly Bean | 17 | x86 | Android 4.2 |
| Jelly Bean | 17 | armeabi-v7a | Android 4.2 |




KitKat


API Level
19
 Android
4.4
 Android Open
 Source Project
 System Image
x86

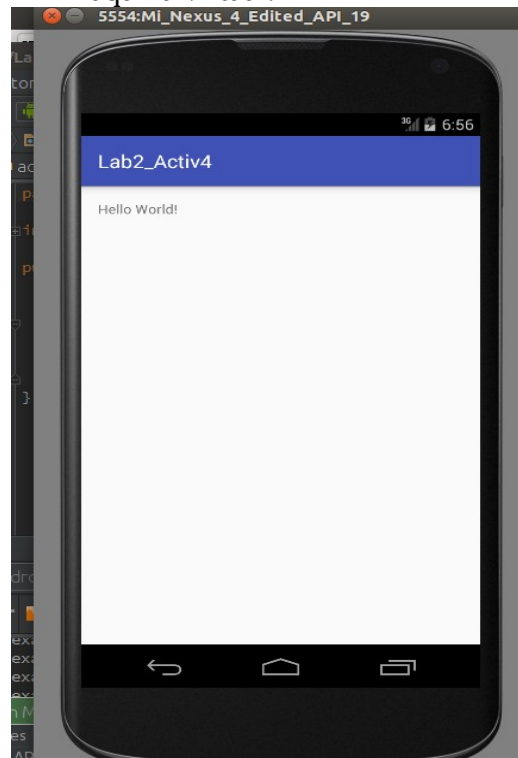
Recommendation
 Consider using a system image with Google
 APIs to enable testing with Google Play
 Services.

Máquina virtual creada


Your Virtual Devices
 Android Studio

| Type | Name | Resolution | API | Target | CPU/ABI | Size on Disk | Actions |
|---|--------------------------|-------------------|-----|---------------|---------|--------------|---|
|  | Mi Nexus 4 (Edited) A... | 768 × 1280: xhdpi | 19 | Android 4.4.2 | x86 | 300 MB |   ▾ |

Ejecutando la App con mi máquina virtual.



Actividad 6

1. Dada la teoría vista en clase realice las siguientes cuestiones para entender dos conceptos claves: las librerías y el *framework* de aplicación.

1. Describa 4 librerías de las expuestas en el esquema y su funcionalidad.

OpenGL ES: OpenGL ES (Open Graphics Library Embedded Systems) es una variante simplificada de la API gráfica OpenGL diseñada para dispositivos integrados tales como , PDAs y consolas de videojuegos. Aceleración y optimización de renderización de gráficos.

SQLite: es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña (~275 kiB) la librería esta escrita en C, Almacenamiento de Datos sin usar muchos recursos.

WebKit: Es una plataforma para aplicaciones que usan como base algún navegador web, mediante renderizado de HTML5, CSS, basado originalmente en el motor KHTML. Compatible con Chrome, Opera, Safari, Epiphany, etc. Renderizador de páginas web.

SSL: *Secure Sockets Layer* (SSL o «capa de conexión segura») es un protocolo criptográfico que proporciona comunicaciones seguras por una red, comúnmente Internet.

2. Describa 4 *framework* de las expuestas en el esquema y su funcionalidad. Describa algunas clases y métodos de relevancia, para ello busca su explicación en la web oficial de desarrolladores Android.

Location Manager: Esta clase proporciona el acceso del sistema a los servicios de localización. Estos servicios permiten a la aplicación actualizar periódicamente la posición geográfica del dispositivo, o matar a un Intent específico cuando el dispositivo entra a una zona geográfica determinada.

Métodos:

addGpsStatusListener() Añade al GPS el estado de listener.

addProximityAlert() gestiona la alerta de proximidad, con un determinado radio de tu posición actual.

getLastKnownLocation() Retorna la última localización registrada por el provider.

Bluetooth: Analizar en busca de otros dispositivos Bluetooth, Consultar el adaptador Bluetooth local para los dispositivos Bluetooth emparejados, Establecer canales RFCOMM, Conectarse a otros dispositivos a través de descubrimiento de servicios, Transferencia de datos hacia y desde otros dispositivos, Gestión de múltiples conexiones.

BluetoothAdapter representa nuestro adaptador bluetooth(radio) sirve como punto central para detectar otros dispositivos.

BluetoothDevice representa un dispositivo bluetooth distinto, hace la petición mediante BluetoothSocket(es la interfaz de bluetooth, similar a TCP socket).

BluetoothServerSocket representa el servidor abierto que escucha las peticiones en camino(similar a TCP serverSocket), en orden de conectividad 2 dispositivos Android, uno debe activar el servicio socket con esta clase.

Cuyos métodos accept() para aceptar conexión o close() para desactivar el servicio.

Activity Manager: Interactúa con todas las actividades en ejecución en el sistema, algunas clases importantes son: ActivityManager.AppTask, permite el gestionamiento de las tareas de nuestras aplicaciones.

ActivityManager.MemoryInfo información que recuperas de tu memoria.

ActivityManager.RunningTaskInfo muestra información de un task en particular ejecutándose en el sistema.

Algunos métodos:

addAppTask(Activity, Intent, Activity ...) añade una nueva ActivityManager.AppTask para llamar la aplicación.

getAppTasks() retorna la lista de tasks asociados a una aplicación.

GetLargeMemoryClass() Retorna la memoria por aplicación en el dispositivo cuando esta ejecutando un largo tiempo.

Content Provider: Gestiona el acceso a un conjunto estructurado de datos. Ellos encapsulan los datos, y proporcionan mecanismos para la definición de seguridad de los datos. Los Content Providers son la interfaz estándar que conecta los datos en un solo proceso con código que se ejecuta en otro proceso. Mediante el objeto ContentResolver en el contexto de aplicación para comunicar con el provider como un cliente, el ContentResolver comunica con el objeto Provider, una instancia de esa clase implementa ContentProvider, recibe data desde el cliente, ejecuta una acción y retorna un resultado. Método query() recupera data desde el provider, insert() inserta una nueva fila en tu provider, usa el argumento para seleccionar el destino en la tabla, onCreate() inicializa el provider, Android llama este método inmediatamente después de crear tu provider.

2. A partir del Android 5.0 se ha sustituido la Dalvik VM por otra máquina virtual. Busque el nombre de dicha máquina y describa sus mejoras respecto a Dalvik VM.

La nueva máquina virtual (como mencione en el apartado 1 que pedían describir mejoras y lanzamientos importantes por cada API) tiene de nombre como ART.

ART, Soporta y mejora la ejecución y velocidad en arquitecturas ARM, x86, MIPS, debido a que una vez que se genera java byte code por el compilador de java, se ejecuta en el runtime environment del dispositivo.

Dalvik esta basado en JIT(just in time) significa que cada vez que ejecutas tu aplicación la parte requerida para la ejecución sera traducida (compilada) a código máquina en ese momento. Y según tu sigas ejecutando la app, el código adicional sera compilado y guardado en memoria, entonces así el sistema puede reusar el código mientras la app esta en ejecución.

ART compila el lenguaje intermedio, Dalvik ByteCode, en el sistema binario dependiente. Todo el código de la app será pre-compilado durante la instalación (una vez), eliminando el lag que se ve cuando abrimos una app en el dispositivo. Sin necesidad de ejecutar JIT, se ejecuta mucho más rápido.

Referencias

https://en.wikipedia.org/wiki/Application_programming_interface
[http://www.htcmania.com/mediawiki/index.php/Programaci%C3%B3n de aplicaciones para m%C3%B3viles Android -
Unidad 1#Las versiones de Android y niveles de API](http://www.htcmania.com/mediawiki/index.php/Programaci%C3%B3n_de_aplicaciones_para_m%C3%B3viles_Android_-_Unidad_1#Las_versiones_de_Android_y_niveles_de_API)
[https://es.wikipedia.org/wiki/Anexo:Historial de versiones de Android](https://es.wikipedia.org/wiki/Anexo:Historial_de_versiones_de_Android)
<https://joefebrian.wordpress.com/2012/10/23/how-to-find-jdk-version-on-ubuntu-12-04/>
<http://developer.android.com/intl/es/sdk/installing/index.html?pkg=studio>
<http://developer.android.com/intl/es/tools/support-library/index.html>
<http://definicion.de/dispositivo/>
<https://infinum.co/the-capsized-eight/articles/art-vs-dalvik-introducing-the-new-android-runtime-in-kit-kat>
<http://developer.android.com/intl/es/guide/topics/providers/content-provider-creating.html#ContentProvider>