



TALLER DE APLICACIONES MÓVILES

SEMANA 2: ARQUITECTURA DE UN SISTEMA

OPERATIVO





ESCUELA DE INGENIERÍA Y CONSTRUCCION

Director: Marcelo Lucero

ELABORACIÓN

Experto disciplinar: Javier Ignacio Miles Avello

Diseño instruccional: Felipe Molina

VALIDACIÓN Experto disciplinar: Helmut

Aubel

Jefa de Diseño Instruccional: Alejandra San Juan

EQUIPO DE DESARROLLO

AIEP

AÑO

2023



Tabla de contenidos

Aprendizaje esperado de la semana.....	4
Introducción	4
1. Definición de arquitectura de sistema operativo Android	5
2. Kernel y controladores de dispositivos periféricos.....	7
3. Librerías OpenGL, FreeType, libc, Surface manager, webkit, sqlite, Media Framework.....	9
4. Android Runtime	11
5. Definición de Framework de aplicaciones.....	11
6. Definición de aplicaciones móviles, en relación con la arquitectura de Android	12
Conclusiones.....	14
Referencias.....	15



Aprendizaje esperado de la semana

Analizar componentes de sistema operativo Android, considerando arquitectura, middleware, lenguajes de programación y aplicaciones móviles.

Introducción

- ¿Qué es un sistema operativo?
- ¿Cuáles son los componentes del sistema?
- ¿Cuáles son las distintas capas de comunicación dentro del sistema?

Estas son interrogantes que dan continuidad a la lección de la semana anterior. En esta semana, podrás profundizar aún más en relación con los componentes internos de un sistema operativo, y cómo interactúan entre ellos, para dar funcionalidad al sistema, en completitud.

En este documento, veremos de qué se conforman los distintos elementos del sistema operativo Android, para poder conocer cómo se ejecutarán las aplicaciones que desarrollemos en un futuro próximo.



1. Definición de arquitectura de sistema operativo Android

Como mencionamos la semana anterior, un sistema operativo, es un conjunto de programas e instrucciones, que permite realizar las acciones básicas de un computador, para poder operarlas. Sin embargo, de acuerdo con (Real Academia Española), la definición de arquitectura, indicado para informática, como una “Estructura lógica y física de los componentes de una computadora.”

Entonces, uniendo ambos términos, la arquitectura de un sistema operativo, es la disposición de los distintos elementos e interacción de ellos, para que pueda operar un aparato electrónico.

En la siguiente ilustración, se especifica la pila de distintos componentes para el sistema operativo Android, el cual consta de: Kernel, Capa de Abstracción de Hardware, Librerías Nativas, Entorno de ejecución Android, Framework de Java, y por último, las aplicaciones.

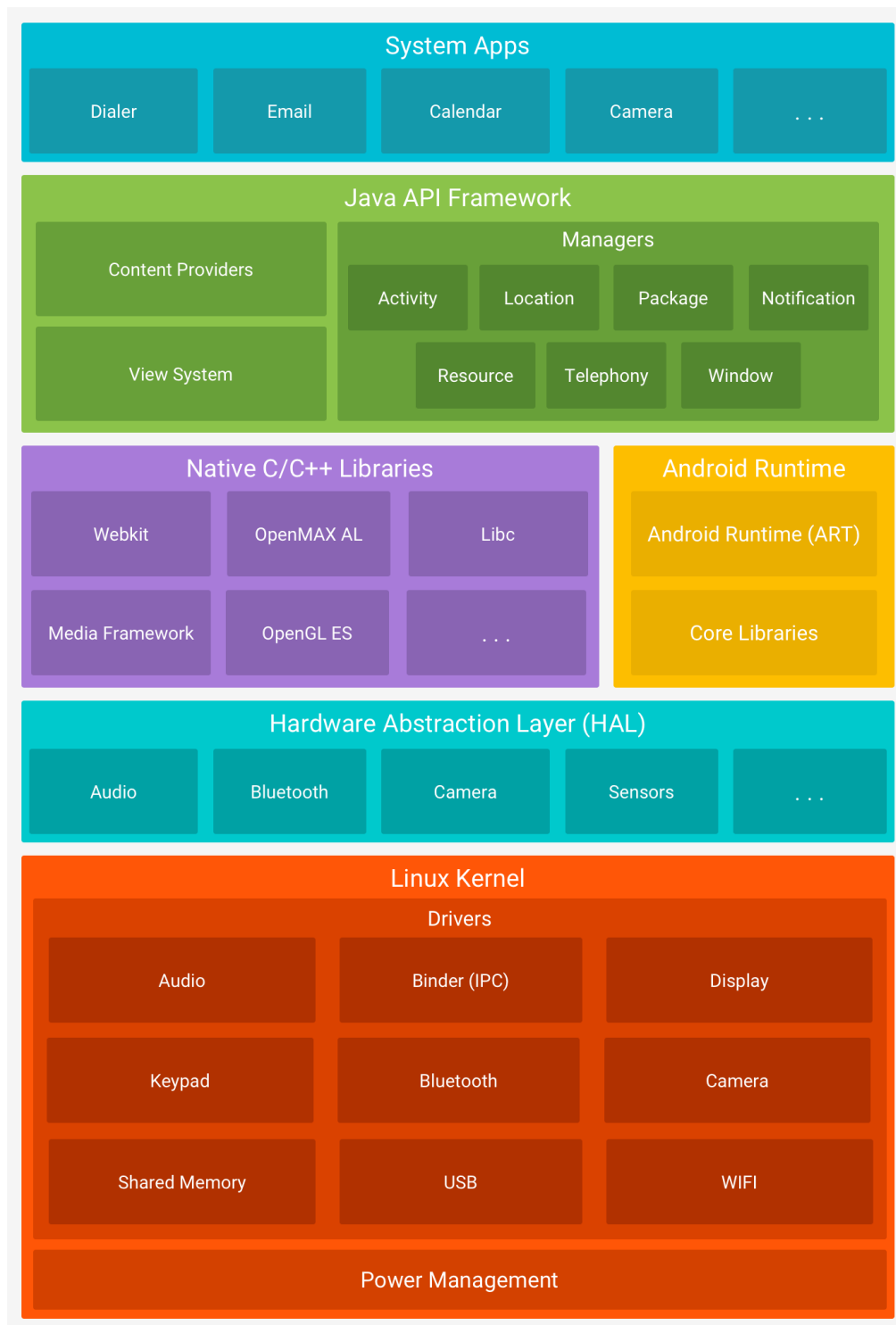


ILUSTRACIÓN 1: PILA DE SOFTWARE DE ANDROID

Fuente: developer.android.com

2. Kernel y controladores de dispositivos periféricos

En la lección anterior (semana 1), pudimos identificar los elementos como virtualización, kernel (en general como sistema operativo).

También, hablamos acerca de la virtualización, en donde se especifica como una capa intermedia entre hardware y software, el cual permite que el hardware, sea compartido por varios sistemas, y de esta manera aprovechar mejor los recursos.

Cada dispositivo, cuenta con distintos elementos electrónicos que lo componen. En la siguiente ilustración, demuestra los dispositivos con el cual cuenta un computador con sistema operativo Windows 10.

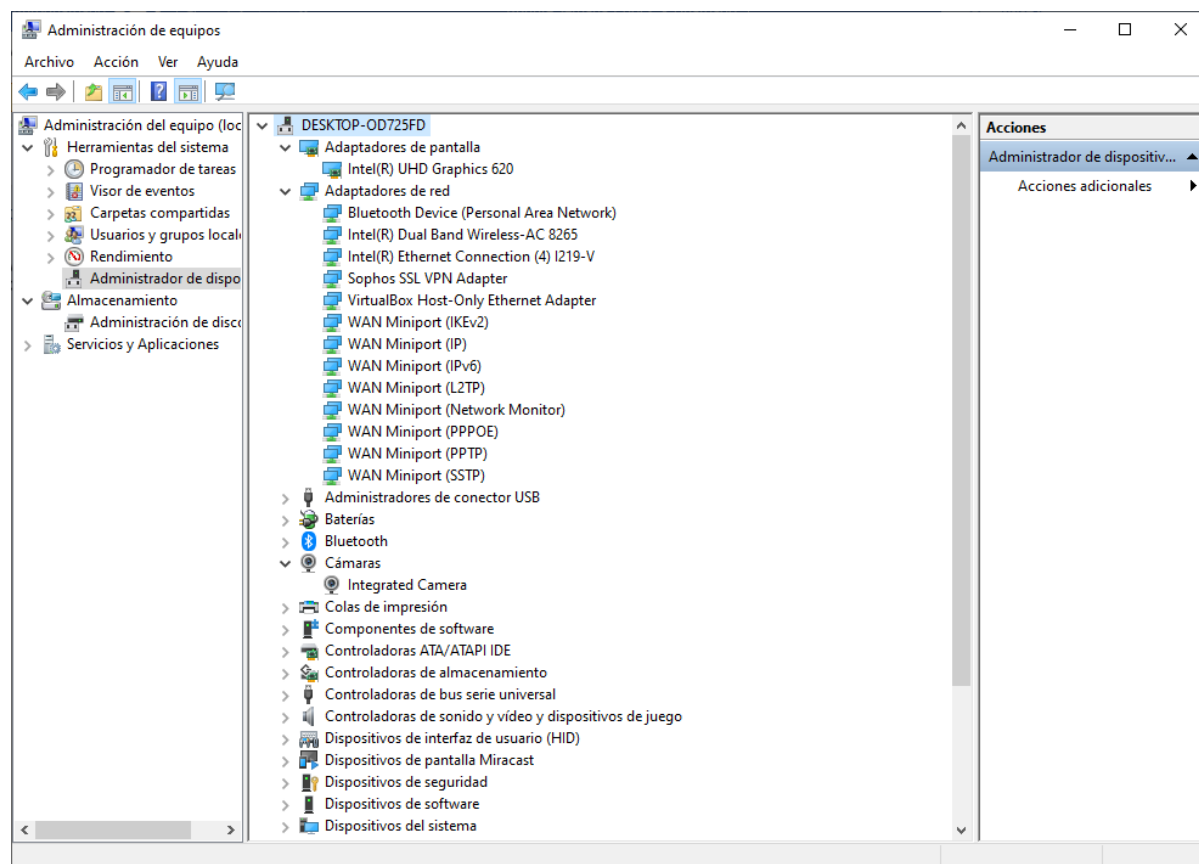


ILUSTRACIÓN 2: ADMINISTRADOR DE DISPOSITIVOS W10

Fuente: Elaboración Propia




Para que pueda operar un sistema, se requiere de dispositivos de entrada y salida, los cuales pueden ser: Teclado y Video, respectivamente. Lo anterior, para que un usuario pueda ingresar las instrucciones al computador, y luego ver los resultados en una pantalla.

En lo que se refiere a utilidad de uso, se han ido adicionando nuevos elementos a los computadores, como: Impresoras, cámaras, entrada táctil, audio, etc. Además, se han ido incorporando nuevos elementos, como sensores: Giroscopio, GPS, inclinómetro, lector de huella, sensor de temperatura, lectores de código (QR, PDF417 y EAN-13).

Pues bien, para cada uno de los elementos que componen un dispositivo electrónico, se van adicionando distintos componentes de distintos fabricantes. La electrónica de ellos, varía en las instrucciones que reciba, y es por ello que los drivers (ver Ilustración 1), son necesarios, para poder traducir las instrucciones del hardware hacia el software (y vice-versa), de manera que el uno se entienda con el otro.

“La capa de abstracción de hardware (HAL) brinda interfaces estándares que exponen las capacidades de hardware del dispositivo al marco de trabajo de la API de Java de nivel más alto. La HAL consiste en varios módulos de biblioteca y cada uno de estos implementa una interfaz para un tipo específico de componente de hardware, como el módulo de la cámara o de Bluetooth. Cuando el marco de trabajo de una API realiza una llamada para acceder a hardware del dispositivo, el sistema Android carga el módulo de biblioteca para el componente de hardware en cuestión.” (Google Inc.)



3. Librerías OpenGL, FreeType, libc, Surface manager, webkit, sqlite, Media Framework

Cada vez que desarrollamos un sistema, para que este pueda ser ejecutado, necesita de librerías para realizarlo. Por ejemplo, para poder ejecutar un archivo java, entonces necesitarías la máquina virtual para poder ejecutarlo. Si desarrollas una aplicación web con .NET Framework, entonces para ejecutar la aplicación, necesitarías el Framework Mismo. En definitiva, para la ejecución de ciertos programas, necesitas de otros componentes de software, llamados Librerías. Por ejemplo, si el lenguaje de desarrollo para una aplicación móvil requiere es C o C++, entonces podemos utilizar el NDK de Android para acceder a las bibliotecas de plataformas nativas directamente desde el código nativo.

“Muchos componentes y servicios centrales del sistema Android, como el ART y la HAL, se basan en código nativo que requiere bibliotecas nativas escritas en C y C++. La plataforma Android proporciona API del marco de trabajo de Java para exponer la funcionalidad de algunas de estas bibliotecas nativas a las apps. Por ejemplo, puedes acceder a OpenGL ES a través de la API de OpenGL de Java del marco de trabajo de Android para agregar a tu app compatibilidad con los dibujos y la manipulación de gráficos 2D y 3D.” (Google Inc.)

Algunas de las librerías, son:

- OpenGL (Open Graphics Library): Es un estándar que define una API multiplataforma y multilenguaje, utilizado para programar aplicaciones que generen gráficos 2D y 3D. Básicamente, contiene distintas funciones para dibujar objetos en 3D complejas utilizando puntos, líneas y triángulos (por ejemplo). Se usa principalmente en productos de software basados en



realidad virtual, realidad aumentada y distintos simuladores. También se utiliza para el desarrollo de videojuegos; Su equivalente es Direct3D en plataformas Microsoft Windows.

- FreeType: Es una biblioteca o librería, escrita en lenguaje de programación C, el cual implementa un motor para los diferentes tipos de letra de un software. Es el traductor entre las imágenes vectoriales de las tipografías (TrueType) en mapas de bits. Esta librería, es distribuida bajo licencias GNU y BSD.
- LibC: Es la biblioteca estándar de C, la cual es una recopilación de bibliotecas con ciertas rutinas estandarizadas, que permite realizar operaciones comunes (como entrada/salida), y manejo de cadena de caracteres.
- Surface Manager: Esta librería, es la encargada de administrar los diferentes elementos de navegación de pantalla. Esta Librería, además gestiona las ventanas de las distintas aplicaciones activas en el sistema.
- Webkit: Es una API, desarrollada en Objective-C y una de las características que posee, es permitir la interacción con un servidor web para obtener y renderizar páginas web, descarga de archivos, y administración de ciertos plugins.
- SQLite: Es mayormente conocido como un sistema de gestión de bases de datos relacional, y que está contenida en una biblioteca escrita en C.
- Media Framework: Esta API, permite la utilización de dispositivos multimedia.



4. Android Runtime


Es un entorno utilizado para la ejecución de aplicaciones en un sistema operativo móvil Android. ART(Android Runtime), reemplaza a la máquina virtual Dalvik, utilizada originalmente por Android, y permite la transformación de la aplicación en instrucciones de máquina, que luego son ejecutadas por el entorno de ejecución nativo del dispositivo.

“Para los dispositivos con Android 5.0 (nivel de API 21) o versiones posteriores, cada app ejecuta sus propios procesos con sus propias instancias del tiempo de ejecución de Android (ART). El ART está escrito para ejecutar varias máquinas virtuales en dispositivos de memoria baja ejecutando archivos DEX, un formato de código de bytes diseñado especialmente para Android y optimizado para ocupar un espacio de memoria mínimo. Crea cadenas de herramientas, como Jack, y compila fuentes de Java en código de bytes DEX que se pueden ejecutar en la plataforma Android.” (Google Inc.)

De acuerdo a (Google Inc.), indica las funciones principales del ART, las cuales son: compilación ahead-of-time (AOT), el cual crea un archivo de compilación posterior a la instalación de la aplicación y por otro lado, just-in-time (JIT) para compilar el código cada vez que se inicia una aplicación. Además, ART permite la recolección optimizada de elementos no utilizados (GC).

5. Definición de Framework de aplicaciones

En resumen, dado todo lo anterior, android está disponible mediante API escritas en lenguaje Java. Dichas API, son los elementos fundamentales para poder crear



aplicaciones en Android, de manera que se simplifica la reutilización de componentes del sistema, y los servicios asociados, dentro de los cuales se puede señalar:

- Un sistema de vista enriquecido y extensible que puedes usar para compilar la IU(Interfaz de Usuario) de una aplicación, en donde se incluyen listas, cuadrículas, cuadros de texto, botones e incluso un navegador web integrable.
- Un administrador de recursos que brinda acceso a recursos sin código, como strings localizadas, gráficos y archivos de diseño.
- Un administrador de notificaciones que permite que todas las aplicaciones muestren alertas personalizadas en la barra de estado.
- Un administrador de actividad que administra el ciclo de vida de las aplicaciones y proporciona una pila de retroceso de navegación común.
- Proveedores de contenido que permiten que las aplicaciones accedan a datos desde otras apps, como la app de Contactos, o compartan sus propios datos.

Fuente: (Google Inc.)

6. Definición de aplicaciones móviles, en relación con la arquitectura de Android

“En Android se incluye un conjunto de aplicaciones centrales para correo electrónico, mensajería SMS, calendarios, navegación en Internet y contactos, entre otros elementos. Las aplicaciones incluidas en la plataforma no tienen un estado especial entre las apps que el usuario elije instalar; por ello, una app externa se puede convertir en el navegador web, el sistema de mensajería SMS



o, incluso, el teclado predeterminado del usuario (existen algunas excepciones, como la app Settings del sistema)."

(Google Inc.)

Como mencionamos en la lección anterior, un sistema operativo es una Kernel, que provee de funciones básicas para operar un dispositivo. Sin embargo, si no existieran las aplicaciones que son las que dan solución a la problemática de los usuarios, el sistema operativo en sí, no tiene sentido.

"Las apps del sistema funcionan como apps para los usuarios y brindan capacidades claves a las cuales los desarrolladores pueden acceder desde sus propias apps. Por ejemplo, si en tu app se intenta entregar un mensaje SMS, no es necesario que compiles esa funcionalidad tú mismo; como alternativa, puedes invocar la app de SMS que ya está instalada para entregar un mensaje al receptor que especifiques." (Google Inc.)



Conclusiones

Muchos desarrolladores, han creado software específico para que sean utilizados por otros sistemas. Dichos softwares, se pueden entregar como API, o como librerías, dependiendo de la naturaleza por la cual fue creada.

Android se compone de diferentes softwares, que provee de muchas funcionalidades ya desarrolladas (no inventar la rueda de nuevo), y en su conjunto, generan un sistema completamente distinto.

Otros lenguajes como Python, utiliza librerías para poder generar nuevos sistemas, de una manera más simple, sin tener que desarrollar muchos módulos, que ya vienen listos para su uso.



Referencias

Google Inc. (s.f.). Android Developers. Obtenido de
<http://developer.android.com>

Real Academia Española. (s.f.). rae. Obtenido de Real Academia Española:
<http://www.rae.es>