Unidad 1 / Escenario 1

Lectura fundamental

Introducción al desarrollo de aplicaciones móviles en Android

Contenido

Palabras clave: Android, aplicación, arquitectura, lenguaje, librería, móvil.

1. Introducción al desarrollo de aplicaciones móviles

En la actualidad, el uso de los dispositivos móviles se ha vuelto cotidiano no solo en servicios de voz, sino también en servicios de datos. Todo esto ha sido posible porque las redes, los protocolos y los dispositivos han evolucionado hacia mayores velocidades y capacidades de almacenamiento y procesamiento. Así mismo, ha crecido el auge por la construcción de nuevos servicios de acuerdo con los requerimientos que demanda la vida cotidiana.

En la medida que exista una gama de dispositivos móviles con características de *software* y *hardware* que los diferencien, las arquitecturas y los lenguajes también van a variar, lo cual se constituye en un reto para el programador, quien debe estar en capacidad de encontrar soluciones utilizando tecnologías móviles.

En el momento de desarrollar aplicaciones móviles se debe tener en cuenta la capacidad del dispositivo, es decir, cuáles son sus características de *hardware*. Esto con el objeto de no sobrecargar el dispositivo durante el procesamiento de sus aplicaciones; además, la entrada de datos debe ser sencilla y amigable.

1.1. Tipos de aplicaciones móviles

En lo referente a las aplicaciones, no todas tienen las mismas características ni son del mismo tipo. A continuación, se describen las clases y sus características:

1.1.1. Aplicaciones móviles nativas

Son las que se desarrollan utilizando el lenguaje nativo del propio dispositivo, es decir, específicamente para cada sistema operativo (Android, iOS o Windows).

La siguiente tabla muestra el fabricante y lenguaje de programación propio para los sistemas operativos Android, iOS y Windows.

Tabla 1. Lenguaje nativo de sistemas operativos móviles

Sistema operativo	Fabricante	Lenguaje de programación
Android	Google	Java
iOS	Apple	Objetive C, Swift
Windows	Microsoft	C#, Visual Basic.NET

Fuente: elaboración propia

1.1.2. Aplicaciones móviles WEB

El desarrollo de estas aplicaciones es independiente del sistema operativo; por tanto, se pueden ejecutar en cualquier dispositivo o navegador y lo único que se requiere es un navegador para poder acceder a ellas.

Estas aplicaciones utilizan lenguajes como HTML, CCS y JavaScript. En el momento que se desee ejecutar una aplicación, esta se adapta al dispositivo correspondiente, sin que se necesite hacer ninguna instalación.

1.1.3. Aplicaciones móviles híbridas

Estas aplicaciones son una combinación de las aplicaciones nativas y las WEB; esto significa que además de utilizar los lenguajes empleados en las WEB (HTML, CCS y JavaScript), usa el lenguaje de programación propio de cada sistema operativo, lo cual permite el acceso a las funcionalidades del dispositivo.

1.2. Ventajas y desventajas de los tipos de aplicaciones

La Tabla 2 muestra algunas ventajas y desventajas de cada uno de los tipos de aplicaciones móviles.

Tabla 2. Ventajas y desventajas de los diferentes tipos de aplicaciones móviles

Tipo de aplicaciones	Ventajas	Desventajas
Nativas	• Se tiene acceso a las funcionalidades del dispositivo. • Disponible en App Store. • Actualización constante. • Mayor rendimiento. • Mayor velocidad.	• El código de la aplicación no es reutilizable para las otras plataformas. • Su desarrollo tiende a ser más costoso. • Se deben conocer diferentes lenguajes y herramientas para cada plataforma.
WEB	• El mismo código se puede utilizar en diferentes plataformas. • El desarrollo de las aplicaciones es más sencillo y menos costoso con respecto a las aplicaciones nativas. • No necesita instalación.	• Requiere conexión a internet. • El acceso a los elementos del *hardware* no es óptimo. • No está disponible en las tiendas APP Store.
Híbridas	• Es posible distribuirlas desde las tiendas App Store. • El código de la aplicación puede ser utilizado en diferentes plataformas. • Se puede acceder a partes del *hardware* del dispositivo.	• No cuenta con todas las funcionalidades de las aplicaciones nativas.

Fuente: elaboración propia

2. Introducción al sistema operativo Android

Android es un sistema operativo desarrollado por Google para dispositivos como celulares, tabletas, televisores y otros. Está basado en Linux con un *software* intermedio que interactúa entre las aplicaciones y el sistema operativo.

Android es de código abierto libre por lo que cualquier desarrollador que tenga conocimientos de Java puede crear fácilmente una gran cantidad de aplicaciones, ya sea para comercializar, ofrecerlas gratuitamente o para adaptarlas a una gran variedad de dispositivos de diferentes fabricantes.

2.1. Versiones disponibles en Android

Debido al surgimiento de nuevas versiones, es necesaria la creación de otras API (*Application Programming Interface*) compatibles con las versiones anteriores, en las que hay que corregir algunos fallos, si los hubiera, y adicionar funcionalidades novedosas. Las versiones de Android representan los nombres de diferentes postres en inglés; para cada una de las versiones el nombre del postre elegido empieza con una letra distinta, siguiendo el orden alfabético. La siguiente tabla muestra: el nombre, la versión, la API que incluye y la fecha de creación de cada una.

Tabla 3. Versiones Android

Nombre	Versión	Nivel de API	Fecha
A: *Apple pie* (tarta de manzana)	1.0	1	Septiembre 23 de 2008
B: *Banana bread* (pan de banano)	1.1	2	Febrero 9 de 2009
C: *Cupcake* (magdalena glaseada)	1.5	3	Abril 30 de 2009
D: *Donut* (rosquilla)	1.6	4	Septiembre 15 de 2009
E: *Éclair* (pastel francés, conocido en España como pepito)	2.0/2.1	5-7	Octubre 26 de 2009

Nombre	Versión	Nivel de API	Fecha
F: *Froyo* (abreviatura de frozen yogurt o yogur helado)	2.2	8	Mayo 20 de 2010
G: *Gingerbread* (pan de jengibre)	2.3	9-10	Diciembre 6 de 2010
H: *Honeycomb* (panal)	3.0/3.1/3.2	11-13	Febrero 22 de 2011
I: *Ice cream sandwich* (emparedado de helado)	4.0	14-15	Septiembre 15 de 2009
J: *Jelly bean* (gomitas de gelatina)	4.1/4.2/4.3	16-18	Julio 9 de 2012
K: *Kit Kat* (chocolatina)	4.4	19-20	Octubre 31 de 2013
L: *Lollipop* (piruleta)	5.0/5.1	21-22	Noviembre de 2014
M: *Marshmallow* (esponja o masmelo)	6.0	23	Octubre de 2015
N: *Nougat* (turrón)	7.0/7.1	24-25	Julio de 2016
O: *Oreo* (galleta Oreo)	8.0/8.1	26-27	Agosto de 2017

Fuente: elaboración propia

3. Arquitectura de Android

La arquitectura Android tiene en cuenta los recursos disponibles y el consumo de sus aplicaciones instaladas; por tanto, las aplicaciones Android se ejecutan con restricciones de disponibilidad de memoria, consumo de batería, capacidad de almacenamiento, formas de visualización, etc.

Con base en la arquitectura Android, todo desarrollador debe tener en cuenta los siguientes aspectos (Benbourahla, 2015):

A. La creación de nuevos objetos, es decir, elementos con sus atributos y funcionalidades.

B. El uso de recursos como el procesador, la memoria RAM, el almacenamiento interno, entre otros.

C. El consumo de la batería.

D. Los diferentes tamaños y resoluciones de pantalla.

E. Las diversas versiones de Android que se encuentran disponibles.

3.1. Descripción arquitectura Android

La arquitectura Android está conformada por cuatro capas, las cuales de describen a continuación:

3.1.1. Applications

Esta capa contiene tanto las aplicaciones propias del sistema operativo Android (como contactos, programa SMS, mapas, navegador, calendario etc.) como aquellas que instalen los programadores.

3.1.2. Application framework

Concede a los desarrolladores tener acceso a las mismas API del *framework* para crear aplicaciones utilizando vistas, servicios, actividades, etc. Esto permite la simplicidad y reutilización de componentes basados en las bibliotecas y en el entorno de ejecución.

Los servicios más importantes que usan las aplicaciones son:

a. *Activity Manager:* serie de las APIs que se encargan de manejar el ciclo de vida de las actividades (pantallas) y el conjunto de estas que conforman las aplicaciones en Android; también proporcionan un sistema de navegación entre actividades o ventanas.

b. *Window Manager:* gestiona las ventanas de las aplicaciones; utiliza la librería *Surface Manager*.

c. Telephone Manager: concede el acceso a la telefonía del dispositivo Android; realiza llamadas o envía y recibe mensajes.

d. *Content Provider*: permite encapsular un conjunto de datos en cualquier aplicación para compartirlos con otras aplicaciones Android. Por ejemplo, gracias a este servicio los datos de contactos, agenda, mensajes, entre otros, pueden ser utilizados por otras aplicaciones.

e. *View System:* provee un conjunto de controles que se pueden incluir dentro de las ventanas. Estos controles o elementos proporcionan unas herramientas para construir interfaces de usuario (GUI), como listas, botones, cuadros de texto, etc.

f. *Location Manager*: permite, en las aplicaciones, obtener información sobre la localización geográfica del dispositivo, utilizando el GPS o la red, y trabajar con los mapas.

g. *Notification Manager*: posibilita que todas las aplicaciones le muestren al usuario alertas de eventos que ocurran durante su ejecución en la barra de estado; por ejemplo, una llamada entrante, un mensaje recibido, conexión Wi-Fi disponible, etc.

h. XMPP *Service:* por medio de estas funciones se puede realizar intercambio de mensajes con base en el lenguaje XML.

i. *Sensor Manager:* permite utilizar los sensores de *hardware* del dispositivo, como el acelerómetro, el giroscopio, el sensor de luminosidad, la brújula, el sensor de presión, el sensor de proximidad, el sensor de temperatura, etc.

j. Cámara: proporciona acceso a las cámaras del dispositivo Android, tanto para tomar fotografías como para grabar video.

k. Multimedia: por medio de este conjunto de bibliotecas se pueden reproducir y visualizar audios, videos e imágenes en el dispositivo.

3.1.3. Libraries

Esta capa constituye un conjunto de librerías utilizadas por Android, que suministran una gran funcionalidad a cada una de las características del sistema operativo y son utilizadas por el desarrollador a través del *framework*.

Entre las librerías más importantes ubicadas en esta capa se pueden encontrar las siguientes:

a. libc: proporciona las cabeceras y funciones del lenguaje C optimizadas, permitiendo la funcionalidad básica para la ejecución de las aplicaciones.

b. *Surface Manager:* es la encargada de componer las diferentes imágenes que se muestran en pantalla, lo cual permite realizar fácilmente diversos efectos, como superposición de elementos, animaciones, transiciones, transparencias, entre otros.

c. OpenGL/SL y SGL: estas librerías tienen que ver con la parte gráfica de Android. OpenGL/SL maneja gráficos en 3D utilizando la aceleración del propio dispositivo móvil o un *software* optimizado cuando el *hardware* del dispositivo no posee aceleración.
Por otro lado, SGL maneja gráficos en 2D, lo que significa que es el motor gráfico 2D de Android y permite el desarrollo de aplicaciones combinando gráficos en 3D y 2D.

d. *Media framework:* suministra los códecs (JPG, GIF, PNG, MPEG4, AVC, MP3, etc.) necesarios para el manejo de la multimedia en numerosos formatos de imagen, audio y video.

e. FreeType: proporciona una serie de fuentes tipográficas, que permiten trabajar de forma rápida y sencilla con distintos tipos de fuentes.

f. SSL: es una librería que provee seguridad cuando se accede a internet, es decir, es un protocolo que ofrece cifrado o criptografía.

g. SQLite: permite la creación y administración de bases de datos relacionales, disponibles para las aplicaciones.

h. Librería WebKit: representa el motor para las aplicaciones de tipo navegador y forma el núcleo del navegador por defecto incluido en el sistema operativo Android.

3.1.4. Android Runtime

Al mismo nivel de las librerías de Android se encuentra el entorno de ejecución. Este está conformado por un conjunto de librerías que suministran una gran parte de las funcionalidades disponibles en las librerías del lenguaje Java. Dada las limitaciones, tanto de memoria como de procesador, no se utiliza la máquina virtual de Java estándar, sino que se crea una nueva máquina virtual, llamada Dalvik.

Para las últimas versiones, de la 5.0 en adelante, se crea la máquina virtual ART (Android RunTime), la cual reemplaza a la Dalvik, disponible en versiones anteriores a la 5.0. La ART consta de las siguientes características (Benbourahla, 2015):

a. AOT (*Compilation ahead of time* – compilación anticipada), que permite compilar las aplicaciones en tiempo de instalación y no en tiempo de ejecución, como lo realizaba la máquina virtual Dalvik. Esto hace que el rendimiento de las aplicaciones mejore sustancialmente, ya que no se compila con cada ejecución de la aplicación.

b. Mejora del *garbage collector*, es decir, el limpiador o recolector de basura.

c. Mejora en el desarrollo y depuración de aplicaciones; cuando se presenta una falla en el desarrollo de las aplicaciones aquí se presentan mensajes más detallados.

De lo anterior, se puede decir que la máquina virtual Dalvik (utilizada en versiones anteriores) realiza la compilación en tiempo de ejecución (*just in time*), lo cual hace que las aplicaciones se compilen cada vez que se ejecutan. En cambio, en la máquina virtual ART se compilan las aplicaciones en tiempo de instalación, lo cual aumenta el rendimiento, aunque requiere más espacio, dado que las almacena.

3.1.5. Linux Kernel

Esta capa utiliza Linux para los servicios del sistema como la gestión de procesos, la gestión de memoria, la seguridad y la red, entre otros.

Esta capa se encarga de los controladores o *drivers* del *hardware*, de tal forma que cualquier componente *hardware* pueda ser utilizado por medio del llamado correspondiente a este. Cada vez que un fabricante desarrolle un nuevo elemento *hardware*, este debe crear las respectivas librerías de control o *drivers* dentro del Kernel de Linux embebido en el sistema operativo Android.

Referencias

Arias, A. (2016). *Curso de Programación de Apps. Android y iPhone.* Editorial: IT Campus Academy.

Benbourahla, N. (2015). *Android 5. Principios del desarrollo de aplicaciones Java. Barcelona*, España: Eni.

Clodoaldo, R. y Robledo, D. (2013). *Programación en Android*. Madrid, España: Ministerio de Educación, Cultura y Deporte.

Gironés, J. (2013). *El gran libro de Android*. Barcelona, España. Editorial: Marcombo.

INFORMACIÓN TÉCNICA

FACULTAD DE
INGENIERÍA, DISEÑO
E INNOVACIÓN

Módulo: Herramientas de Programación Móvil I

Unidad 1: Conceptos básicos de Android y estructura de programación

Escenario 1: Arquitectura Android

Autor: Manuel Báez

Asesor Pedagógico: Angie Laitón

Diseñador Gráfico: Andrés Felipe Figueroa

Asistente: Alejandra Morales