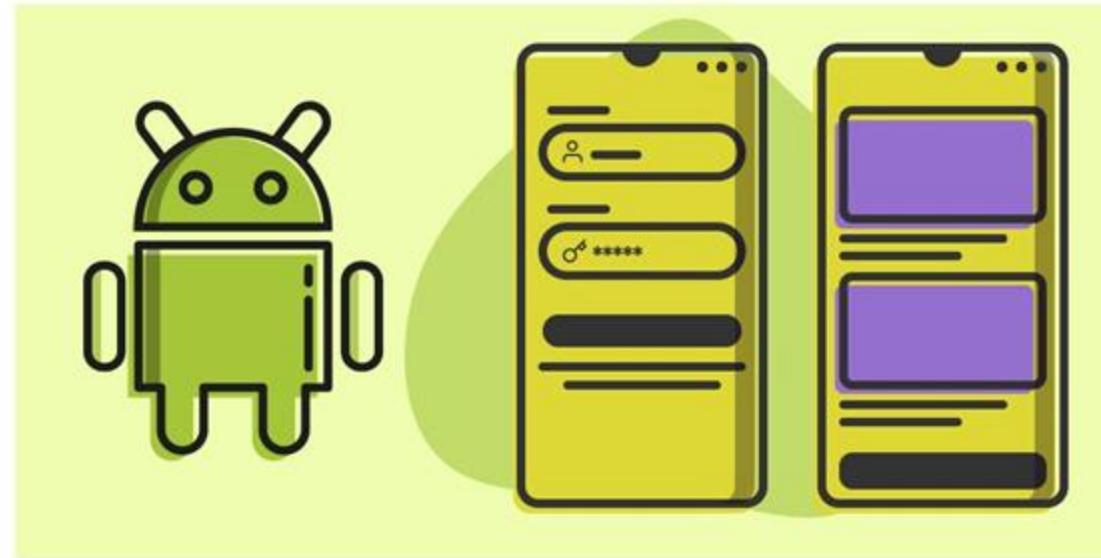


FACULTAD DE INGENIERÍA ESCUELA DE COMPUTACIÓN



Desarrollo de Software para Móviles Fundamentos de una aplicación móvil (APP)



¿Qué es Android?

Android es un sistema operativo al igual que Windows, Linux, etc. Por lo tanto, tiene el control total del dispositivo que lo contiene, así que cuando desarrollamos una aplicación, estamos desarrollando para el sistema operativo y podemos tener control de ciertos elementos que generar una experiencia más interesante y agradable para el usuario si lo hacemos de buena forma, en caso contrario podríamos estar generando una de las peores experiencias como usuario y afectando el rendimiento del dispositivo.

No obstante hoy en día, puede encontrarse en multitud de dispositivos móviles tales como relojes, televisiones, automóviles entre otros.

Características de Android

Las principales características a destacar son las siguientes:

- Se adapta a diferentes pantallas
- Navegador incluido soporta HTML5
- Soporta mensajes SMS y MMS
- Posee emulador para realizar desarrollo
- Soporte para pantallas multitáctil
- Realiza la conectividad mediante 3G, 4G, Bluetooth, WIFI.
- Es multitarea

Ventajas de Android

Entre las ventajas de Android podemos comentar las siguientes:

- Código abierto y por tanto facilidad para adaptarse a multitud de dispositivos además de los teléfonos móviles.
- Facilidad para instalar aplicaciones.
- Sistemas multitarea.
- Amplia comunidad de desarrolladores.
- Gama de aplicaciones muy extensa.
- Google Maps proporciona un GPS gratuito.

Desventajas de Android

Como desventajas, mencionaremos las siguientes:

- El ser multitarea el consumo de batería es importante.
- Es posible que las aplicaciones queden abiertas.

Versiones de Android

Una de las peculiaridades de las versiones de Android es que, además de tener nombres de postres en Inglés, dichos nombres van usando las letras del alfabeto de forma que cada versión utiliza la siguiente letra del alfabeto incrementando cada vez una. Es decir, empieza con la A, continúa con la B, C y así sucesivamente.

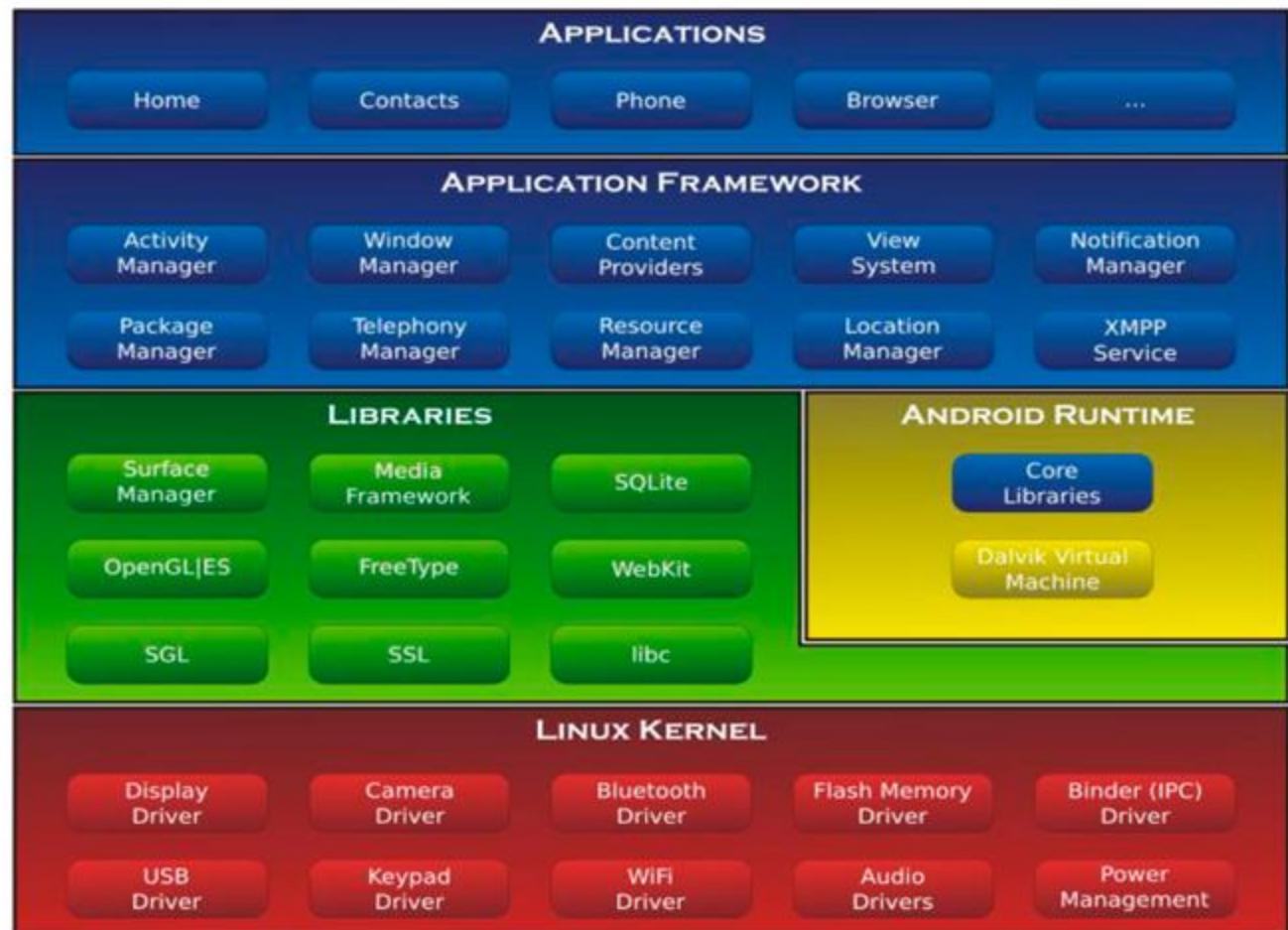
Versión	Nombre	Fecha aparición
1.0	Apple Pie	Septiembre 2008
1.1	Banana Bread	Febrero 2009
1.5	Cup Cake	Abril 2009
1.6	Donut	Septiembre 2009
2.0	Eclair	Octubre 2009
2.2	Froyo	Mayo 2010

Versiones de Android

Versión	Nombre	Fecha aparición
2.3	Gingerbread	Diciembre 2010
3.0	Honeycomb	Febrero 2011
4.0	Ice Cream Sandwich	Octubre 2011
4.1	Jelly Bean	Julio 2012
4.4	KitKat	Octubre 2013
5.0	Lollipop	Noviembre 2014
6.0	Marshmallow	Octubre 2015
7.0	Nougat	Junio 2016
8.0	Oreo	Agosto 2017
9.0	Pie	Agosto 2018
10.0	Android 10 (Quince Tart)	Septiembre 2019
11.0	Android 11 (Red Velvet Cake)	Septiembre 2020
12.0	Android 12 (Snow Cone)	Octubre 2021

Arquitectura de Android

Android está compuesto por una pila de software que contiene el sistema operativo, un middleware y las aplicaciones básicas para el usuario. Concretamente, las capas que lo componen son las que aparecen en la figura:



Arquitectura de Android

- **Aplicaciones:** cualquier tipo de aplicación escrita en Java.
- **Framework de las aplicaciones:** Acceso al API para rehusar componentes o modificarlos.
- **Bibliotecas en C/C++:** el desarrollador puede usarlas a través del Framework.
- **Runtime de Android:** bibliotecas del lenguaje Java y única instancia en la máquina virtual Dalvik.
- **Núcleo Linux:** Capa de abstracción del hardware y servicios de seguridad, gestión de memoria, de procesos, pila de red, modelo de los controladores, etc.

Fundamentos de una aplicación móvil

Para escribir aplicaciones de Android, es posible usar los lenguajes Kotlin, Java y C++. Las herramientas de Android SDK compilan tu código, junto con los archivos de recursos y datos, en un APK: *un paquete de Android, que es un archivo de almacenamiento con el sufijo .apk.*

Un archivo APK incluye todos los contenidos de una aplicación de Android y es el archivo que usan los dispositivos con tecnología Android para instalar la aplicación.

Cada aplicación de Android reside en su propia zona de pruebas de seguridad y está protegida por las siguientes características de seguridad de Android:

- El sistema operativo Android es un sistema Linux multusuario en el que cada aplicación es un usuario diferente.

Fundamentos de una aplicación móvil

- De forma predeterminada, el sistema le asigna a cada aplicación un ID de usuario de Linux único (solo el sistema utiliza el ID y la aplicación lo desconoce). El sistema establece permisos para todos los archivos en una aplicación de modo que solo el ID de usuario asignado a esa aplicación pueda acceder a ellos.
- Cada proceso tiene su propia máquina virtual (VM), por lo que el código de una aplicación se ejecuta de forma independiente de otras aplicaciones.
- De forma predeterminada, cada aplicación ejecuta su propio proceso de Linux. El sistema Android inicia el proceso cuando se requiere la ejecución de alguno de los componentes de la aplicación y, luego, lo cierra cuando el proceso ya no es necesario o cuando el sistema debe recuperar memoria para otras aplicaciones.

Fundamentos de una aplicación móvil

El sistema Android implementa el principio de mínimo privilegio. Es decir, de forma predeterminada, cada aplicación tiene acceso solo a los componentes que necesita para llevar a cabo su trabajo y nada más. Esto crea un entorno muy seguro, en el que una aplicación no puede acceder a partes del sistema para las que no tiene permiso.

Sin embargo, hay maneras en las que una aplicación puede compartir datos con otras aplicaciones y en las que una aplicación puede acceder a servicios del sistema:

- Es **possible coordinar que dos aplicaciones compartan el mismo ID de usuario** de Linux para que puedan acceder a los archivos de la otra. Para conservar recursos del sistema, **las aplicaciones con el mismo ID de usuario también pueden coordinar la ejecución en el mismo proceso de Linux y compartir la misma VM**. Las aplicaciones también deben estar firmadas con el mismo certificado.

Fundamentos de una aplicación móvil

- Una *aplicación puede solicitar permiso para acceder a datos del dispositivo*, como los contactos de un usuario, los mensajes de texto, el dispositivo de almacenamiento (tarjeta SD), la cámara y la conexión Bluetooth. *El usuario debe conceder de manera explícita estos permisos.*

Componentes de la aplicación

Los componentes de la aplicación son bloques de creación esenciales de una aplicación para Android. Cada componente es un punto de entrada por el que el sistema o un usuario ingresan a tu aplicación. Algunos componentes dependen de otros.

Las aplicaciones tienen cuatro tipos de componentes diferentes:

- Actividades
- Servicios
- Receptores de emisiones
- Proveedores de contenido

Cada tipo tiene un fin específico y un ciclo de vida característico que define cómo se crea y se destruye el componente.



Componentes de la aplicación - Actividades

Una actividad es el punto de entrada de interacción con el usuario. Representa una pantalla individual con una interfaz de usuario.

Por ejemplo, *una aplicación de correo electrónico* tiene una actividad que muestra una lista de los correos electrónicos nuevos, otra actividad para redactar el correo electrónico y otra actividad para leerlo.

Si bien las actividades funcionan juntas para ofrecer una experiencia de usuario uniforme en la aplicación de correo electrónico, cada una es independiente de las demás.

De esta manera, una aplicación diferente puede iniciar cualquiera de estas actividades (si la aplicación de correo electrónico lo permite).

Nota: Las actividades se implementan como subclases de la clase Activity



Componentes de la aplicación - Servicios

Un servicio es un punto de entrada general que permite mantener la ejecución de una aplicación en segundo plano por diversos motivos. Es un componente que se ejecuta en segundo plano para realizar operaciones de ejecución prolongada o para realizar tareas de procesos remotos. Un servicio no proporciona una interfaz de usuario.

Por ejemplo, un servicio podría reproducir música en segundo plano mientras el usuario se encuentra en otra aplicación, o podría capturar datos en la red sin bloquear la interacción del usuario con una actividad. Otro componente, como una actividad, puede iniciar el servicio y permitir que se ejecute o enlazarse a él para interactuar.

Nota: Un servicio se implementa como una subclase de Service.

Componentes de la aplicación - Servicios

Servicios iniciados

Los servicios iniciados le indican al sistema que los siga ejecutando hasta que finalicen su trabajo. Dos ejemplos serían la sincronización de datos en segundo plano o la reproducción de música después de que el usuario sale de la aplicación. La sincronización de datos en segundo plano o la reproducción de música también son dos tipos de servicios iniciados muy diferentes que modifican la manera en que el sistema los administra.

Componentes de la aplicación - Servicios

Servicios enlazados

Los servicios enlazados se ejecutan porque otra aplicación (o el sistema) indicó que quiere usarlos. Básicamente, lo que sucede es que un servicio le brinda una API a otro proceso.

Por lo tanto, el sistema sabe que hay una dependencia entre estos procesos. En consecuencia, si el proceso A está enlazado a un servicio en el proceso B, sabe que debe mantener funcionando el proceso B (y el servicio correspondiente) para el proceso A.

Además, si el proceso A es de interés para el usuario, también sabe que debe tratar el proceso B teniendo esto en cuenta.



Componentes de la aplicación - Receptores de emisiones

Un receptor de emisión es un componente que posibilita que el sistema entregue eventos a la aplicación fuera de un flujo de usuarios habitual, lo que permite que la aplicación responda a los anuncios de emisión de todo el sistema.

Dado que los receptores de emisión son otro punto bien definido de entrada a la aplicación, el sistema puede entregar emisiones incluso a las aplicaciones que no estén en ejecución.

Por ejemplo, una aplicación puede programar una alarma para publicar una notificación sobre un evento futuro destinada al usuario. Al entregar dicha alarma al receptor de emisión de la aplicación, no hace falta que dicha aplicación siga ejecutándose hasta que se active la alarma.

Componentes de la aplicación - Receptores de emisiones

Las aplicaciones también pueden iniciar emisiones, por ejemplo, para avisarles a las demás aplicaciones que se descargaron datos al dispositivo y que están disponibles para el uso.

Si bien los receptores de emisión no exhiben una interfaz de usuario, pueden crear una notificación de la barra de estado para alertar al usuario cuando se produzca un evento de emisión. Sin embargo, por lo general, un receptor de emisión es simplemente una puerta de enlace a otros componentes y está destinado a realizar una cantidad mínima de trabajo. Por ejemplo, podría programar un servicio ***JobService*** para que realice algunas tareas en función del evento con ***JobScheduler***.

Nota: Un receptor de emisión se implementa como una subclase de `BroadcastReceiver` y cada receptor de emisión se entrega como un objeto `Intent`.



Componentes de la aplicación - Proveedores de contenido

Un proveedor de contenido administra un conjunto compartido de datos de la aplicación que puedes almacenar en el sistema de archivos, en una base de datos SQLite, en la Web o en cualquier otra ubicación de almacenamiento persistente a la que tenga acceso tu aplicación.

A través del proveedor de contenido, otras aplicaciones pueden consultar o modificar los datos si el proveedor de contenido lo permite.

Por ejemplo, el sistema Android proporciona un proveedor de contenido que administra la información de contacto del usuario. De esta manera, cualquier aplicación con los permisos correspondientes puede consultar el proveedor de contenido (como `ContactsContract.Data`) para la lectura y la escritura de información sobre una persona específica.

El archivo de manifiesto (AndroidManifest.xml)

Para que el sistema Android pueda iniciar un componente de la aplicación, debe reconocer la existencia de ese componente leyendo el archivo de manifiesto de la aplicación (AndroidManifest.xml).

La aplicación debe declarar todos sus componentes en este archivo, que debe encontrarse en la raíz del directorio de proyectos de la aplicación.

El manifiesto puede hacer ciertas cosas además de declarar los componentes de la aplicación, por ejemplo:

- Identificar los permisos de usuario que requiere la aplicación, como acceso a Internet o acceso de lectura para los contactos del usuario.

El archivo de manifiesto (AndroidManifest.xml)

- Declarar el nivel de API mínimo que requiere la aplicación en función de las API que usa.
- Declarar características de hardware y software que la aplicación usa o exige, como una cámara, servicios de Bluetooth o una pantalla multitáctil.
- Declarar bibliotecas de la API a las que la aplicación necesita estar vinculada (además de las API del marco de trabajo de Android), como la biblioteca de Google Maps.

¿Preguntas?

