



INTRODUCCIÓN A ANDROID

PROGRAMACIÓN MOVILES



22 DE SEPTIEMBRE DE 2023

CARLOS MORAIS BLANCO

Contenido

Introducción.....	3
Android como Sistema Operativo	4
1. Historia.....	4
2. Cuota de mercado.....	6
3. Ventajas e Inconvenientes	7
Ventajas.....	7
Inconvenientes.....	8
4. Convivencia con otros sistemas operativos.....	9
Android como Framework.....	12
1. Android SDK	12
2. Android Studio.....	13
Estructura de Android.....	15
1. Kernel	16
2. Capa de abstracción de hardware.....	16
3. Android Runtime	16
4. Bibliotecas C/C++ nativas	16
5. Framework del API de Java	17
6. Apps del sistema.....	17

Introducción

El sistema operativo Android, desarrollado por Google, ha desempeñado un papel transformador en la forma en que interactuamos con la tecnología en la era moderna. Su historia, estructura interna y el entorno de desarrollo asociado son componentes esenciales para comprender su impacto en nuestra vida cotidiana.

En este trabajo, emprendaremos un viaje de descubrimiento en tres etapas fundamentales. En primer lugar, exploraremos la historia de Android, desde sus orígenes modestos hasta su consolidación como el sistema operativo líder en dispositivos móviles a nivel mundial.

Luego, nos adentraremos en la estructura interna del sistema operativo Android. Examinaremos sus componentes esenciales, como el kernel de Linux, el administrador de aplicaciones, la máquina virtual Dalvik (y más tarde, ART), así como los servicios fundamentales que permiten que las aplicaciones funcionen de manera eficiente en una variedad de dispositivos. Comprender esta arquitectura subyacente es esencial para apreciar la versatilidad y la estabilidad que Android ofrece.

Finalmente, nos sumergiremos en el mundo del desarrollo de aplicaciones Android, destacando dos pilares clave: Android Studio y Android SDK. Android Studio, el entorno de desarrollo integrado oficial, proporciona herramientas poderosas para la creación de aplicaciones. Por otro lado, el Android SDK ofrece una gama completa de recursos y bibliotecas que los desarrolladores utilizan para dar vida a sus ideas.

A medida que avanzamos en esta investigación, descubriremos cómo Android se ha convertido en una plataforma omnipresente y altamente personalizable que abarca una amplia variedad de dispositivos, desde teléfonos inteligentes y tabletas hasta relojes inteligentes, televisores y más. A través de este análisis exhaustivo, obtendremos una comprensión más profunda del funcionamiento del sistema Android y su influencia en nuestra vida digital actual.

Android como Sistema Operativo

1. Historia

Android fue creado por la empresa Android Inc. que fue fundada en 2003 por Andy Rubin, Rich Miner, Nick Sears y Chris White con el objetivo de desarrollar dispositivos móviles que estén al corriente de la ubicación y de las preferencias del usuario.

En un principio la intención era desarrollar un sistema operativo avanzado para cámaras digitales, pero más tarde al estudiar el mercado de las cámaras digitales y ver que no era lo bastante grande, se cambió el foco hacia el objetivo de crear un sistema que pudiera competir con Symbian y Windows Mobile.

En 2005, Google compró Android Inc. y empezó el desarrollo de una plataforma móvil basada en el kernel de Linux. La idea era crear un sistema parecido al de BlackBerry, basado en teclado QWERTY. Entonces llegó el iPhone y si el móvil de Apple tenía pantalla táctil, el de Google no podía quedarse atrás.

Se cambiaron las especificaciones para incluir el uso de la pantalla táctil que sería complementada por botones físicos. El 23 de septiembre de 2008 se lanzaba el primer teléfono con Android, el HTC Dream.

La primera versión de Android se hizo pública el 23 de septiembre de 2008, era la versión Android 1.0 que tenía ya algunos elementos que siguen prevaleciendo en Android hoy en día pero menos desarrollados.



Después de esta versión, se han ido mejorando y añadiendo funcionalidades a lo largo de los años como puede ser el teclado táctil en Android 1.5 Cupcake, Google maps en Android 2.0, el soporte para NFC en Android 2.3, y demás actualizaciones que mejoraron la interfaz y funcionalidad de Android, hasta llegar a la última versión de Android, Android 13.



2. Cuota de mercado

Según las cifras de marzo de 2023, Android mantiene en la actualidad su posición como sistema operativo móvil líder a nivel mundial, con una cuota de mercado del 71%, mientras que iOS representa el 28%.

Esto varía según los países, ya que en países como EEUU iOS sigue predominando sobre Android.



Los motivos por los que Android sigue siendo la cabeza en el ámbito de los sistemas operativos móviles puede darse por varios motivos; como puede ser el precio, ya que Android es mucho más asequible que iOS, por lo que, en países con un índice de pobreza mayor, Android será más viable que IOS.

3. Ventajas e Inconvenientes

Ventajas

Código abierto

Cualquiera puede crear aplicaciones y contribuir a aumentar la oferta de estas en el universo Android, siendo la gran mayoría de ellas, gratuitas.

El hecho de que su código sea abierto, también posibilita una gran variedad de ventajas más, como por ejemplo el hecho de que los errores puedan ser revisados y reparados con mayor rapidez.

Permite una gran personalización de la interfaz

Android admite el cambio del lanzador de aplicaciones predeterminado por otro. Pero no es demasiado complicado cambiar el aspecto de la pantalla, los iconos y mucho más.

Posee una multitarea real

La multitarea de Android siempre ha sido muy potente. De esta manera, las tareas en segundo plano no se interrumpen pasado cierto tiempo. Esto permite llevar a cabo trabajos avanzados mientras haces cualquier otra cosa con el teléfono.

Admite la instalación de tiendas de aplicaciones

Android te permite instalar tiendas de aplicaciones de terceros, si no quieres depender de Google para bajar software o quieres acceder a herramientas que no se encuentran en la Play Store.

Inconvenientes

Duración de la batería

La primera desventaja del sistema Android, y una de las más conocidas, se desprende de su sistema multitarea, y es el hecho de que la batería de los dispositivos se consume más y, por consiguiente, se agota mucho antes, que en el caso de otros sistemas.

Vulnerabilidad

El hecho de tener un código abierto, también posibilita que este sistema sea más susceptible a ataques y a que algunos hackers aprovechen errores del propio sistema para atacarlo.

Falta de soporte de actualización

A diferencia de otros sistemas operativos, Android no posee un soporte de actualización que permita obtener la última versión del equipo para nuestro uso y disfrute.

El rendimiento no es el mejor en dispositivos de gama baja

Los dispositivos de gama baja suelen presentar un peor rendimiento con Android.

4. Convivencia con otros sistemas operativos

La convivencia de Android con otros sistemas operativos se puede analizar por medio de diferentes ámbitos: ventas, compatibilidad y otros aspectos:

Ventas y Cuota de Mercado:

Android: Android sigue siendo el sistema operativo móvil más utilizado en todo el mundo, con una cuota de mercado significativamente alta. Esto se debe en parte a que Android es utilizado por una amplia variedad de fabricantes de dispositivos, lo que resulta en una gama diversa de opciones para los consumidores.

iOS: iOS, de Apple, aunque no tiene la cuota de mercado más grande en términos de unidades vendidas, suele tener una participación significativa en el segmento de dispositivos de gama alta, lo que le permite generar ingresos considerables y mantener una base de usuarios leales.

Otros Sistemas Operativos: Fuera de Android e iOS, hay sistemas operativos menos comunes, como HarmonyOS de Huawei y KaiOS, que se utilizan en dispositivos específicos y en regiones particulares. Estos sistemas tienen una cuota de mercado más limitada en comparación con Android y iOS.

Compatibilidad:

Interoperabilidad Limitada: En términos de compatibilidad, la mayoría de los sistemas operativos móviles son propietarios y no están diseñados para funcionar directamente en hardware de otros fabricantes. Esto significa que, en general, no puedes instalar un sistema operativo de una marca en un dispositivo de otra marca.

Aplicaciones y Servicios Cruzados: Sin embargo, muchas aplicaciones y servicios, como redes sociales y aplicaciones de mensajería, están disponibles en múltiples plataformas, lo que permite a los usuarios de diferentes sistemas comunicarse y compartir contenido.

Coexistencia y Elección del Usuario:

Elección del Consumidor: Los usuarios pueden elegir el sistema operativo móvil que mejor se adapte a sus necesidades y preferencias. Esto ha llevado a una diversidad de dispositivos móviles en el mercado, lo que beneficia a los consumidores al ofrecerles una variedad de opciones.

Cambios entre Plataformas: Algunos usuarios cambian entre sistemas operativos móviles a lo largo del tiempo, ya sea por preferencia personal o por razones específicas (por ejemplo, la integración con otros dispositivos).

Desafíos y Competencia:

Competencia y Evolución: La competencia entre los sistemas operativos móviles ha llevado a mejoras constantes en términos de características, seguridad y privacidad. Los fabricantes y desarrolladores buscan innovar para atraer y retener a los usuarios.

Desafíos de Ecosistema: Cada sistema operativo móvil tiene su propio ecosistema de aplicaciones y servicios, lo que a veces puede dificultar la transición de un sistema a otro debido a la dependencia de aplicaciones específicas.

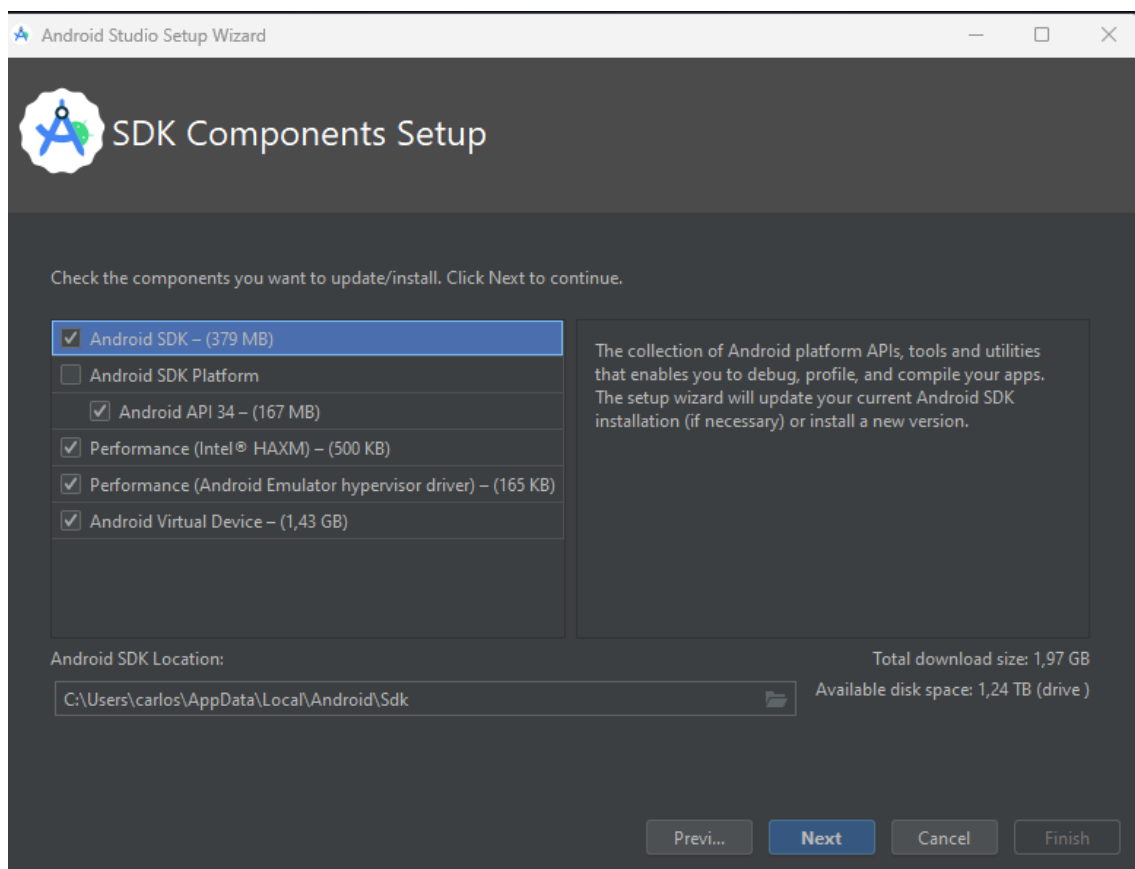
En resumen, Android es el sistema operativo móvil más utilizado, mientras que iOS también mantiene una base de usuarios significativa. Otros sistemas operativos móviles, como HarmonyOS y KaiOS, tienen presencia en nichos específicos. La coexistencia entre estos sistemas se basa en la elección del consumidor y la disponibilidad de aplicaciones y servicios en múltiples plataformas. Aunque la compatibilidad directa entre sistemas operativos móviles es limitada, la interoperabilidad entre dispositivos y plataformas ha mejorado con el tiempo.

Android como Framework

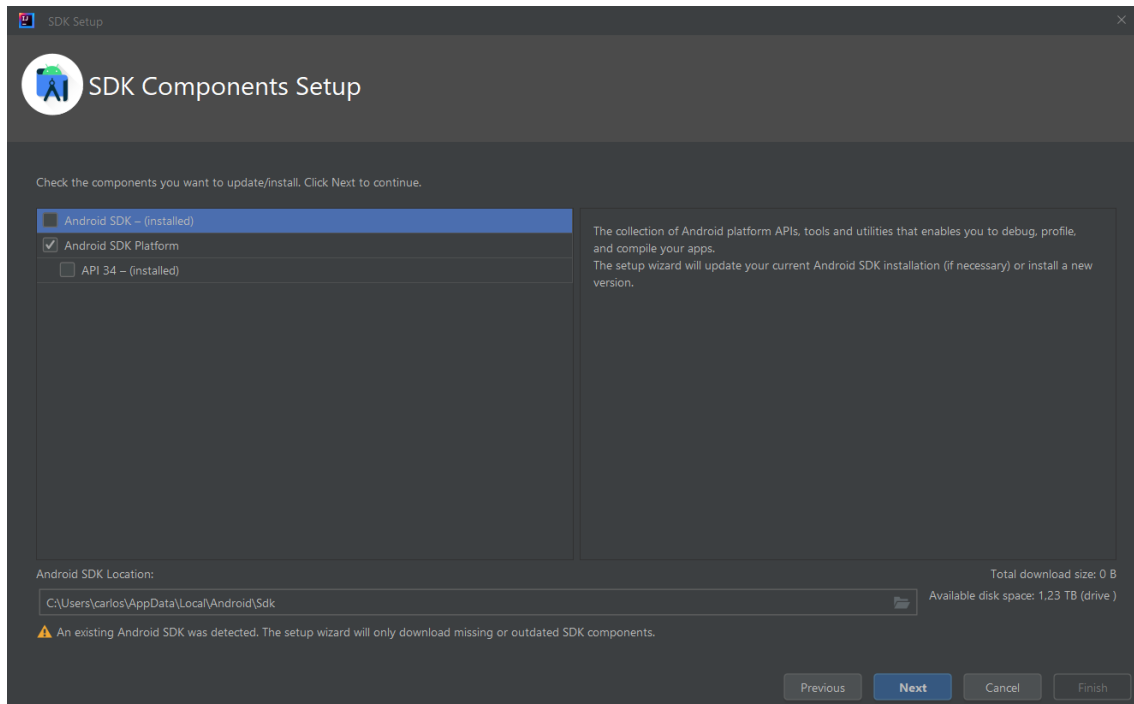
1. Android SDK

Android SDK es kit de desarrollo de software, el cual fue diseñado por Google con el propósito de proporcionarle a los desarrolladores algunos elementos necesarios para las apps destinadas a dispositivos Android. Se suele implementar junto con el entorno de desarrollo llamado Android Studio. Esto se debe a que la forma más fácil de descargar el Android SDK es a través del entorno de desarrollo que propone Google.

Cuando instalar Android Studio, te da la posibilidad de instalar junto con el Android SDK.



También hay complementos en los diferentes IDEs para poder realizar aplicaciones que precisen de Android SDK, por ejemplo en IntelliJ se puede instalar de la siguiente forma.



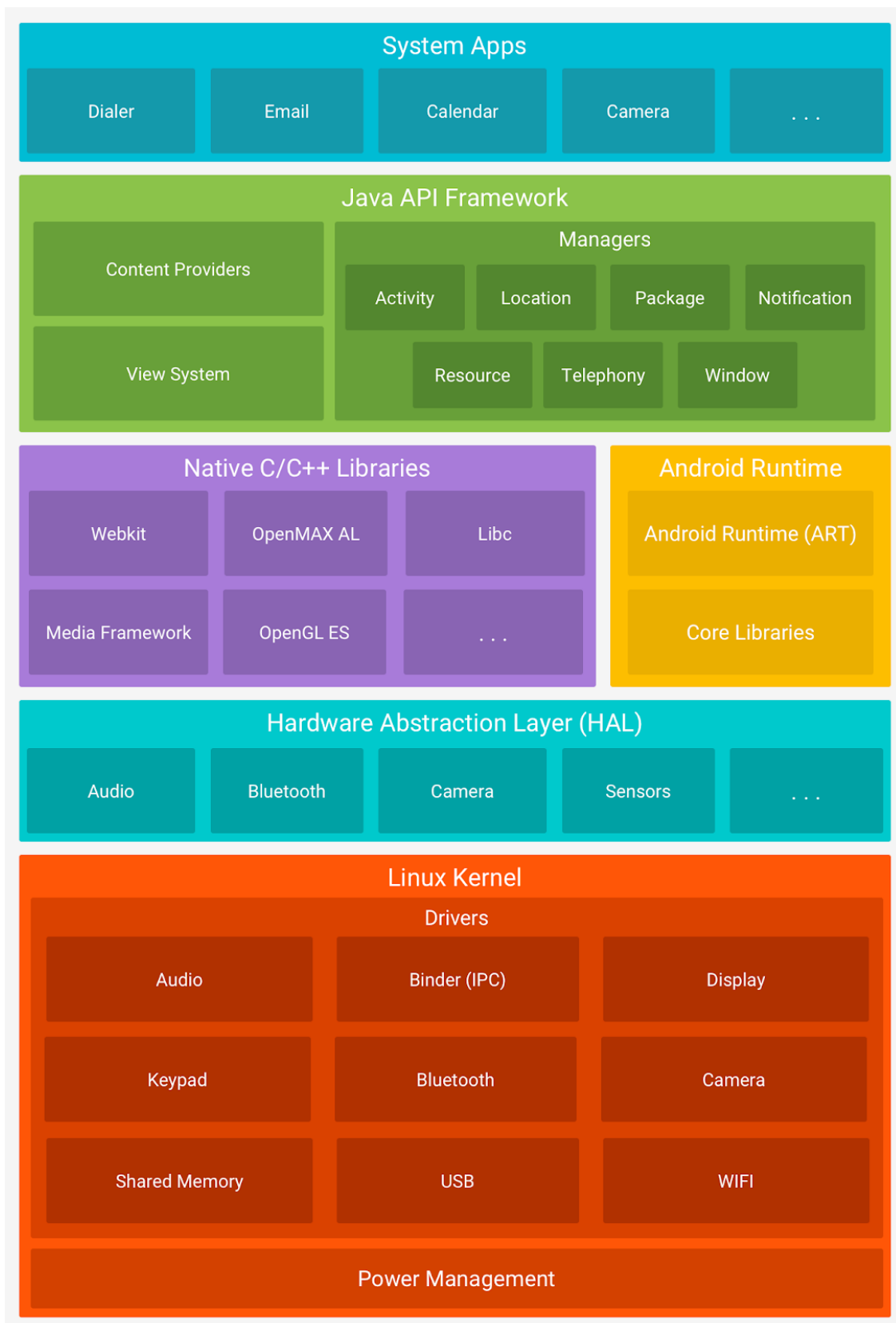
2. Android Studio

Es un entorno de desarrollo integrado que se centra en el desarrollo de apps Android y, por tanto, al igual que Android SDK, también está diseñado por Google. Además, está basado en otro entorno de desarrollo, IntelliJ IDEA.

Una gran ventaja de Android Studio es que se encuentra disponible para diferentes plataformas o sistemas, tales como Linux, MacOS, Windows y Chrome. Asimismo, es de gratuito. Estos son algunos de los aspectos por los que se puede categorizar a Android Studio como un entorno de desarrollo accesible y práctico.

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help My Application - .../Carlo's Android Studio Projects/example/app/src/main/java/com/example/myapplication/MainActivity.java
example / app / src / main / java / com / example / myapplication / MainActivity.java Add Configuration... Pixel_3a_API_34_extension_level_7_x86c
Project loading...
no usages
24 private AppBarConfiguration appBarConfiguration;
no usages
25 private ActivityMainBinding binding;
26
no usages
27 @Override
28 protected void onCreate(Bundle savedInstanceState) {
29     super.onCreate(savedInstanceState);
30
31     binding = ActivityMainBinding.inflate(getLayoutInflater());
32     setContentView(binding.getRoot());
33
34     setSupportActionBar(binding.toolbar);
35
36     NavController navController = Navigation.findNavController(this, R.id.nav_host_fragment_content_main);
37     appBarConfiguration = new AppBarConfiguration.Builder(navController.getGraph()).build();
38     NavigationUI.setupActionBarWithNavController(this, navController, appBarConfiguration);
39
40     binding.fab.setOnClickListener(new View.OnClickListener() {
41         no usages
42         @Override
43         public void onClick(View view) {
44             Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
45                 .setAnchorView(R.id.fab)
46                 .setAction("Action", null).show();
47         }
48     });
49
no usages
50 @Override
51 public boolean onCreateOptionsMenu(Menu menu) {
52     // Inflate the menu; this adds items to the action bar if it is present.
53     getMenuInflater().inflate(R.menu.menu_main, menu);
54     return true;
55 }
56
Version Control TODO Problems Terminal Services App Quality Insights App Inspection Logcat Build
Gradle sync started (moments ago) Gradle: Build model... Show all (2) 3:8 LF UTF-8 4 spaces
```

Estructura de Android



1. Kernel

El Kernel se encarga de hacer de puente entre los componentes de hardware y las aplicaciones, con unos cuantos pasos intermedios. En el Kernel se encuentran los controladores para componentes como la pantalla, el audio, las cámaras integradas o la gestión energética. Gestionar los recursos de la CPU, la memoria, los sistemas de archivos y básicamente todas las tareas más básicas que son necesarias para el funcionamiento del móvil, incluyendo los procesos de cada aplicación.

2. Capa de abstracción de hardware

Proporciona interfaces estándares que exponen las capacidades de hardware del dispositivo al framework de la API de Java. 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 framework de una API realiza una llamada para acceder al hardware del dispositivo, el sistema Android carga el módulo de biblioteca para el componente de hardware en cuestión.

3. Android Runtime

Android Runtime es un entorno de ejecución utilizado por el sistema operativo móvil Android. Lleva a cabo la transformación de la aplicación en instrucciones de máquina, que luego son ejecutadas por el entorno de ejecución nativo del dispositivo.

4. Bibliotecas C/C++ nativas

Muchos componentes y servicios de 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 el API del framework 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 framework de Android para agregar a tu app compatibilidad con los dibujos y la manipulación de gráficos 2D y 3D.

5. Framework del API de Java

Todo el conjunto de funciones del SO Android está disponible mediante API escritas en el lenguaje Java. Estas API son los cimientos que se necesitan para crear apps de Android simplificando la reutilización de componentes del sistema y servicios centrales y modulares.

6. Apps del sistema

En Android se incluye un conjunto de apps para correo electrónico, mensajería SMS, calendarios, navegación en Internet y contactos, entre otros elementos. Las apps 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.

Las apps del sistema funcionan como apps para los usuarios y proporcionan 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.