

# Características y Componentes de las Aplicaciones Android

Qué bueno recibirte en este espacio de conocimiento para seguir profundizando en los conceptos relacionados al desarrollo en Android. Seguidamente te proporcionamos un resumen de las características y componentes de una aplicación.

## Características de las Aplicaciones Android

Las aplicaciones de Android se pueden escribir usando los lenguajes **Kotlin, Java y C ++**. Las herramientas del SDK de Android compilan su código junto con cualquier archivo de datos y recursos en un APK, que es un paquete de Android incluido en un archivo con un sufijo .apk. Un archivo APK contiene todo el contenido de una aplicación de Android y es utilizado por los dispositivos Android para instalar la aplicación.

Cada aplicación de Android vive en su propio entorno limitado y seguro, protegido por las siguientes características de seguridad de Android:

- El sistema operativo **Android es un sistema Linux multiusuario** en el que cada aplicación es un usuario diferente.
- De manera predeterminada, **el sistema asigna a cada aplicación una ID de usuario de Linux único** (sólo el sistema usa la ID y la aplicación lo desconoce). El sistema establece permisos para todos los archivos en una aplicación, de forma tal que sólo 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 aislada de otras aplicaciones.
- Por defecto, **cada aplicación se ejecuta en su propio proceso de Linux**. El sistema operativo Android inicia el proceso cuando se necesita ejecutar cualquiera de los componentes de la aplicación, y luego cierra el proceso cuando ya no es necesario o cuando el sistema debe recuperar la memoria para otras aplicaciones.

El sistema Android implementa el principio de menor privilegio. Es decir, cada aplicación, por defecto, tiene acceso sólo a los componentes que necesita para hacer 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 cuales no tiene permiso. Sin embargo, hay formas para que una aplicación comparta datos con otras aplicaciones, así como una aplicación pueda acceder a los servicios del sistema, por ejemplo:

Es posible organizar que dos aplicaciones compartan la misma ID de usuario de Linux, en cuyo caso pueden acceder a los archivos del otro. Para conservar los recursos del sistema, las aplicaciones con la misma ID de usuario también pueden organizar la ejecución en el mismo proceso de Linux y compartir la misma Máquina Virtual (VM). Las aplicaciones también deben estar firmadas con el mismo certificado de seguridad.

Una aplicación puede solicitar permiso para acceder a los datos del dispositivo, como la ubicación del dispositivo, la cámara y la conexión Bluetooth. El usuario tiene que otorgar explícitamente estos permisos.

## Componentes de una Aplicación Android

Las aplicaciones de Android constan de componentes acoplados libremente, vinculados por el manifiesto de la aplicación que describe cada componente y cómo interactúan. Los siguientes son los componentes básicos para todas sus aplicaciones de Android:

- **Actividades:** representan las interfaces gráficas de nuestra aplicación, donde cada interfaz se basa en una o más extensiones de la clase Actividad. Las actividades usan Fragmentos y Vistas para diseñar y mostrar información, además para responder a las acciones del usuario. En comparación con el desarrollo de escritorio, las actividades son equivalentes a las ventanas.
- **Servicios:** los componentes de servicio se ejecutan sin una interfaz de usuario, actualizando sus fuentes de datos, iniciando notificaciones y transmitiendo intenciones (intents). Se utilizan para realizar tareas de larga duración, o aquellas que no requieren la interacción del usuario (tales como las tareas que deben continuar, incluso cuando las actividades de su aplicación no están activas o visibles).
- **Intentos (intents):** constituyen un mecanismo de paso de mensajes entre aplicaciones. Los intentos se usan ampliamente en todo Android para iniciar y detener Actividades y Servicios, para transmitir mensajes a todo el sistema o a una Actividad, Servicio o Receptor de Difusión específico, o para solicitar que se realice una acción en un dato en particular.
- **Receptores de difusión:** los receptores de difusión (o simplemente "receptores") se utilizan para recibir Intents de difusión, lo que permite que su aplicación escuche Intents que coincidan con los criterios que especifique. Los receptores de difusión configuran nuestra aplicación para reaccionar ante cualquier intent recibido, lo que los hace perfectos para crear aplicaciones basadas en eventos.
- **Proveedores de contenido:** los proveedores de contenido son los medios más usados para compartir datos entre las aplicaciones. Se pueden configurar para permitir el acceso desde otras aplicaciones y puede acceder a los proveedores de contenido expuestos por otros. Los dispositivos Android incluyen varios proveedores de contenido nativos que exponen bases de datos, tales como la tienda de medios y los contactos.
- **Notificaciones:** las notificaciones nos permiten alertar a los usuarios sobre los eventos de la aplicación sin robar el foco o interrumpir su actividad actual. Es la técnica preferida para llamar la atención de un usuario, cuando la aplicación no está visible o activa, pero es activada desde un Servicio o Receptor de difusión. Por ejemplo, cuando un dispositivo recibe un mensaje de texto o un correo electrónico, las aplicaciones de mensajería y Gmail usan notificaciones para alertarnos.

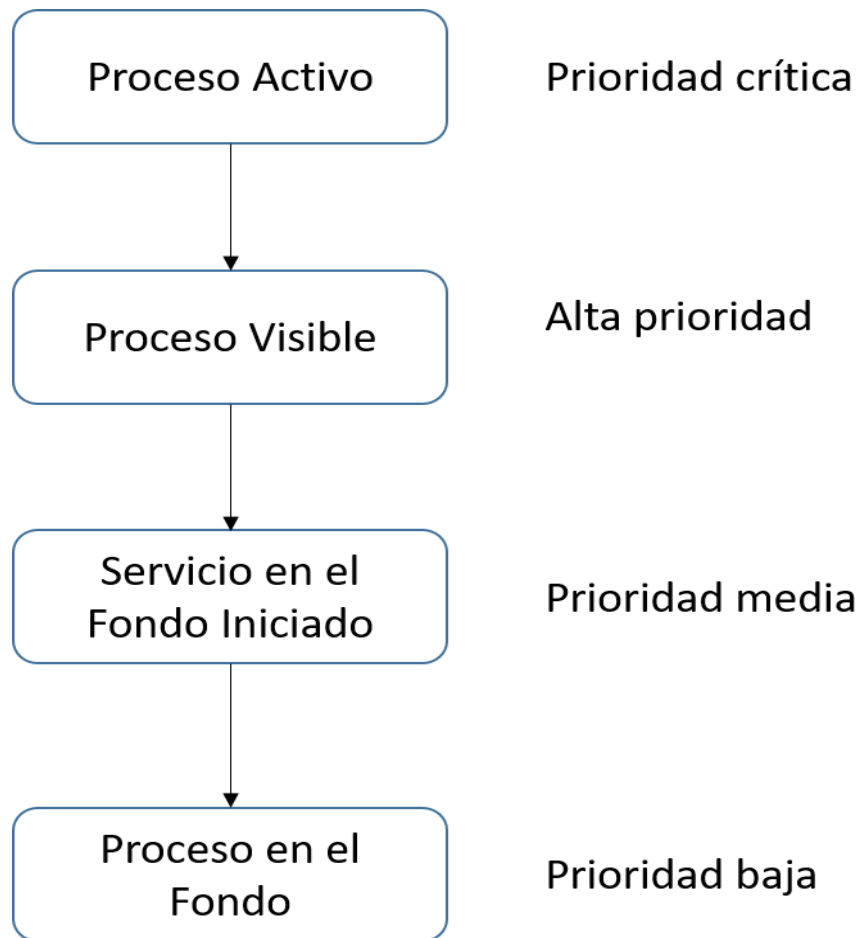
Al desacoplar las dependencias entre los componentes de la aplicación, se puede compartir y utilizar proveedores de contenido, servicios e incluso actividades individuales con otras aplicaciones, tanto las nuestras como las de terceros.

## **Ciclo de Vida de una Aplicación Android**

A diferencia de muchas plataformas de aplicaciones tradicionales, las aplicaciones de Android tienen un control limitado sobre su propio ciclo de vida. En su lugar, los componentes de la aplicación deben escuchar los cambios en el estado de la aplicación y reaccionar en consecuencia, teniendo especial cuidado de estar preparados para la finalización prematura.

Android administra agresivamente sus recursos, haciendo lo que sea necesario para garantizar una experiencia de usuario estable y fluida. En la práctica, eso significa que los procesos (y, por lo tanto, las aplicaciones) se eliminarán, en algunos casos sin previo aviso, para liberar recursos si las aplicaciones de mayor prioridad así lo requieren. El orden en que se eliminan los procesos para reclamar recursos está determinado por la prioridad de sus aplicaciones alojadas. La prioridad de una aplicación es igual a la de su componente de mayor prioridad.

Si dos aplicaciones tienen la misma prioridad, el proceso que ha tenido esa prioridad durante más tiempo normalmente se eliminará primero. La prioridad del proceso también se ve afectada por las dependencias entre procesos. Si una aplicación depende de un proveedor de servicios o contenido suministrado por una segunda aplicación, la aplicación secundaria tiene asignada al menos una prioridad tan alta como la aplicación que admite. Es importante estructurar nuestra aplicación para garantizar que su prioridad sea adecuada para el trabajo que está realizando. Si no lo hacemos, nuestra aplicación podría ser eliminada mientras está en medio de alguna acción importante, o podría seguir ejecutándose mientras pueda terminar de forma segura para liberar recursos y así mantener una experiencia de usuario fluida. La figura 1 muestra el árbol de prioridad utilizado para determinar el orden de finalización de las aplicaciones.



La siguiente lista detalla cada uno de los estados de la aplicación que se muestran en la figura anterior, explicando cómo el estado está determinado por los componentes de la aplicación:

- **Procesos activos (prioridad crítica):** los procesos activos (primer plano) tienen componentes de la aplicación con los que el usuario está interactuando. Estos son los procesos que Android intenta que se sigan ejecutando sin problemas y de manera fluida, recuperando recursos de otras aplicaciones. Generalmente hay muy pocos de estos procesos.

Los procesos activos incluyen uno o más de los siguientes componentes:

- Actividades en un estado activo, es decir, aquellas en primer plano que responden a eventos de interacción con el usuario.
  - Receptores de difusión que ejecutan controladores de eventos `onReceive`.
  - Servicios que ejecutan controladores de eventos `onStart`, `onCreate` u `onDestroy`.
- **Procesos visibles (alta prioridad):** los procesos visibles pero inactivos son aquellos que alojan actividades "visibles" o servicios en primer plano. Como su

nombre lo indica, estas actividades son visibles pero no están en primer plano ni responden a la entrada del usuario. Esto sucede cuando una Actividad está solo parcialmente oculta (por una Actividad que no es de pantalla completa o transparente), o es una ventana inactiva en un entorno de múltiples ventanas. En general, hay muy pocos procesos visibles y se eliminarán sólo en circunstancias extremas para permitir que continúen los procesos activos.

- **Procesos de servicio en segundo plano iniciados (Prioridad media):** procesos que alojan servicios en segundo plano que se han iniciado. Debido a que estos Servicios no interactúan directamente con el usuario, reciben una prioridad ligeramente menor que las Actividades visibles o los Servicios en primer plano.
- **Procesos en segundo plano (baja prioridad):** procesos que alojan actividades que no son visibles y que no tienen ningún servicio en ejecución. Cada vez que cambia de una aplicación a otra, presionando la tecla de inicio o usando el selector de "aplicaciones recientes", la aplicación activa anterior pasa a segundo plano. En general, habrá una gran cantidad de aplicaciones en segundo plano que Android eliminará utilizando el patrón de prioridades descrito previamente, dando cierta consideración a priorizar la eliminación de aplicaciones que consumen más memoria, a fin de obtener recursos para procesos en primer plano.

Esperamos hayas disfrutado de este breve contenido y sigas adelante con las actividades y recursos que hemos preparado para ti.