

FEUD

Fundación de egresados U. Distrital
Construyendo Profesionales



Somos el Centro de Entrenamiento Autorizado por marcas representativas en Gobierno TI y empresa, con el portafolio más amplio en Latinoamérica:



Todas las marcas, nombres comerciales, marcas de servicios y logotipos a los que se hace referencia en el presente documento pertenecen a sus respectivas empresas. ITIL® es una marca registrada de AXELOS Limited. El Swirl logo™ es una marca de AXELOS Limited. PMP, PMI, PMI-RMP, PMI-ACP, PMI-SP, PgMP, CAPM, REP Logo son marcas registradas del Project Management Institute.

FEUD
Fundación de egresados U. Distrital
Construyendo Profesionales



Desarrollo de aplicaciones utilizando SDK - Android

Diplomado Desarrollo de Software para móviles

PROGRAMACIÓN NATIVA ANDROID

¿Por qué el nombre Android?

- Un androide es un *autómata de figura de hombre*, es decir, un robot que se comporta como humano, se ve como humano, pero no es humano
- Similarmente, la plataforma Android se comporta como Java, se ve como Java, pero no es Java

Java Vs Android

JVM (Maquina Virtual de Java)

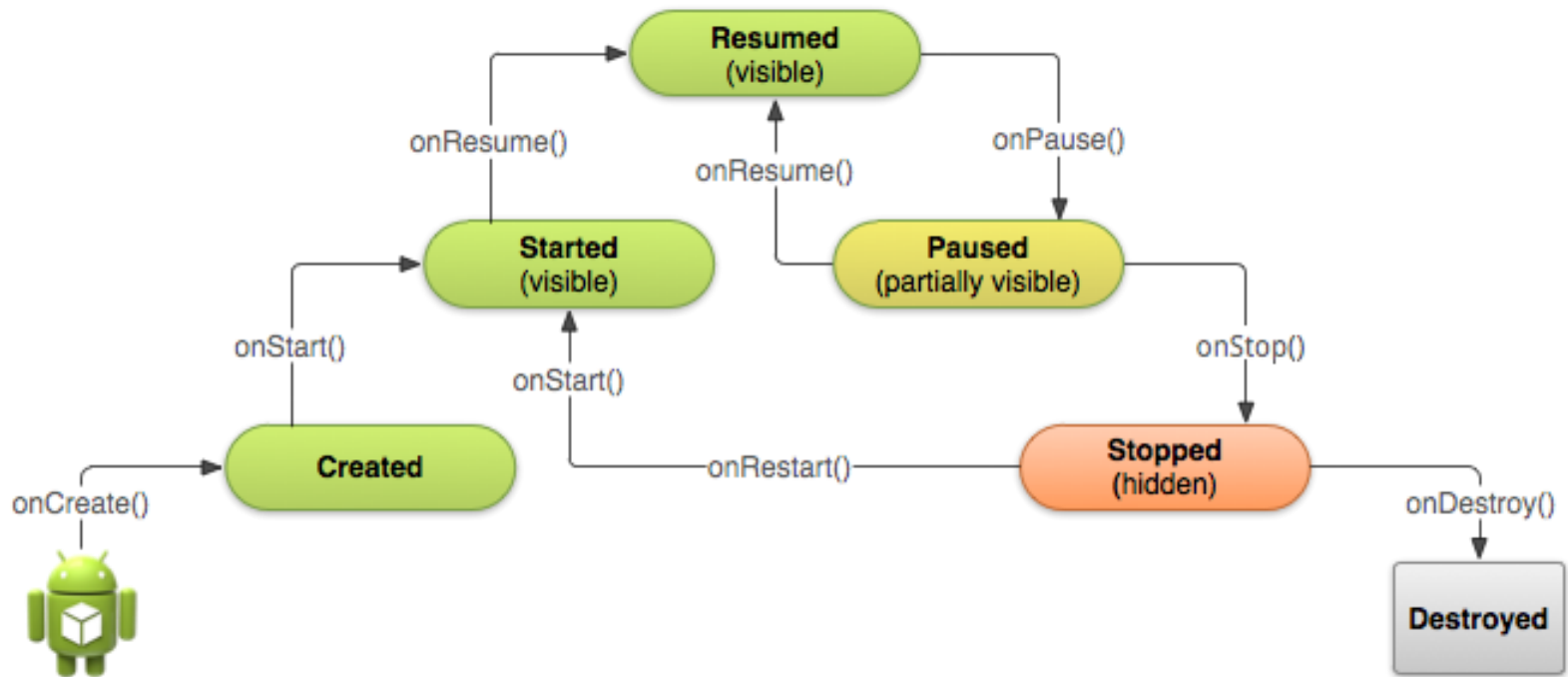
- Archivos .java se compilan en archivos .class
- Los archivos .class se empaquetan en un .jar/ear/war/ETC.

Dalvik (Maquina virtual de Android)

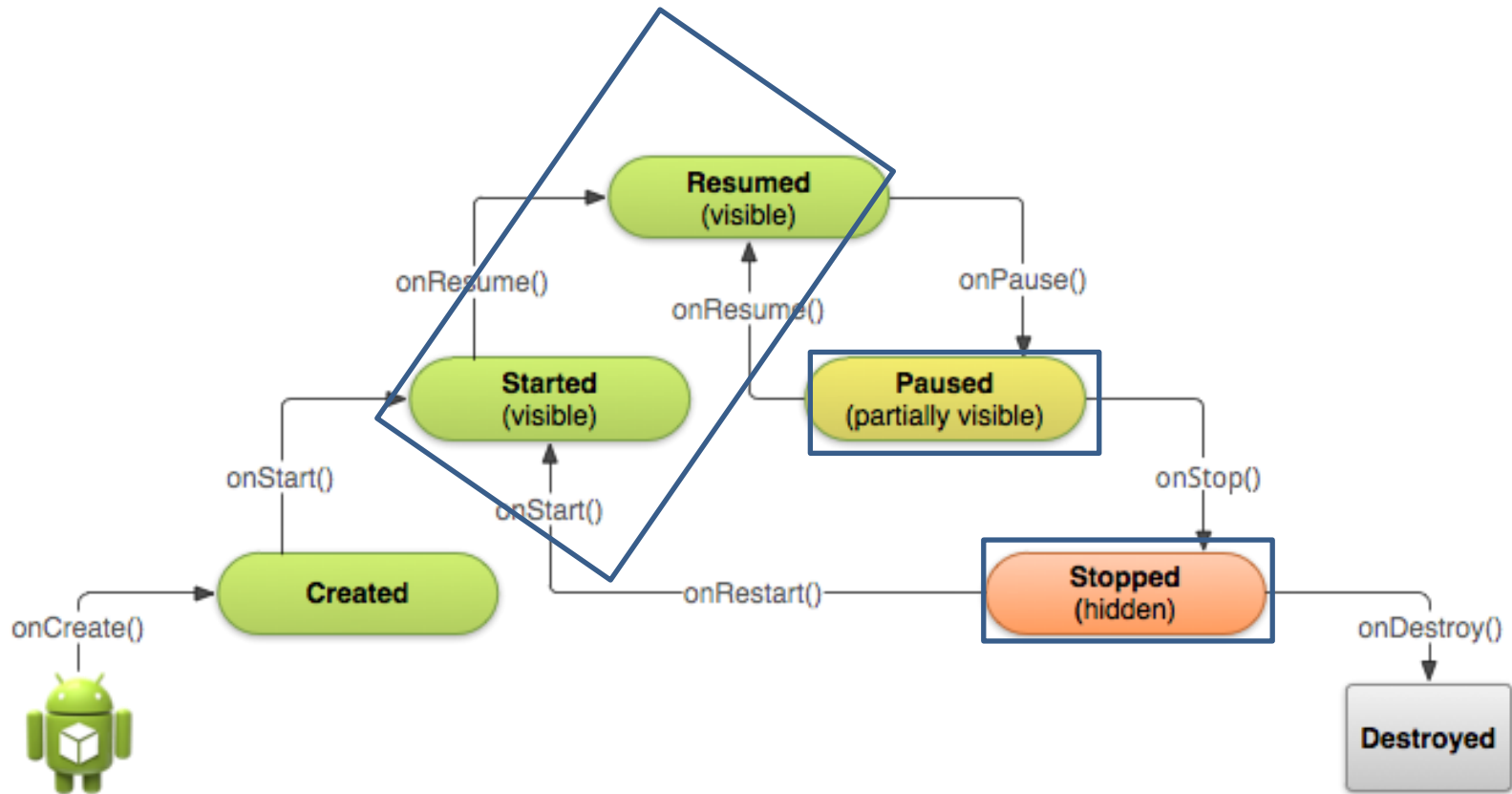
- Archivos .java se compilan en archivos .dex
- Los archivos .dex se empaquetan en un .apk
- El bytecode es completamente diferente del de Java

ANDROID: CICLO DE VIDA

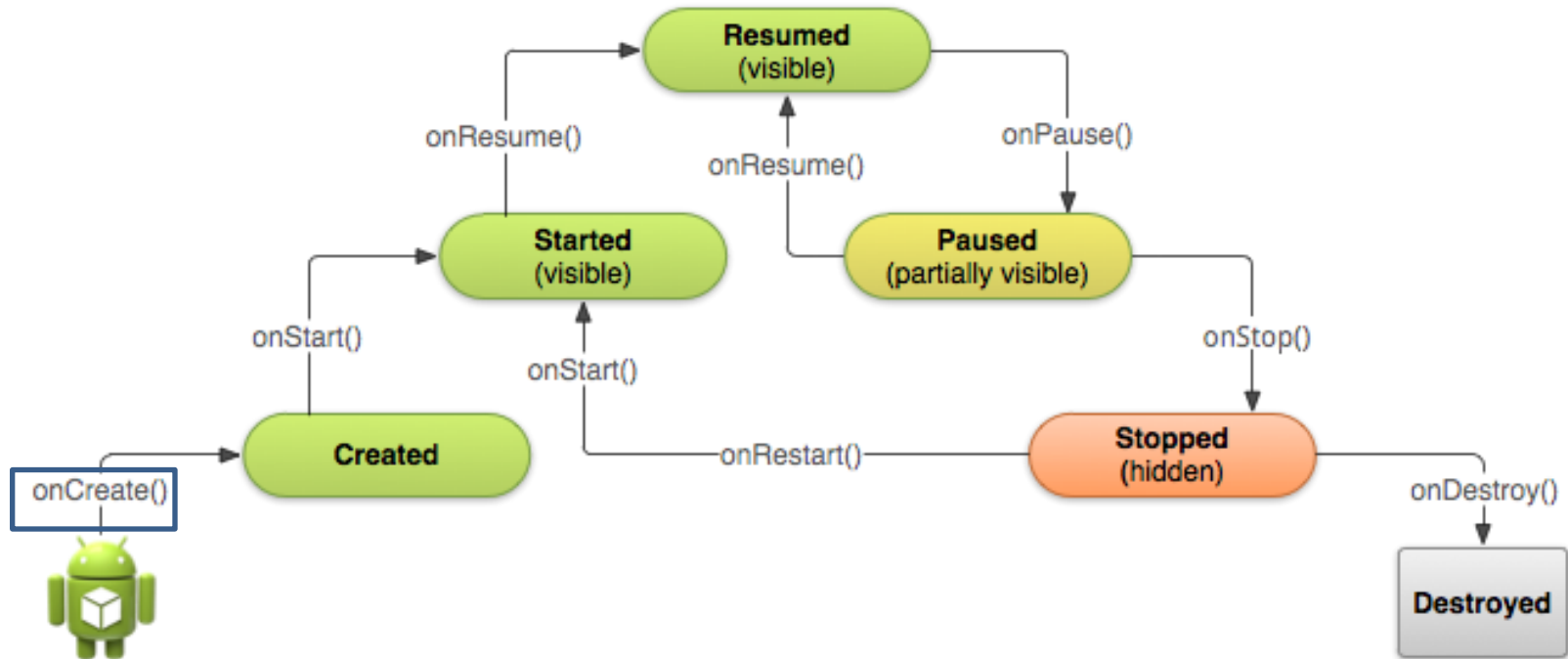
Android: Ciclo de vida



Android: Ciclo de vida: Estados

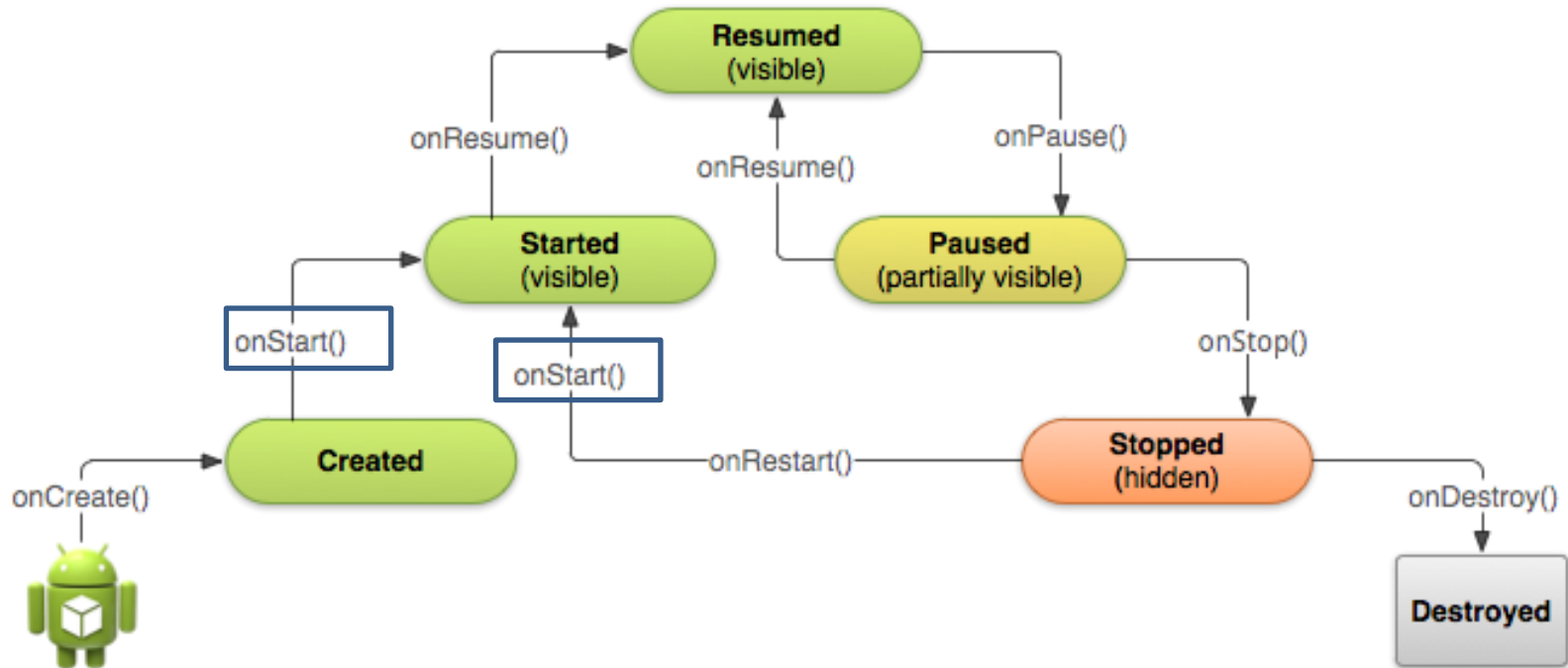


Android: Ciclo de vida



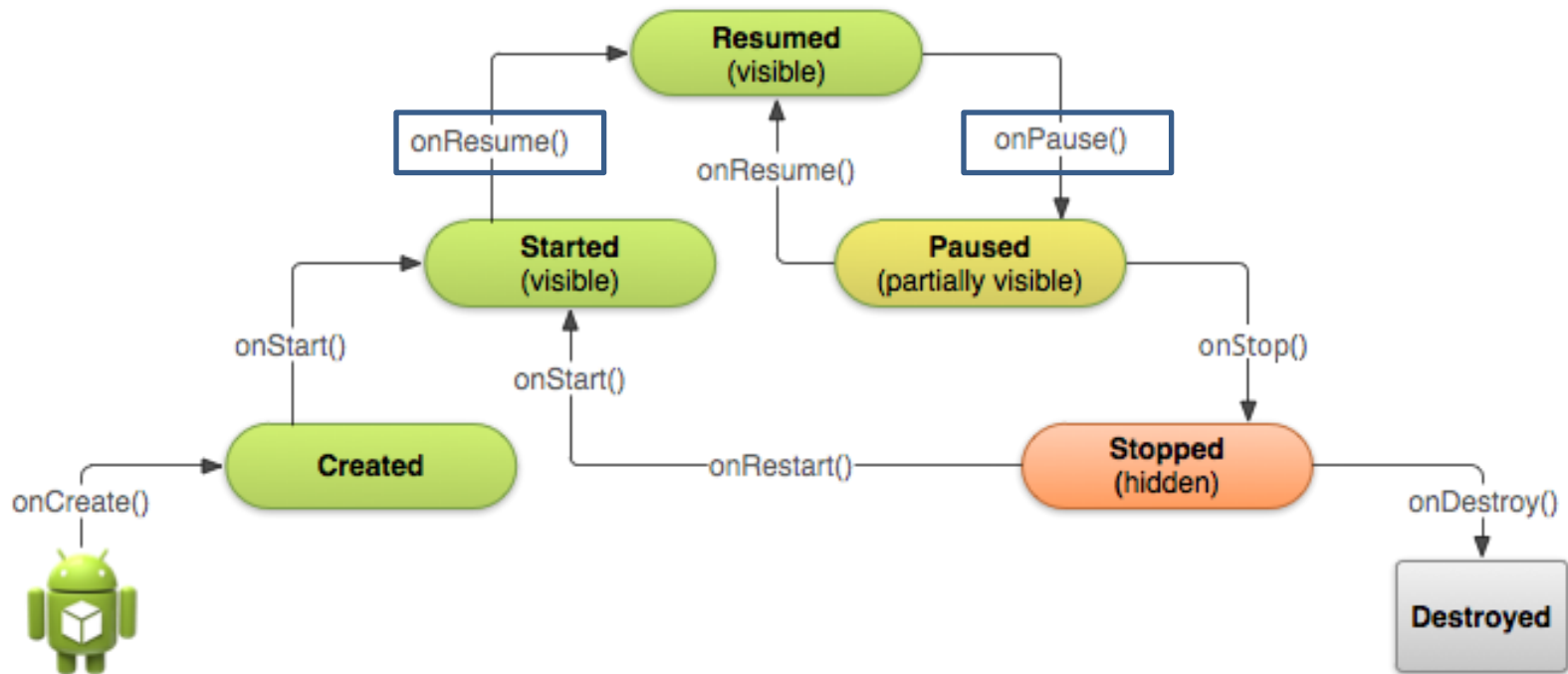
Función `onCreate()`, se ejecuta cuando se crea la aplicación. En este punto se deben hacer los llamados a las vistas y obtener recursos mas no iniciarlos

Android: Ciclo de vida



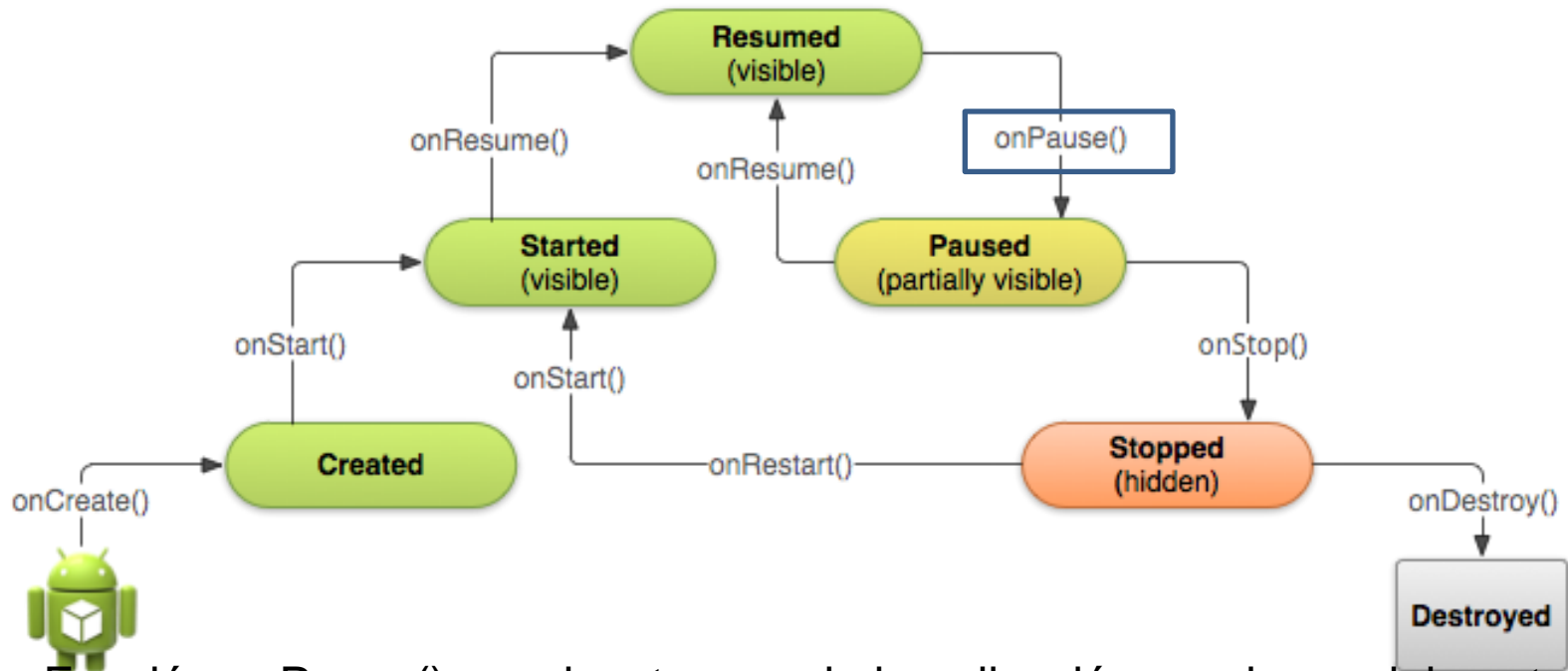
Función `onStart()`, se ejecuta cuando se crea la aplicación, después de `onCreate()` o cuando se “despierta” la aplicación estando detenida. En este paso se deben iniciar los recursos.

Android: Ciclo de vida



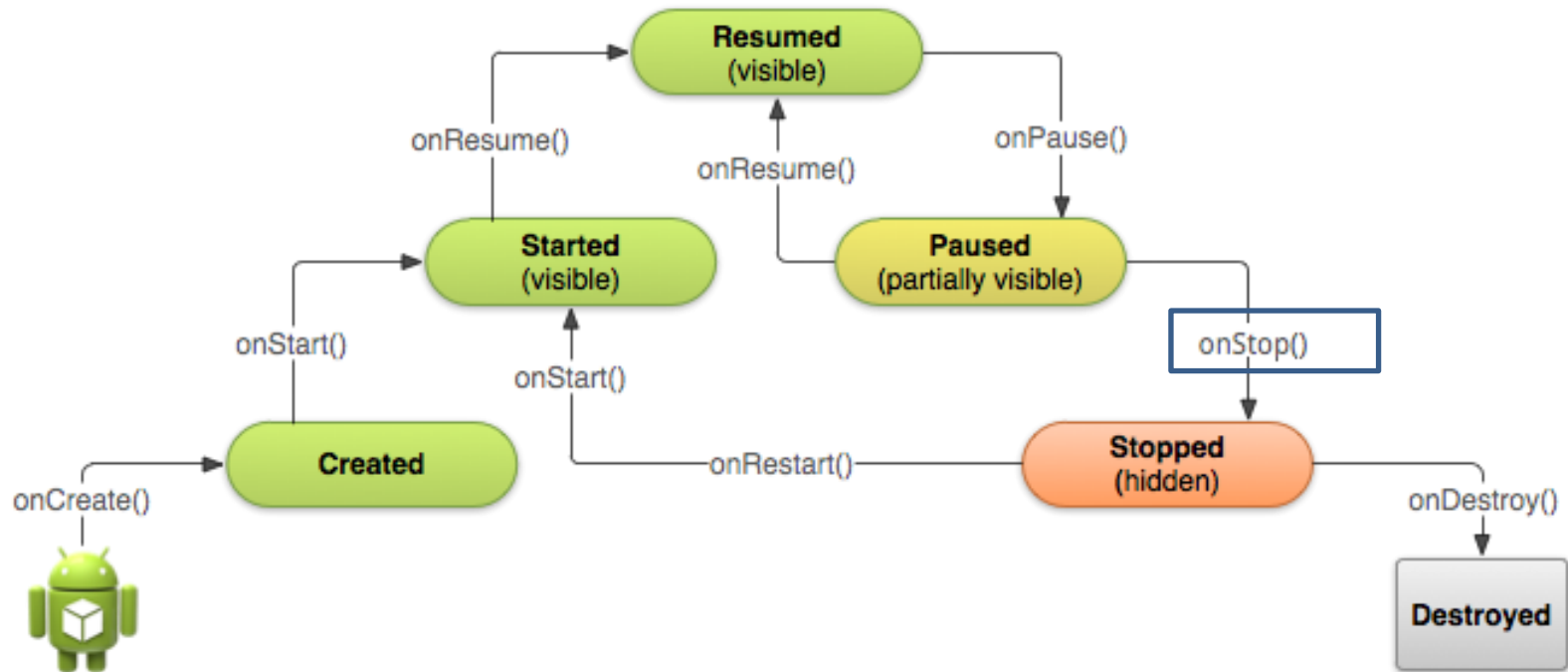
Función `onResume()`, se ejecuta cuando se crea la aplicación, después de `onCreate()` o cuando se “enfoca” a la aplicación parcialmente visible. Se debe recuperar la información que se almacenó en el proceso de pausa

Android: Ciclo de vida



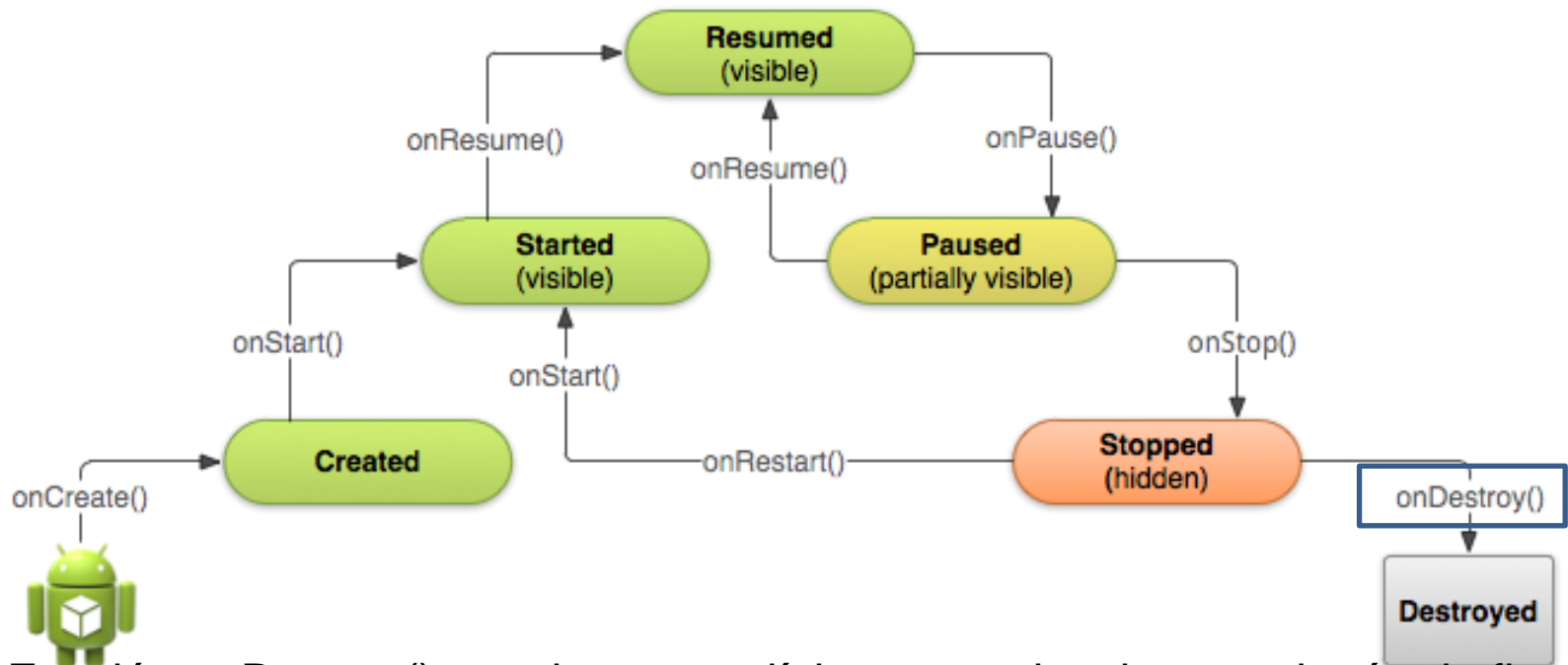
Función `onPause()`, se ejecuta cuando la aplicación queda parcialmente visible, como tapada por un menú o un aviso del sistema. Se deben almacenar datos sensibles en Bundles

Android: Ciclo de vida



Función `onStop()`, se ejecuta cuando la aplicación queda totalmente invisible. Se deben liberar recursos adicionales y almacenar datos necesarios para su reinicio. Se pueden hacer operaciones de almacenamiento de datos con archivos

Android: Ciclo de vida



Función `onDestroy()`, se ejecuta explícitamente si se invoca el método `finalize()` o implícitamente cuando el sistema operativo requiere mayor memoria. En este punto se debe realizar el almacenamiento en bases de datos y liberar la totalidad de los recursos

Bloques Básicos de una Aplicación

- Activities
- Intents
- Views
- Services
- Content Providers
- Broadcast Receivers

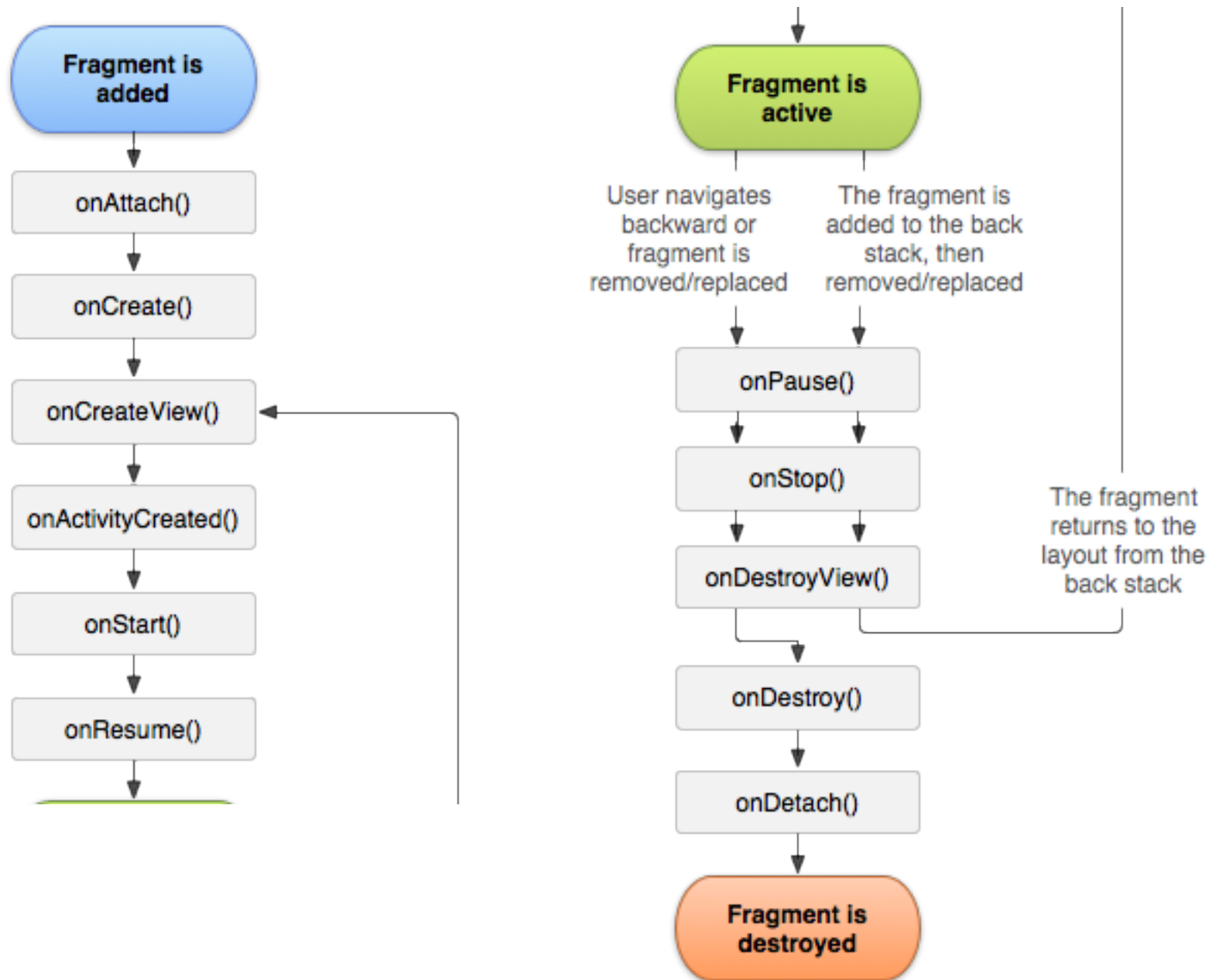
Activities

- Una actividad se asocia con una pantalla.
- Una aplicación puede tener varias actividades.
- Por ejemplo, una aplicación de un Reloj puede tener dos actividades:
 - Una actividad que muestra el tiempo actual.
 - Una actividad utilizada para configurar la hora.

Fragments

Un Fragment representa un comportamiento o una parte de la interfaz de usuario en una Activity. Puedes combinar múltiples fragmentos en una sola actividad para crear una IU multipanel y volver a usar un fragmento en múltiples actividades. Puedes pensar en un fragmento como una sección modular de una actividad que tiene su ciclo de vida propio, recibe sus propios eventos de entrada y que puedes agregar o quitar mientras la actividad se esté ejecutando (algo así como una “subactividad” que puedes volver a usar en diferentes actividades).

Ciclo de vida de Fragments



Intents

- Mensajes que generan cambios o notificaciones.
- Pueden ser recibidos por actividades o servicios para ejecutar un proceso
- Mecanismo de comunicación para componentes dentro de una aplicación o entre aplicaciones

Views

- Componentes de la GUI.
- Se asocian en una forma jerárquica.
- Los componentes se disponen a través de un archivo XML (Esto ayuda a asegurar la separación de la vista con el modelo de una aplicación)

Services

- Componentes que ejecutan operaciones en segundo plano.
- Ejemplo: Un reproductor de música que se ejecuta en segundo plano sin importar la aplicación que se esté usando.

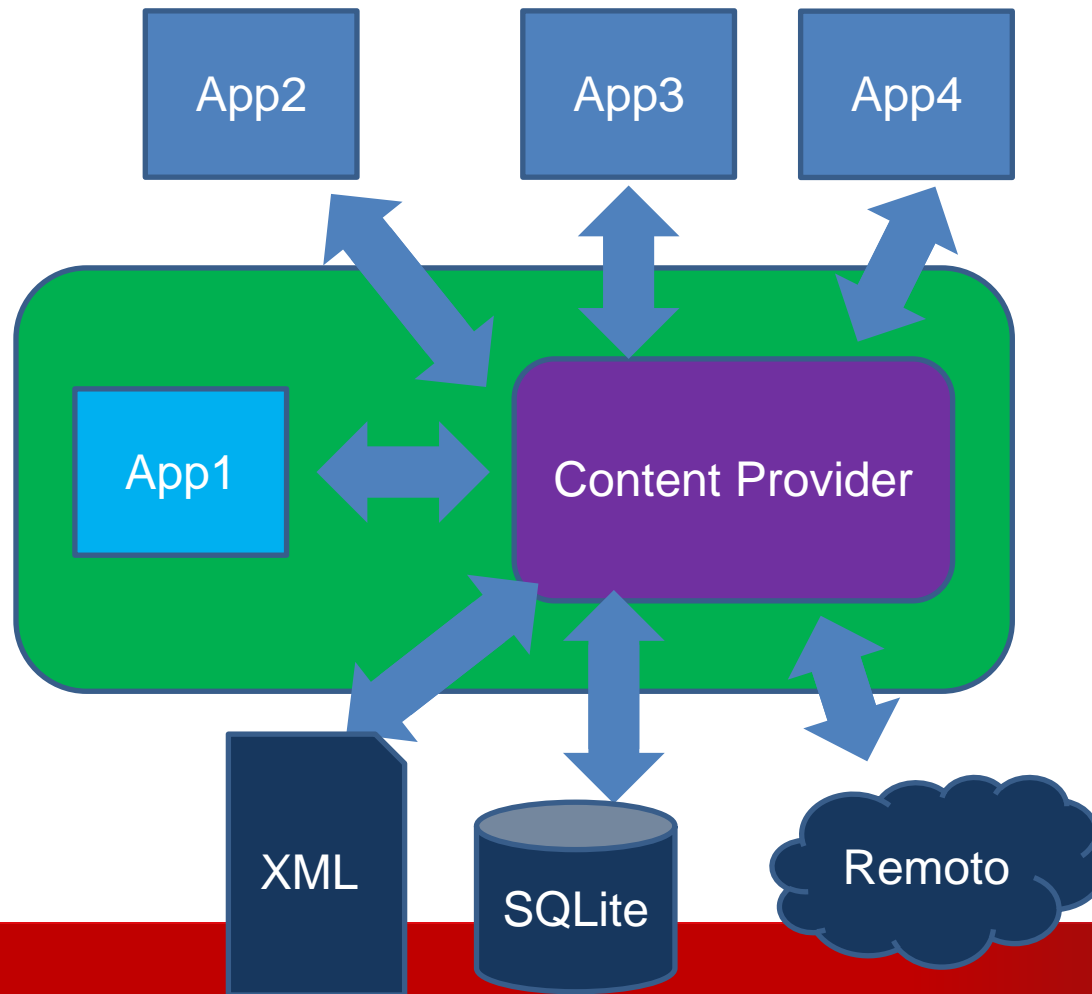
Content Providers

- Componentes que gestionan el acceso a un conjunto de datos estructurados (usualmente permanentes).
- Conecta datos entre procesos.
- Los datos pueden ser audio, video, imágenes, contactos, etc.
- Ejemplo: El sistema operativo maneja los contactos dentro de un Content Provider

Content Providers

- Un ContentProvider está pensado para que varias aplicaciones accedan a un mismo repositorio de datos a través de él
 - Ejemplo: la lista de contactos
- Es requerido siempre que se quiera **compartir datos entre múltiples aplicaciones**
- Su uso es similar al de una base de datos

Content Providers



Broadcast Receivers

- Iniciados por Intents de tipo difusión (broadcast)
- Pueden venir del sistema (Ej.: Batería baja, llamada entrante, etc.)
- Pueden ser generadas por el programador

HERRAMIENTAS DE DESARROLLO

Herramientas

- Gradle
- Android SDK
- Un IDE
- Gradle + Android SDK + IDE = Android Studio



Android
Studio

GRADLE

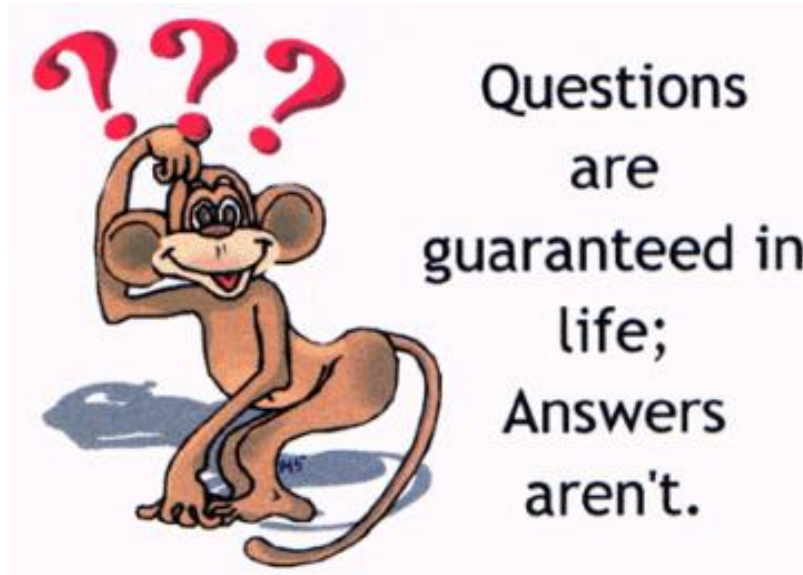


- Herramienta para construcción de un proyecto (compilación, pruebas, versiones, tareas, etc.) usada por Android
- Manejo de dependencias (librerías).
- Open – Source
- Funciona con Ant y Maven

Android SDK



- Herramientas de compilación para Android



¿Preguntas?

Diego Alberto Rincón Yáñez MSc.
Twitter: @d1egoprog.