



Android: Conceptos Básicos

Fundamentos de aplicación

- Las aplicaciones de Android están escritas en el lenguaje de programación Java.
- Las herramientas del SDK de Android compilan el código -junto con todos los datos y archivos de recursos- en un APK: un paquete de Android, que es un archivo con un sufijo “apk.”.
- Un archivo APK contiene todo el contenido de una aplicación para Android y es el archivo que los dispositivos con Android utilizan para instalar la aplicación.



Android: Conceptos Básicos

Fundamentos de aplicación

- Una vez instalada en un dispositivo, cada aplicación Android vive en su propio entorno limitado de seguridad:
 - El sistema operativo Android es un sistema Linux para múltiples usuarios en el que cada aplicación es un usuario diferente.
 - Por defecto, cada aplicación se ejecuta en su propio proceso de Linux.
 - Android comienza el proceso cuando alguno de los componentes de la aplicación necesita ser ejecutado, cierra el proceso cuando ya no es necesario o cuando el sistema debe recuperar la memoria para otras aplicaciones.
 - Por defecto, el sistema asigna a cada aplicación un ID de usuario único de Linux (el ID es utilizado sólo por el sistema y no se conoce a la aplicación).
 - El sistema establece permisos para todos los archivos en una aplicación para que sólo el ID de usuario asignado a esa aplicación se puede acceder a ellos.
 - Cada proceso tiene su propia máquina virtual (VM), por lo que el código de una aplicación se ejecuta en forma aislada de otras aplicaciones.



Android: Conceptos Básicos

Fundamentos de aplicación

- De esta manera, el sistema Android implementa el principio de *privilegios mínimos*. Es decir, cada aplicación, por defecto, sólo tiene acceso a los componentes que necesita para hacer su trabajo y nada más.
 - Esto crea un ambiente muy seguro en el que una aplicación no puede tener acceso a partes del sistema para el cual no se le da permiso.
- Sin embargo, hay maneras para que una aplicación para compartir datos con otras aplicaciones y para una aplicación de acceso a los servicios del sistema:
 - Es posible hacer que dos aplicaciones compartan el mismo Linux ID de usuario, en cuyo caso son capaces de acceder a los archivos de cada uno. Para conservar los recursos del sistema, las aplicaciones con el mismo ID de usuario también se pueden realizar las gestiones para ejecutar en el mismo proceso Linux y compartir la misma máquina virtual (las aplicaciones también deben estar firmadas con el mismo certificado).
 - Una aplicación puede solicitar permiso para acceder a los datos del dispositivo, como los contactos del usuario, mensajes SMS, el almacenamiento externo (tarjeta SD), cámara, Bluetooth, y mucho más. Todos los permisos de una aplicación deben ser concedidos por el usuario en el momento de la instalación.



Android: Conceptos Básicos

Componentes básicos

- Hay cuatro tipos diferentes de componentes en las aplicaciones Android.
- Cada tipo tiene un propósito distinto y tiene un ciclo de vida diferente que define cómo se crea y cómo se destruye el componente.
- Los cuatro componentes son:
 - Actividades
 - Servicios
 - Receptores de Broadcast
 - Proveedores de contenido



Android: Conceptos Básicos

Componentes básicos

- Actividad (**Activity**):
 - Representa una pantalla independiente con una interfaz de usuario. A pesar de que nuestra aplicación dispondrá de múltiples pantallas interconectadas entre sí, nosotros deberemos generarlas individual e independientemente (pudiendo pasar datos entre ellas, en caso de ser necesario).
 - Las actividades se implementan como una subclase de **Activity**.
 - Más info:
<http://developer.android.com/guide/components/activities.html>



Android: Conceptos Básicos

Componentes básicos

- Servicio (**Service**):
 - Es un componente que corre de fondo para hacer operaciones de larga duración o trabajo en procesos remotos. Contrario a la actividad, no dispone de interfaz gráfica.
 - Otro componente, como una actividad, puede iniciar un servicio y dejar que se ejecute o conectarse con el fin de interactuar con él.
 - Los servicios se implementan como una subclase de **Service**.
 - Más info:
<http://developer.android.com/guide/components/services.html>



Android: Conceptos Básicos

Componentes básicos

- Proveedor de contenidos (**Content Provider**):
 - Un proveedor de contenido gestiona un conjunto compartido de datos de aplicaciones.
 - Una aplicación puede almacenar los datos en el sistema de archivos, una base de datos SQLite, en la web, o cualquier otro lugar de almacenamiento permanente al que tenga acceso. Sólo a través de un proveedor de contenido otras aplicaciones podrán consultar o incluso modificar los datos (si el proveedor de contenido lo permite).
 - Los proveedores de contenido se implementan como una subclase de **ContentProvider** y deben implementar un conjunto estándar de API's que permiten a otras aplicaciones realizar transacciones.
 - Más info:
<http://developer.android.com/guide/topics/providers/content-providers.html>



Android: Conceptos Básicos

Componentes básicos

- Receptor de difusión (**Broadcast Receiver**):
 - El receptor de difusión nos permite responder a anuncios difundidos por todo el sistema.
 - Muchas emisiones se originan en el sistema, por ejemplo, una anunciar que la pantalla se ha apagado, la batería está baja, o una imagen fue capturada.
 - Las aplicaciones también pueden iniciar las emisiones, por ejemplo, informar a otras apps de que algunos datos se han descargado al dispositivo y están disponibles para que los utilicen.
 - No muestran una interfaz de usuario, pero es posible crear una notificación en la barra de estado para alertar al usuario cuando se produce un evento de difusión.
 - Habitualmente son una "puerta de entrada" a otros componentes y tienen la función de hacer una cantidad muy mínima de trabajo. Por ejemplo, iniciar un servicio para realizar algún trabajo basado en el evento.



Android: Conceptos Básicos

Activando componentes

- Un aspecto único del diseño del sistema Android es que cualquier aplicación se puede iniciar el componente de otra aplicación.
 - Por ejemplo, si desea que el usuario capturar una fotografía con la cámara del dispositivo, es probable que haya otra aplicación que hace eso y que su aplicación pueda utilizarlo, en lugar de desarrollar una actividad para capturar una fotografía por sí mismo.
 - No es necesario incorporar o incluso enlazar con el código de la aplicación de la cámara. En lugar de ello, sólo tiene que iniciar la actividad en la aplicación de la cámara que captura una foto.
 - Cuando se haya completado, la foto es devuelta a la aplicación para que pueda utilizarla. Para el usuario, parece como si la cámara es en realidad una parte de su aplicación.



Android: Conceptos Básicos

Activando componentes

- Cuando el sistema inicia un componente, se inicia el proceso para la aplicación (si no está ya en ejecución) y se crea una instancia de las clases necesarias para el componente.
 - Por ejemplo, si nuestra aplicación inicia la actividad que captura una foto en la aplicación de la cámara, la actividad se ejecuta en el proceso que pertenece a la aplicación de la cámara, no en el proceso de nuestra aplicación.
- A diferencia de las aplicaciones en la mayoría de los sistemas, las aplicaciones de Android no tienen un solo punto de entrada (no hay función `main()`, por ejemplo).



Android: Conceptos Básicos

Activando componentes

- Debido a que el sistema ejecuta cada aplicación en un proceso separado con permisos de archivos que restringen el acceso a otras aplicaciones, una aplicación no puede activar directamente un componente de otra aplicación.
- El sistema Android, sin embargo, si que puede. Por lo tanto, para activar un componente de otra aplicación, una aplicación debe entregar un mensaje al sistema que especifica su intención de iniciar un componente particular y el sistema activa el componente por delegación.



Android: Conceptos Básicos

Activando componentes

- Tres de los cuatro tipos de componentes básicos -actividades, servicios y receptores de difusión - son activados por un mensaje asíncrono llamado **Intent**.
- Las intenciones unen componentes individuales entre sí en tiempo de ejecución (se puede pensar en ellos como los mensajeros que solicitan una acción de otros componentes), tanto si el componente pertenece a la misma aplicación o a otra diferente.
- Un intento se crea con un objeto **Intent**, que define un mensaje para activar ya sea un componente específico o un tipo específico de componente (una intención puede ser explícita o implícita, respectivamente).



Android: Conceptos Básicos

Activando componentes

- Para las actividades y servicios, una intención define la acción a realizar (por ejemplo, "ver" o "enviar" algo) y puede especificar el URI de los datos sobre los que actuar (entre otras cosas que el componente que se está inicializando necesite saber).
 - Por ejemplo, una intención podría transmitir una petición a una actividad para mostrar una imagen o abrir una página web.
- En algunos casos, se puede iniciar una actividad para recibir un resultado, en cuyo caso, la actividad también devuelve el resultado en una intención.
 - Por ejemplo, puede emitir una intención para que el usuario elija un contacto personal y la intención de retorno incluye un URI que apunta al contacto elegido.



Android: Conceptos Básicos

Activando componentes

- Para los receptores de difusión, una intención simplemente define el anuncio que se está transmitiendo (por ejemplo, una emisión para indicar que la batería del dispositivo está baja incluye sólo una cadena de acción conocida que indica "*battery is low*").
- Las intenciones se implementan como una subclase de **Intent**
- Más info:
<http://developer.android.com/reference/android/content/Intent.html>



Android: Conceptos Básicos

Activando componentes

- El último tipo de componente, el proveedor de contenido, no se activa por medio de intenciones. Se activa cuando es el objetivo de una solicitud de un [ContentResolver](#).
- Un sistema de resolución contenido maneja todas las transacciones directas con el proveedor de contenido de modo que el componente que ha de realizar transacciones con el proveedor lo hace invocando métodos del objeto [ContentResolver](#).
- Esto deja una capa de abstracción entre el proveedor de contenidos y la información que solicita el componente (por seguridad).



Android: Conceptos Básicos

Componentes básicos



Android: Conceptos Básicos

Más Información

- **Documentación adicional en**

- **Android.com** <http://developer.android.com/>

- Descargas
 - Guías de Instalación
 - Tutoriales
 - Doc. de Referencia
 - <http://www.android-spa.com/> (en español)

- **Foros**

- <http://androidcommunity.com/>
 - <http://www.mail-archive.com/android-beginners@googlegroups.com/>
 - <http://stackoverflow.com/questions/tagged/android>



Samsung **TECH INSTITUTE**