Fundamentos de la plataforma Android y bloques básicos de una aplicación

*Adaptado de: http://www.clasepro.com/fundamentos-plataforma-android/*

Kernel de Linux

En la base tenemos el kernel 2.6 de Linux, Android lo utiliza por su robustez demostrada y por la implementación de funciones básicas para cualquier sistema operativo, por ejemplo: seguridad, administración de memoria y procesos, implementación de conectividad de red (*network stack*) y varios interpretes (*drivers*) para comunicación con los dispositivos físicos(hardware).  
  
**[](http://www.maestrosdelweb.com/images/2011/03/1_arquitectura_android.jpg)**  
  
Android utiliza como base el kernel de Linux pero los dos sistemas no son lo mismo, Android no cuenta con un sistema nativo de ventanas de Linux ni tiene soporte para glibc (libería estándar de C) ni tampoco es posible utilizar la mayoría de aplicaciones de GNU de Linux.  
  
Además de todo lo ya implementado en el kernel de Linux, Android agrega algunas cosas específicas para plataformas móviles como la comunicación entre procesos (lograda a través del binder), la forma de manejar la memoria compartida (ashmem) y la administración de energía (con wakelocks). De las características únicas del kernel utilizado por Android encuentran más información en **[Android Kernel Features](http://elinux.org/Android_Kernel_Features" \t "_blank)**.

Librerías y ejecución

Sobre el kernel, tenemos un conjunto de librerías de C y C++ utilizadas por el sistema para varios fines como el manejo de la pantalla (surface manager), mapas de bits y tipos de letra (Free Type), gráficas en 2D y 3D (SGL y OpenGL), manejo de multimedia (Media Framework), almacenamiento de datos (SQLite) y un motor para las vistas web y el navegador (WebKit).  
  
Junto a estas librerías, encontramos lo necesario para la ejecución de las aplicaciones a través de la máquina virtual Dalvik. Cada aplicación utiliza una instancia de la máquina virtual ejecutando un archivo DEX (Dalvik Executable) y el sistema está optimizado para que se ejecuten múltiples instancias de la máquina virtual. Se desarrolla en java pero no se utiliza una máquina virtual de Sun para su ejecución ni tampoco archivos CLASS.  
  
**[](http://www.maestrosdelweb.com/images/2011/03/2_arquitectura_android.jpg)**

Estructura de aplicaciones

Sobre las librerías encontramos una estructura que nos brinda un contexto para desarrollar, este *framework* permite a los desarrolladores aprovechar un sistema de vistas ya construido, administrar notificaciones y acessar datos a través de proveedores de contenido entre otras cosas.  
  
**[](http://www.maestrosdelweb.com/images/2011/03/3_arquitectura_android.jpg)**

Aplicaciones

Las aplicaciones centrales que incluye el sistema por defecto son: teléfono, navegador, manejo de contactos, etc. En esta capa de la arquitectura es donde trabajaremos desarrollando aplicaciones.  
  
**[http://www.maestrosdelweb.com/images/2011/03/4_arquitectura_android-450x44.jpg](http://www.maestrosdelweb.com/images/2011/03/4_arquitectura_android.jpg)**

Bloques básicos de una aplicación

Una vez vista la arquitectura, empezaremos con lo fundamental para desarrollar una aplicación. Los componentes básicos de una aplicación son: **activities**, **intents**, **views**,**services**, **content providers** y **broadcast receivers**. Si dominan bien estos términos pueden saltar directo al código.  
  
**[](http://www.maestrosdelweb.com/images/2011/03/5_bloques_basicos.jpg)**  
  
**-   Activities:**son componentes de la interfaz que corresponde a una pantalla. Podemos visualizarlo como un mazo de cartas en el que tenemos varias cartas pero solamente una está hasta arriba. Una aplicación para una lista de cosas por hacer (remember the milk) puede tener una actividad para ingresar las cosas por hacer y otra actividad para mostrar el listado, en conjunto estas actividades conforman la aplicación.  
  
**-   Intents:** son mensajes que provocan notificaciones o cambios de estatus,  que al ser recibidos por actividades o servicios pueden levantar procesos. De esta forma se unen componentes dentro de la misma aplicación o de diferentes aplicaciones.  
  
**-   Views:**  son los componentes de la interfaz de usuario, diferentes vistas pueden agruparse a través de grupos logrando una jerarquía, esto se logra a través de la disposición de los componentes a través de un archivo XML.  
  
**-   Services:** son componentes que ejecutan operaciones en segundo plano y no tienen una interfaz de usuario. Por ejemplo, al escuchar música, hay un servicio encargado de la reproducción que se ejecuta de fondo y la aplicación que manipulamos le manda mensajes a este servicio diciéndole que se detenga, pause o reproduzca la siguiente canción.  
  
**-   Content Providers:** representan la abstracción para almacenar y obtener datos permanentes e incluso entre aplicaciones diferentes. El sistema incluye algunos proveedores de contenido útiles (audio, video, etc) y además pueden desarrollarse nuevos.  
  
**-   Manifest:**El archivo AndroidManifest.xml es donde se configura la aplicación, se agregan actividades, se asignan permisos, etc.  
  
**-   Broadcast Receivers:**son componentes que responden a avisos y anuncios de difusión (broadcast). Estos avisos provienen del sistema (batería baja, una llamada entrante, etc) y de aplicaciones (pasando avisos de una aplicación a otra). Aun que no muestran una interfaz de usuario algunas veces utilizan barras de progreso para mostrar avances. Estos se activan a través de mensajes asincrónicos llamados intents (mencionados arriba).