

# ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

---

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

Lenguajes de Programación  
2012 — II Término

Ana Arias - acarias@espol.edu.ec  
Liliana Ramos - ljramos@espol.edu.ec  
Denny Schuldt - dschuldt@espol.edu.ec

## Observaciones, Conclusiones y Experiencias del Desarrollo

# Comic It!

## Objetivo General

Dar a conocer todas las experiencias vividas durante el desarrollo de nuestra aplicación para el sistema operativo Android.

## Observaciones

- a Android es un sistema operativo muy curioso. Trabaja con un sistema de capas que está conformado por xmls (layouts), imagenes (drawables) y demás recursos (values). Los controladores (clases) son programados en Java, y extienden de una super clase llamada Activity. Esto fue un gran reto para nosotros, puesto que tuvimos que estudiar un nuevo API, así como la estructura de Android.
- b Nuestro proyecto incluye muchas funcionalidades básicas del teléfono en el que se lo vaya a instalar. Usamos lo que es la cámara, la galería y arrastrar y soltar. En particular, esta última fue la que se nos hizo muy complicada, puesto que no solo queríamos que se arrastre una imagen, sino que también queríamos que se redimensione (zoom). Aún estamos trabajando en aquello para que quede perfecto.

## Experiencias

- Una de las experiencias o anécdotas que podemos contar, es acerca del manejo de los Layouts. Fue muy complicado trabajar con ellos. En especial cuando usabamos el drag and drop con el zoom.

Como ya deberíamos saber, cuando se agrega algo a una ventana, este tiene su nivel de profundidad en el eje de las z. Cada una está encima de otra. Cuando trabajamos con estos Layouts, opacaban aquellos componentes que estaban en un nivel inferior, haciendo que estos no puedan ser seleccionados. Fue muy cómico como nos fijamos en donde estaba el error, porque probabamos varias veces con diferentes ideas y ninguna nos resultaba.

- Uno de los mayores inconvenientes que tuvimos fue con la generación del R. El R.java, es una clase generada por android donde se inicializa sus variables contenidos en los xml. Este archivo es de suma importancia para poder compilar la aplicacion y generar el apk para poder instalarlo en el dispositivo.

Eclipse y el R, a nuestra apreciación, no se llevan muy bien. Eclipse no lo generaba correctamente cada vez que debía y teníamos que hacer debug y muchas otras maravillas para poder generarlo.

- Decidirnos por el IDE a utilizar fue todo un reto, ya que estábamos acostumbrados a Netbeans. Al principio tuvimos varios conflictos con las versiones de los proyectos, pues nos dimos cuenta que no era lo mismo trabajar en NetBeans que en Eclipse, este último organiza sus carpetas de manera diferente. Una de las vetajas de usar Eclipse es la paleta de elementos (Palette), ya que editar layouts es mucho más fácil allí que en Netbeans.

Lo negativo (muy negativo) de Eclipse al desarrollar un proyecto, es la gran cantidad de errores que puede generar, y muchas veces errores ocasionados por el mismo IDE. (R es uno de ellos, probablemente el más grande). Para ejecutar una aplicación, en ciertas ocasiones era necesario actualizar el proyecto, y así eliminar errores.

- Utilizar un emulador en Eclipse puede no ser tan fácil, dependiendo del dispositivo de prueba. En nuestro caso, al principio fue fácil, ya que utilizabamos un emulador general. Pero luego, al tener el dispositivo de prueba, era necesario adaptar el emulador a nuestras necesidades. Aquí surge el error "Failed to allocate memory: 8", probablemente el segundo error más importante, luego de R.

Luego de investigar mucho, y luego de muchos intentos fallidos, encontramos una página

en japonés, en donde explicaban cómo solucionar el error. El idioma no fue un problema, pues una imagen puede decir mucho, y la guía estaba muy bien ilustrada.

## **Conclusiones**

- 1 Para poder desarrollar en android, es necesario tener nociones básicas de archivos xml y saber programar en Java. En general, hay que tener un buen nivel de abstracción.
- 2 Siempre debemos instalar todos los APIs que android nos ofrece para no tener problemas con los emuladores o con las pruebas en los teléfonos. Haciendo esto, también estamos generando una mejor aplicación con muchas más características.
- 3 Todos los miembros del grupo deberían trabajar con el mismo IDE, para así evitar conflictos al momento de desarrollar.
- 4 Paciencia y dedicación, componentes claves para el desarrollo de un proyecto en Android.