

2011

# Ingeniería inversa básica en Android



Ghost

[portalhacknet@gmail.com](mailto:portalhacknet@gmail.com)

07/07/2011

## TABLA DE CONTENIDO

1. Introducción.
2. Formatos de archivo .Dex y .JAR.
3. Herramientas necesarias.
4. Configurar Dex2Jar en Eclipse.
5. Descompilar archivos .JAR.
6. Agradecimientos.

## 1. Introducción

La Wikipedia (Enciclopedia en línea) define la ingeniería inversa como un proceso cuyo objetivo es obtener información o un diseño base de la constitución de un producto público de tal manera que también podamos saber de que está hecho, que lo hace funcionar y como fue fabricado.

Sabiendo la definición anterior muchas empresas usan este método para estudiar a su competencia, por lo que **ingeniería inversa** podemos hacerla a un simple caramelo o incluso a un dispositivo electrónico.

El software no se escapa de este proceso, por lo que si tenemos un poco de recorrido en esto recordaremos el popular software alemán **OlllyDBG**, el cual nos permite ver en código ensamblador la constitución interna de un programa, de tal manera que podamos dar con su funcionamiento y posteriormente obtener beneficios de este estudio.

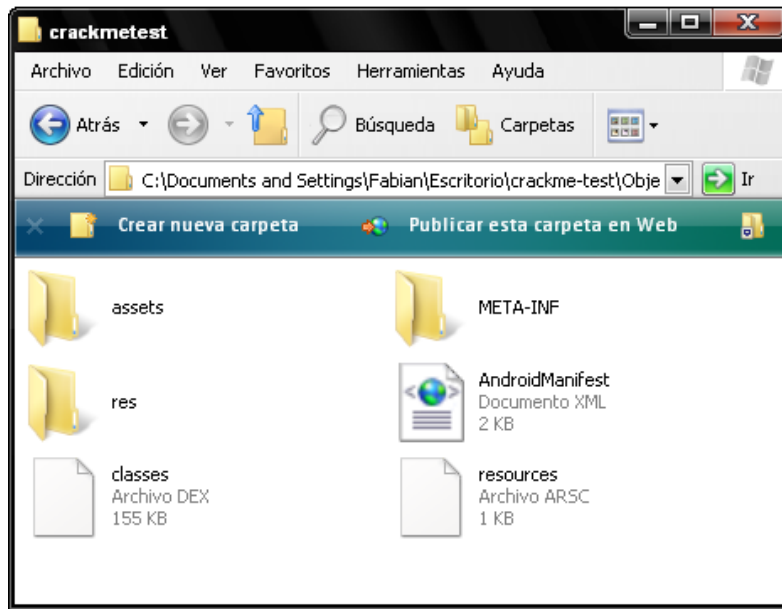
En la mayoría de casos (y en la práctica), hacerle ingeniería inversa a un programa de computadora tiene el fin de obtener un nombre de usuario o serial para poder usar el software sin limitaciones (sabiendo que es software propietario o comercial y requiere de algún tipo de activación).

Esta pequeña guía básica nos orientará en el camino de la **ingeniería inversa en el sistema operativo Android**, para lo cual conoceremos herramientas y una pequeña parte del proceso a seguir.

## 2. Formatos de archivo .DEX y .JAR

Como sabemos la mayoría de **aplicaciones para Android** están programadas en Java, los ejecutables de este sistema operativo tienen una extensión de tipo .APK, hasta aquí todo bien, pero debemos saber que esta extensión no es más que una variación de la extensión .JAR, por lo que si alguna vez hemos programado en Java sabemos que los .JAR funcionan como una especie de contenedor con varios archivos adentro, por lo que con cualquier compresor podemos extraer su contenido y ver una serie de archivos que son los que componen la aplicación.

Por lo tanto, si a una aplicación Android (.APK) le extraemos los ficheros con un compresor cualquiera podremos ver todos los ficheros que componen la aplicación.



Uno de estos archivos es **classes.dex** que será uno de los más importantes pues este es el que contiene las clases y todo el código en general de la aplicación.

### 3. Herramientas necesarias.

Como mencionábamos más arriba, en Windows para poder ver el código en ensamblador de un archivo con extensión .exe usábamos el OllyDB, en esta ocasión haremos uso de una serie de herramientas que harán lo mismo y con esto podremos ver el código Java.

El proceso será sencillo, pasaremos el archivo .dex a .jar y luego descompilaremos este último. Para esto necesitaremos:

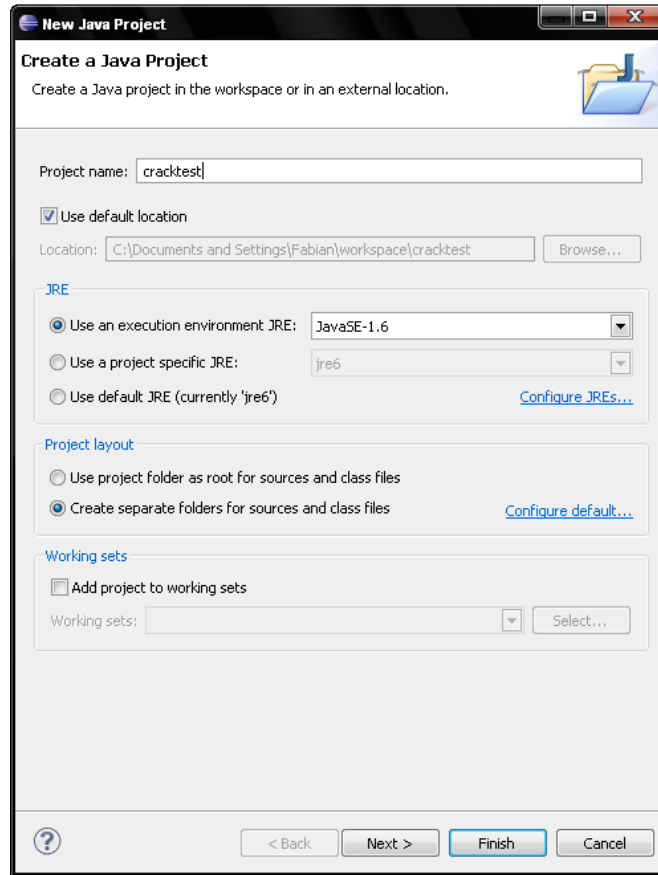
**JD-GUI:** Es una aplicación con interfaz gráfica muy intuitiva con la que podremos ver el código fuente de los archivos .class (básicamente .jar de Java). Con esta herramienta podremos ver clases, métodos y otros componentes del programa. Web oficial: <http://java.decompiler.free.fr/?q=idgui>

**Dex2Jar:** Como su nombre nos lo dice, permite pasar de .dex a .jar, podemos usarlo individualmente o en conjunto con las herramientas que nombraremos aquí. Web oficial: <http://code.google.com/p/dex2jar/>

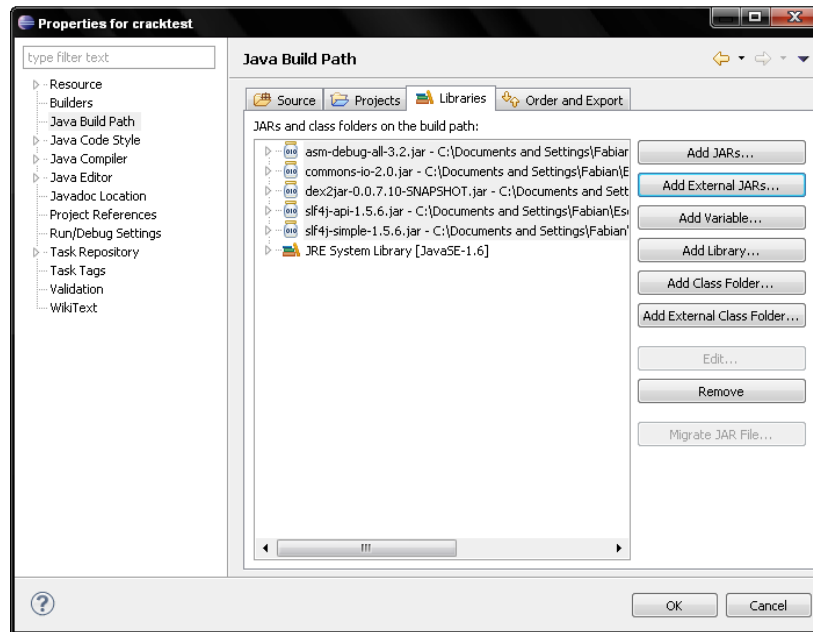
**Eclipse:** Es un IDE de desarrollo para varias plataformas y que funciona con java. Web oficial: <http://www.eclipse.org/downloads/>

#### 4. Configurar .Dex2Jar en Eclipse

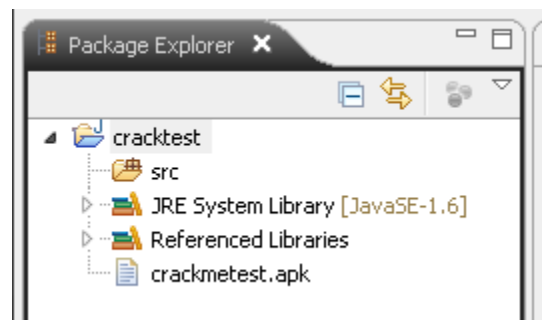
Lo primero será configurar un nuevo proyecto, por lo que vamos a File > New > Java Project, le ponemos un nombre al proyecto y damos a finalizar.



Seleccionamos nuestro proyecto en explorador de paquetes, luego estando seleccionado vamos al menú Project > Properties > Java Build Path. Allí en la pestaña “Libraries” damos click en “Add external Jars”, allí en la ventana que sale ubicamos la carpeta de Dex2Jar y añadimos todos los .Jar de la carpeta a nuestro proyecto y seleccionamos Ok.

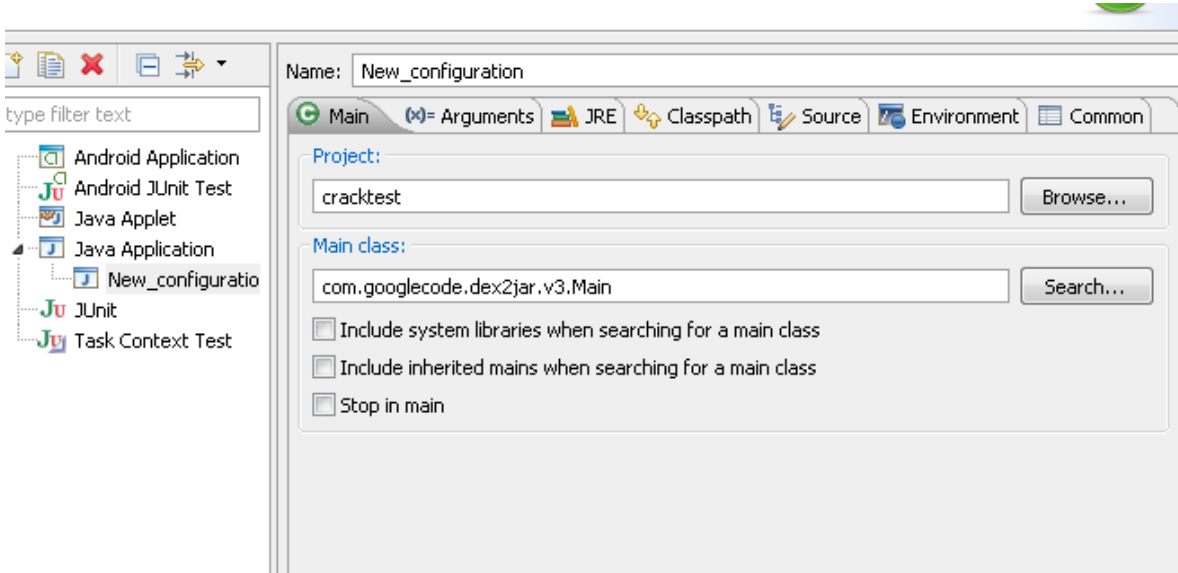


Ahora añadimos nuestro fichero .APK al proyecto, es decir, lo arrastramos hasta el explorador de paquetes

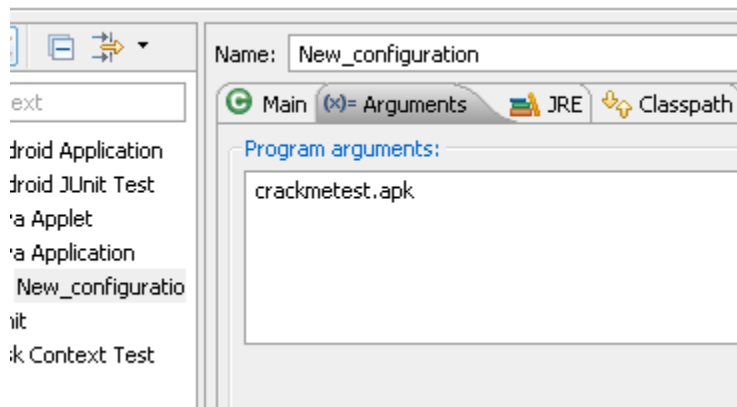


Ahora hacemos click derecho en la carpeta de nuestro proyecto, seleccionamos Run As > Run Configurations y hacemos doble click en “Java application”, por lo que se añadirá una especie de página en la cual podemos pasar argumentos.

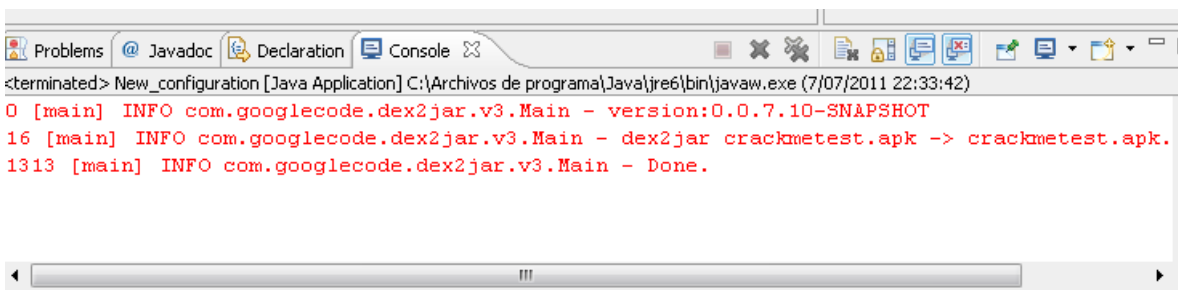
Aquí en la pestaña “Main” añadimos el nombre de la clase Main de Dex2Jar, la cual es posible encontrarla haciendo click en el botón “search” de la misma pestaña, en mi caso tiene una estructura así: **com.googlecode.dex2jar.v3.Main**



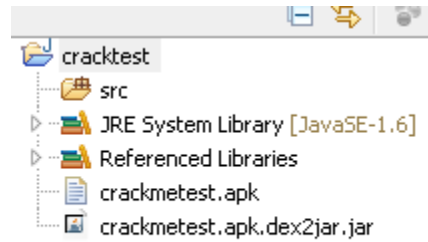
Luego vamos a la pestaña arguments y en la primera caja de texto colocamos el nombre de nuestro fichero APK con la extensión:



Damos click en Aplicar (Apply) y luego en "Run", para ver un resultado de este tipo:

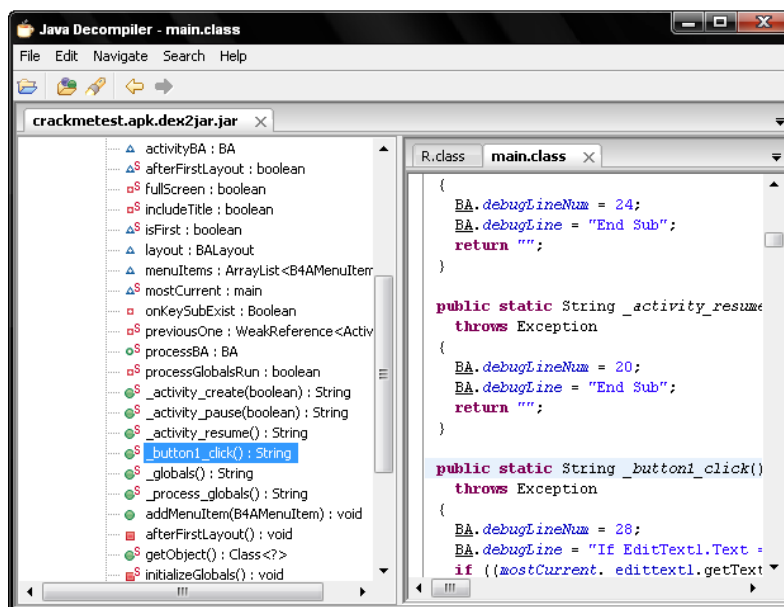


Ahora actualizamos nuestro explorador de proyectos (Click derecho y luego le damos “Refresh”) y veremos un nuevo fichero con extensión .Jar, es el que necesitábamos.



## 5. Descompilar archivos .JAR

Con el proceso anterior obtuvimos un .JAR limpio, ahora debemos “descompilarlo” para así poder ver su código fuente, por lo que abrimos el JD-GUI y allí abrimos nuestro .JAR generado, lo primero que observo son los métodos y en especial uno que dice `button1_click()`, ya que de seguro allí habrá algo.



Pero bueno, ya a partir de aquí podemos seguir e investigar todo el código, obviamente los conocimientos de programación nunca están demás para entender los algoritmos implicados en la aplicación.



## 6. Agradecimientos

Como siempre en mis humildes guías me gusta agradecer de las personas de las que aprendo y admiro, aprovechando que conocí a grandes personajes en el último Campus Party ahí voy:

oooooooo (Radical), UrbaN77, SmartGenius, D-M-K, Monje (Clerigo) y a toda la comunidad de Red Informática Colombiana (RIC) en general ([foro.redinfocol.org](http://foro.redinfocol.org)).

###

#Copyleft (?) Ningún derecho reservado. 2011.

#Comentarios a: [portalhacknet@gmail.com](mailto:portalhacknet@gmail.com)

#By Ghost