



Estudio Comparativo de dos Plataformas de Programación de Dispositivos Móviles

Gregorio Elías Pazmiño Vélez ⁽¹⁾ Magdeline Estefanie Rosero Pérez ⁽²⁾

Facultad de Ingeniería en Electricidad y Computación

Escuela Superior Politécnica Del Litoral

Km 30.5 vía Perimetral, Código Postal: 09-01-5863, Guayaquil, Ecuador

grelpazm@espol.edu.ec (1), magerose@espol.edu.ec (2)

Resumen

Este documento presenta un análisis comparativo entre dos plataformas de desarrollo de programación de dispositivos móviles, con el objetivo de ayudar a un programador a elegir que plataforma es la mejor opción para comenzar a desarrollar sus aplicaciones, tomando en cuenta las ganancias monetarias, el tiempo de desarrollo y la facilidad en la programación. Las plataformas seleccionadas son Android y JME, dos de las más usadas a nivel global.

Para la comparación se consideraron ocho variables: APIs, licencia, líneas de código, lenguajes de programación, tamaño de la aplicación, emulación, conectividad y múltiples pantallas. Basados en los resultados obtenidos por la comparación entre Android y JME en cada variable, se exponen las conclusiones y las recomendaciones basadas en los resultados obtenidos.

Abstract

This document presents a comparative analysis between two develop mobile platform, with a purpose to help to app developer which platform is the best choice to start developing his/her applications, thinking about gains of coins, developing time and property of easily programming. The two platforms chosen were Android and JME, two of the plattform most used by people.

Eight variables were analized for platforms comparison: APIs, license, lines of code, language programming, application size, emulator, connectivity and variety of screen. Finally, this paper shows the conclusions and recommendations based on the results to compare Android and JME in each variable.



1. Introducción

Este proyecto presenta una comparativa entre dos plataformas de programación de dispositivos móviles, aportando una solución para resolver el problema que tiene el programador al momento de seleccionar con cual plataforma debe comenzar a crear sus aplicaciones, tomando en mayor consideración, las ganancias generadas por la compra de la aplicación, el tiempo y la facilidad en que el desarrollador realiza el programa.

En la actualidad existen varias empresas que se dedican a crear dispositivos móviles junto con su sistema operativo para el correcto funcionamiento del mismo, tales como BlackBerry OS, Android, Windows phone, entre otros. Debido a la gran variedad de estos dispositivos (celulares, tablets, PDAs) existen múltiples plataformas en donde un programador de dispositivos móviles pueda elegir en donde realizar sus aplicaciones. Esto presenta un problema, ya que el simple hecho de no haber escogido la plataforma correcta implicaría pérdida de tiempo y como resultado de ello, pérdida de dinero.

Las dos plataformas comparadas fueron Android y JME, dos de las más usadas en la actualidad y sin inversión alguna para comenzar a desarrollar las aplicaciones.

El propósito del proyecto es implementar, comparar y analizar dos plataformas de programación de dispositivos móviles para conocer que plataforma es la mejor opción para comenzar a desarrollar aplicaciones móviles.

2. Comparación entre las plataformas Android y JME

Se comparan las plataformas escogidas para determinar en cada variable, cuál de los dos, Android o JME, presenta ventaja con respecto al otro. Las variables analizadas son: APIs, licencia, lenguaje de programación, líneas de código, tamaño de la aplicación, emulación, conectividad y múltiples pantallas.

2.1. APIs

Se compara la cantidad de librerías que tienen las plataformas a disposición del programador. En la figura 1 se muestra la cantidad de APIs que tiene Android y JME.

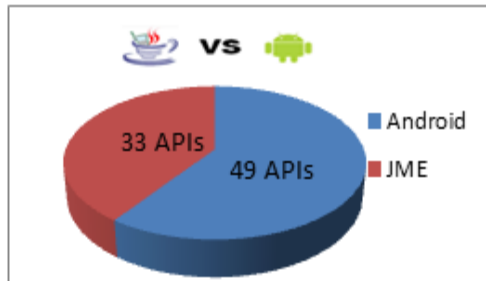


Figura 2.1.- Cantidad de APIs de Android y JME.

Android cuenta con más recursos que pueden ser agregados para desarrollar una aplicación. [1][2]

2.2. Licencia

Se compara el tipo de licencia que posee Android y JME. La plataforma JME posee licencia GPL es decir que le permite al usuario hacer uso de la plataforma, modificarla y distribuir las modificaciones que se le haya hecho a la misma. La licencia GPL establece que todo programa o cualquier material que haya sido desarrollado bajo esta licencia debe estar disponible para ser compartido con el resto de personas.

Android posee dos licencias: Apache 2.0 y GNU GPL 2 es decir que la persona está en la libertad de usar, modificar y distribuir el software sin estar obligado a liberar el código

fuelle que este modificado. Por otro lado la licencia Apache 2.0 también indica que los programas derivados no necesariamente deben ser de este mismo tipo de licencia, ni siquiera exige que el software se tenga que distribuir libremente y con código abierto (se puede comercializar, lo que sugiere un gran beneficio), pero exige que se realice un informe a los receptores que en la distribución se ha usado código con licencia Apache. [3]

2.3. Lenguajes de Programación

Las aplicaciones para Android pueden ser desarrolladas sobre varios lenguajes de programación, los cuales son: Java, VisualBasic, C# y si se desea programar de manera nativa se la puede hacer en C/C++. Existe una plataforma para desarrollar aplicaciones para Android sin la necesidad de saber algo de programación y esta se llama AppInventor. Mientras que para desarrollar aplicaciones en JME se usa un único lenguaje el cual es java para dispositivos móviles. En la tabla 1 se muestra los diferentes lenguajes programación que permiten desarrollar aplicaciones para Android y para JME. [4]



Tabla I.- Lenguajes de Programación

Lenguajes de Programación	
Android	JME
Lenguaje Java para Android	Lenguaje Java para JME
Lenguaje VisualBasic para Android	
Lenguaje C# para Android	
Lenguaje C/C++ para Android	

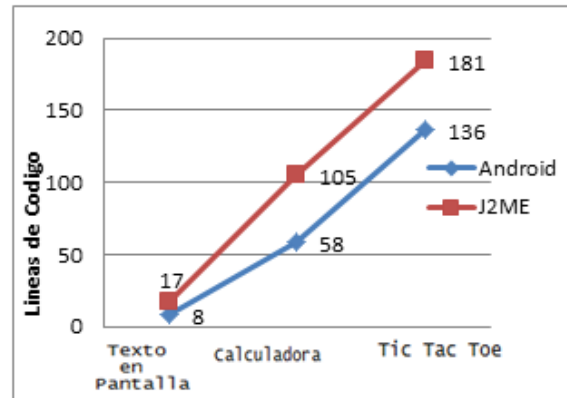


Figura 2.2.- Comparativa Android y J2ME: Líneas de código

2.4. Líneas de Código

Las líneas de código fueron comparadas mediante tres aplicaciones desarrolladas por nosotros: texto en pantalla (imprime un texto en la consola), calculadora básica y tic tac toe. El lenguaje en el que se desarrollaron las aplicaciones para Android fue Java. La figura 2 indica que para JME realizar una aplicación que solo muestre un texto en la consola se llevó 17 líneas de código, en comparación con 8 líneas de código en Android. Con la aplicación más básica se puede observar la diferencia entre programar en Android y en JME.

2.5. Tamaño de la aplicación

La comparación del tamaño de las aplicaciones se las realiza observando el peso que presentan en los teléfonos celulares y tomando como unidad de medida el kiloByte. Con la comparación de 7 aplicaciones (3 hechas por nosotros y 4 descargadas desde internet) se observó que los programas hechos en JME tienen un peso menor que las aplicaciones hechas en Android. En la figura 3 se presentan los pesos de cada aplicación.

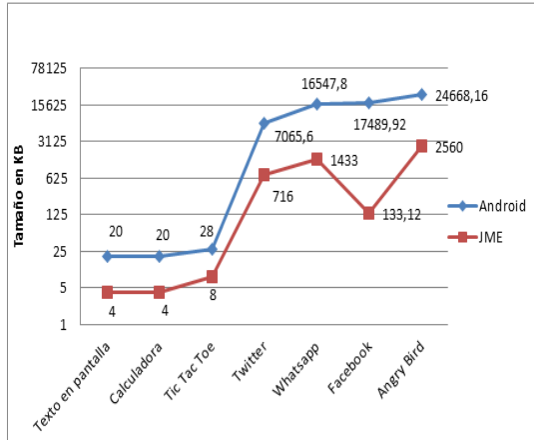


Figura 3.- Comparación del tamaño de las aplicaciones

2.6. Emulación

Los emuladores forman una parte esencial al momento de ejecutar un programa y ver el comportamiento del mismo tal como si fuera en un dispositivo real. En esta variable se comparó el tiempo promedio que tardó el emulador en ejecutar una aplicación y la tendencia con la que pueden ir variando los datos por encima o por debajo del tiempo promedio, dando a conocer la exactitud y precisión que tienen los emuladores.

La ejecución de los emuladores se realizó bajo las mismas condiciones; en la misma PC con un sistema operativo Windows. La figura 4 muestra el tiempo promedio en cada emulador. En Android 15.703 seg y JME 27.175 seg. Lo que indica que el emulador de Android ejecuta las aplicaciones a mayor

velocidad que el emulador de JME (Nokia SDK para java emulador).

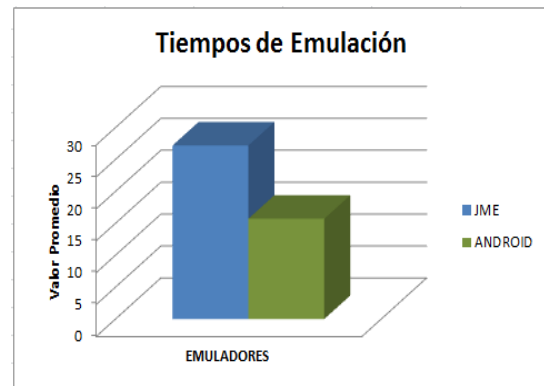


Figura 4.- Tiempo promedio de los emuladores.

Las figuras 5 y 6 muestran la dispersión de los datos en el emulador de JME y en el emulador de Android respectivamente, indicando que el emulador de Android es más preciso que el emulador de JME.

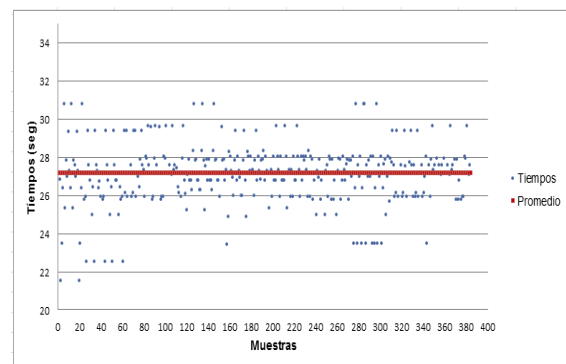


Figura 5.- Dispersión de datos en JME

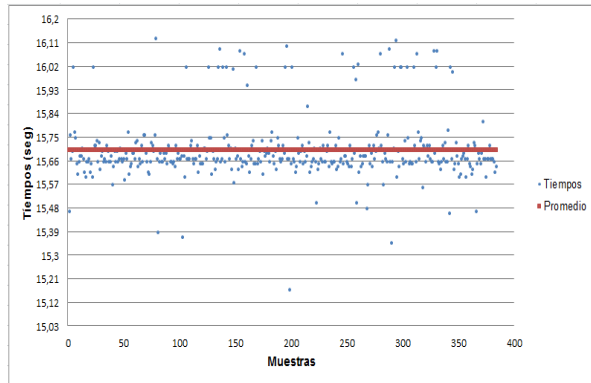


Figura 6.- Dispersión de datos en Android.

2.7. Conectividad

Con esta variable conocemos cuantos tipos de conectividad son soportados por las plataformas Android y JME. Tal como se muestra en la tabla 2, Android soporta más tipos de conectividad que la plataforma JME. Con Android una persona es capaz de realizar una aplicación cuando el dispositivo móvil detecta una red Wimax, es capaz de conectarse con dispositivos de baja potencia como teclado, joystick o un mouse. Tecnologías que pueden llamar la atención a los desarrolladores y vean factible crear aplicaciones que sean de agrado y de utilidad para el usuario. En la tabla 2 se muestra los diferentes tipos de conectividades que soportan estas dos plataformas [6]

Tabla II.- Tipos de conectividad

Android	JME
Wi-Fi	Wi-Fi
GSM/EDGE	GSM/EDGE
NFC	NFC
Bluetooth	Bluetooth
GPRS	GPRS
Infrarrojo	Infrarrojo
Wimax	
CDMA	
ED-VO	
LTE	

2.8. Múltiples pantallas

Android se ejecuta en una variedad de dispositivos que ofrecen diferentes tamaños de pantalla y densidades. La plataforma Android ofrece la herramienta de apoyo de múltiples pantallas para permitirle al programador crear su aplicación de tal manera que con solo un archivo .apk pueda abarcar todas las configuraciones de pantallas compatibles en lugar de programar una aplicación para cada tipo de pantalla. Las variables que se toman en cuenta al momento de diseñar la interfaz de usuario para la plataforma Android son las siguientes:



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL CENTRO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA



Tamaño, densidad, orientación, resolución y píxel independiente de la densidad.[7]

La plataforma JME está enfocada hacia teléfonos móviles pequeños por lo cual esta plataforma no presenta una diversidad de pantallas como Android, los métodos que utiliza para obtener el tamaño de pantalla de un dispositivo son `getWidth()` y `getHeight()` los cuales permiten al programador ajustar los elementos utilizados en la aplicación al tamaño de pantalla de un teléfono en particular, pero si se desea adaptar el tamaño de una imagen para cualquier tipo de pantalla lo que utiliza JME es el método de redimensión de imagen `private Image resizeImage(Image src)`. [8]

3. Conclusiones

Con los resultados obtenidos en cada variable tras comparar las dos plataformas, podemos observar que Android es superior en la mayoría de las variables comparadas, posee mayor conjunto de funcionalidades de la cual un desarrollador de aplicaciones móviles pueda optar para crear una aplicación más llamativa para el usuario.

En la prueba de los emuladores de Android y de JME se obtuvo que el emulador de Android es más rápido en un 43% que el tiempo en que el emulador de JME demora al cargar las aplicaciones, cabe recalcar que las pruebas fueron hechas bajo las mismas condiciones.

JME fue superior en la variable tamaño de la aplicación, esto se debe en parte a que Android requiere de un gran conjunto de librerías para poder funcionar, aparte de que por defecto la aplicación se compila con recursos innecesarios, un ejemplo de ello son las carpetas *drawable* que se encuentran dentro de la carpeta *res*, se pueden eliminar si la aplicación va dirigida a un conjunto específico de dispositivos móviles, es decir el peso de la aplicación no depende en gran medida de las líneas de código que se toma para el desarrollo del programa. Los programas hechos en Android toman menos líneas de código que las aplicaciones hechas en JME, resultado visible al realizar tres aplicaciones sencillas.

4. Agradecimientos

Queremos agradecer encarecidamente al Ing. Marcos Millán y a la Ing. Patricia Chávez por ayudarnos en la revisión del artículo.



5. Referencias

- [1] Anónimo (2013) Developers [online] Disponible en <http://developer.android.com/reference/android/nfc/package-summary.html> (04-12-2013)
- [2] Desarrolladores de Nokia (2013) Nokia Developer [Online] Disponible en <http://developer.nokia.com/Resources/Library/Java/#!supported-apis.html> (28-11-2013).
- [3] Muro, L. (2013) Monografías J2ME [Online] Disponible en <http://www.monografias.com/trabajos55/licencias-de-software/licencias-de-software.shtml> (27-06-2013)
- [4] León, J. (2012) Androideity [Online] Disponible en <http://androideity.com/2012/07/16/5-lenguajes-para-programar-en-android> (22-09-2013)
- [5] Mr. Geek (2012) OTROGEEK [Online] Disponible en <http://www.otrogeek.net/2012/10/emuladores-de-telefonos-moviles.html> (22-09-2013)
- [6] Cosmos (2013) Conectividad Android KitKat [online] Disponible en <http://www.xatakandroid.com/sistema-operativo/android-4-4-kitkat-anunciado-oficialmente> (04-12-2013)
- [7] Anónimo (2013) Developers [Online] Disponible en http://developer.android.com/guide/practices/screens_support.html (30-11-2013)
- [8] alejosoft (2008) todoexpertos [Online] Disponible en <http://www.todoexpertos.com/categorias/tecnologia-e-internet/programacion/java/respuestas/1811667/ajustar-imagen-a-la-pantalla-del-movil-con-j2me> (04-12-2013).