

Un Análisis comparativo de rendimiento en Aplicaciones Móviles Multiplataforma

Lisandro Delía¹, Nicolás Galdamez¹, Leonardo Corbalan¹, Pablo Thomas¹
Patricia Pesado^{1,2}

¹ Instituto de Investigación en Informática LIDI. Facultad de Informática.
Universidad Nacional de La Plata. Argentina

² Comisión de Investigaciones Científicas de la Provincia de Buenos Aires. Argentina

{ldelia, ngaldamez, corbalan, pthomas, ppesado}@lidi.info.unlp.edu.ar

Resumen. Es razonable pensar que la comunidad de desarrolladores de aplicaciones móviles opte por plataformas nativas cuando pretende obtener mejor rendimiento en relación al tiempo de ejecución en sus productos. No obstante, existe la posibilidad de optar por un desarrollo multiplataforma, a priori con más restricciones, pero con mayores beneficios en cuanto a portabilidad y escalabilidad. En este trabajo se realiza un análisis comparativo de rendimiento de distintos modos de desarrollo multiplataforma, a través de un cálculo numérico simple.

Palabras claves: dispositivos móviles, aplicaciones móviles multiplataforma, aplicaciones móviles nativas, performance.

1 Introducción

El desarrollo de software para dispositivos móviles tiene características particulares. La necesidad de tratar con diversas plataformas, estándares, protocolos y tecnologías de red; las capacidades limitadas de los dispositivos y las exigencias de tiempo del mercado, son sólo algunos de los problemas a tratar. Por ello, el desarrollo de software para dispositivos móviles difiere considerablemente del tradicional [1].

Cuando se desarrolla una aplicación para dispositivos móviles, la expectativa es que sea ejecutable en la mayor cantidad de dispositivos posible. Para ello existen dos alternativas: desarrollar en forma “nativa” para cada plataforma del mercado, es decir, utilizar un entorno de desarrollo integrado, un lenguaje y herramientas propias de cada plataforma [2]; o construir aplicaciones que puedan ejecutarse en cualquier plataforma, es decir, que sean multiplataforma.

La elección del modo (nativo o multiplataforma) de desarrollo de aplicaciones para dispositivos móviles depende de varios factores. Uno de ellos, en ocasiones primordial, es el tiempo de ejecución.

Si el objetivo es priorizar el rendimiento de la aplicación sobre otros factores que inciden en la producción de software, el desarrollo nativo se presenta como la opción natural [5] [6] [7]. No obstante, las aplicaciones nativas no son portables y la escalabilidad se reduce al mismo sistema operativo para las cuales fueron creadas. Por

lo tanto, es razonable optar por desarrollos multiplataforma, aunque teniendo en cuenta el rendimiento en término de tiempo, entre otros factores, de las alternativas posibles.

En artículos anteriores [5] [6] [7] hemos analizado ventajas y desventajas de las alternativas mencionadas, desde el punto de vista del Ingeniero de Software. En el presente trabajo nos focalizamos sobre el rendimiento relacionado con la velocidad de procesamiento. Para ello hemos efectuado comparaciones de tiempos de ejecución de distintas aplicaciones multiplataforma, avanzando así en la línea de investigación iniciada en [5].

En la sección 2 se describe el experimento utilizado para comparar tiempos de ejecución entre diversas aplicaciones móviles multiplataforma, en la sección 3 se muestran y analizan los resultados obtenidos y en las secciones posteriores se presentan las conclusiones y trabajo futuro.

2 Experimento

Los Sistemas Operativos Móviles elegidos para la experimentación fueron Android e iOS, dado que a Junio del 2015 entre ambos abarcaban el 80% del mercado mundial, situación también reflejada en la Argentina, en los dos casos de acuerdo a lo indicado en [8].

Se seleccionaron las plataformas de desarrollo multiplataforma, considerando las alternativas presentadas en [6] [7]. Se optó por un desarrollo híbrido utilizando Apache Cordova [9], una aplicación interpretada en Appcelerator Titanium [10], y una generada por compilación cruzada utilizando Xamarin [11]. De este modo se logró una muestra razonablemente representativa de las diversas opciones existentes en la actualidad. Las aplicaciones web móviles, también consideradas multiplataforma, quedaron excluidas de este experimento.

El conjunto de dispositivos móviles utilizados durante la experimentación se describe en la tabla 1. Cada uno de ellos representa un escenario de prueba sobre el que se establecieron los distintos casos multiplataforma a evaluar.

Tabla 1. Dispositivos móviles seleccionados para el experimento.

Escenario de prueba	Dispositivo	Marca	Modelo	Sistema Operativo
1	Smartphone	Samsung	Advance, N° GT - I9070	Android 4.1.2
2	Smartphone	Samsung	S3 NEO, N° I9301I	Android 4.4.2
3	Smartphone	Motorola	Moto-G	Android 5.0.2
4	Smartphone	Motorola	Moto-G2	Android 4.4
5	Tablet	Samsung	Tab 2	Android 4.2.2
6	Smartphone	Apple	Iphone 5S	iOS 8.4

En resumen, se realizó el experimento en dos sistemas operativos (cinco versiones

distintas en el caso de Android), tres ambientes multiplataforma, y seis dispositivos móviles diferentes.

Con el fin de evaluar la velocidad de procesamiento, se planteó un cálculo simple que incluyó varias iteraciones, funciones matemáticas y aritmética de punto flotante y se resume en la siguiente serie:

$$serie = \sum_{j=1}^5 \sum_{k=1}^{100000} (\log_2(k) + \frac{3k}{2^j} + \sqrt{k} + k^{j-1}) \quad (1)$$

A modo de ejemplo a continuación se muestra el código multiplataforma desarrollado en Apache Cordova.

```
var inicio = new Date().getTime();
var serie = 0;
for ( var j=1; j <= 5; j++ )
{
    for ( var k=1; k <= 100000; k++ )
    {
        serie = serie + (Math.log(k)/Math.LN2) + (3*k/2*j) +
            Math.sqrt(k) + Math.pow(k, j-1);
    }
}
var fin = new Date().getTime();
var tiempo = fin - inicio;
document.getElementById('resultado').innerHTML = tiempo +
' -> ' + serie;
```

El experimento planteado es simple pero efectivo y permite medir con precisión la variable analizada, en este caso, el tiempo de ejecución requerido para realizar cálculo intensivo. En la siguiente sección se analizan los resultados obtenidos.

3 Resultados obtenidos

A fin de obtener resultados verosímiles, para cada caso de prueba definido, se realizaron 30 ejecuciones independientes del experimento diseñado obteniendo una muestra $T = T_1, T_2, \dots, T_{30}$, con T_i = tiempo requerido para el cálculo de la serie en la i -ésima ejecución del experimento planteado. El tiempo T_i se expresa en milisegundos.

Para caracterizar cada una de las muestras obtenidas, se han calculado los estadísticos \bar{T} y S que se corresponden con la media (o promedio muestral) y la desviación estándar muestral (ver Tabla 2).

En la tabla 3 se presenta una síntesis de los resultados obtenidos conformada por los valores \bar{T} y S calculados para cada caso de prueba planteado. Estos valores deben ser contrastados entre los distintos modos de desarrollo multiplataforma dentro de un

mismo escenario de prueba. Aunque estos dos estadísticos conforman una buena caracterización de la muestra sobre la que se calculan, los diagramas de caja presentados en la figura 1 complementan esa caracterización ofreciendo mayor información sobre la distribución de los datos en cada muestra.

Tabla 2. Estadísticos empleados en el análisis de los datos

Dada la muestra $T=T_1, T_2, \dots, T_n$	
Media o promedio muestral	$\bar{T} = \left(\frac{1}{n}\right)\sum_{i=1}^n T_i$
Desviación estándar muestral	$S = \sqrt{\frac{1}{n-1}\sum_{i=1}^n (T_i - \bar{T})^2}$

Tabla 3. Síntesis de resultados obtenidos

		Apache Cordova	Titanium	Xamarin
Samsung Advance	\bar{T}	228,07	237,40	468,20
	S	14,43	13,01	8,98
Samsung S3 Neo	\bar{T}	596,10	190,10	339,93
	S	7,35	7,33	10,45
MOTO-G	\bar{T}	189,43	286,23	379,20
	S	6,61	24,04	5,55
MOTO-G 2	\bar{T}	230,33	211,67	395,17
	S	14,22	24,95	8,95
Samsung Tab 2	\bar{T}	190,60	192,70	379,33
	S	9,36	16,80	8,31
Iphone 5S	\bar{T}	287,90	267,20	98,47
	S	7,10	2,78	2,97

Un diagrama de caja consiste en un rectángulo dispuesto en forma horizontal, también puede ser en sentido vertical, que representa el recorrido desde el primer cuartil hasta el tercero. Este rectángulo, sobre el que recaen la mitad de los datos muestreados, está dividido por un segmento vertical que indica donde se posiciona la mediana (segundo cuartil) haciendo visible su relación con los cuartiles primero y tercero. La caja se ubica a escala sobre un segmento que tiene como extremos los valores mínimo y máximo de la muestra.

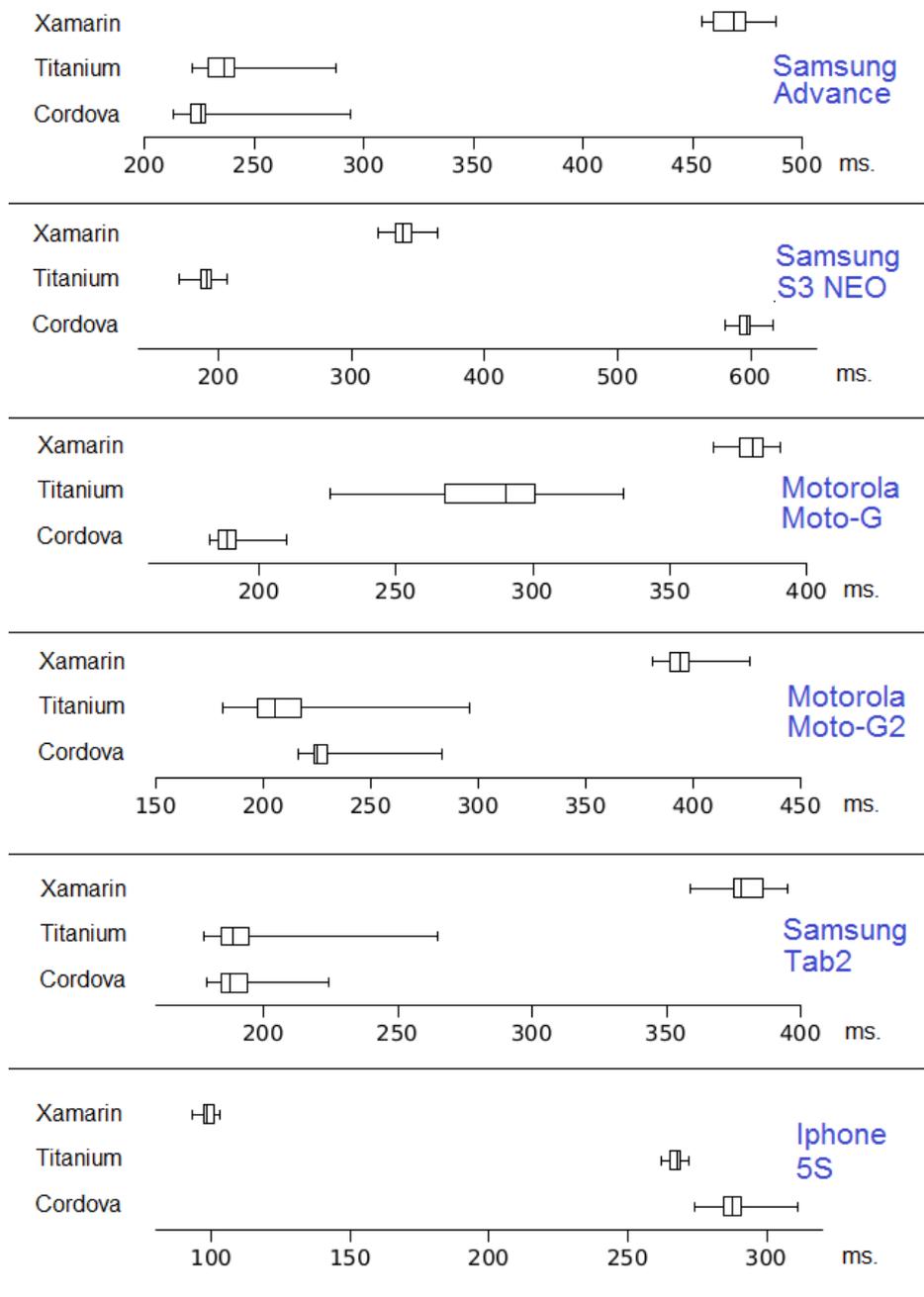


Figura 1. Diagrama de cajas con las muestras recolectadas en los seis escenarios de prueba planteados.

Los diagramas de caja son sumamente útiles pues permiten una rápida comparación visual sobre la dispersión de los datos en cada una de las muestras representadas.

Según se aprecia en la figura 1, en la mayoría de los casos las muestras obtenidas resultaron suficientemente separadas unas con otras. Ello es prueba de que las diferencias observadas sobre \bar{T} en cada escenario de prueba (ver Tabla 3) son significativas.

Sin embargo, en los escenarios 1, 4 y 5, Samsung Advance, Motorola Moto-G2 y Samsung Tab2 respectivamente, las muestras obtenidas para los modos de desarrollo multiplataforma Apache Cordova y Titanium presentan un marcado solapamiento haciendo más difícil el análisis visual a partir de los diagramas de caja. Por lo tanto es necesario establecer si tales diferencias son estadísticamente significativas por medio de un test de hipótesis sobre diferencia de medias.

Debido a que las muestras obtenidas por la experimentación llevada a cabo presentan una distribución cercana a la distribución Normal (se aplicó la prueba para normalidad de Kolmogorov-Smirnov), es posible utilizar el test *t-Student* para diferencias de medias a partir de muestras independientes.

Básicamente el test *t-Student* para diferencia de medias postula como hipótesis nula que ambas medias son iguales y como hipótesis alternativa que son distintas. Como resultado de aplicar el test *t-Student* se obtiene el p-valor que representa la probabilidad de cometer error de tipo I, es decir de rechazar la hipótesis nula cuando ésta es verdadera. Consecuentemente, cuanto más pequeño sea el p-valor, mayor evidencia habrá a favor del postulado que las medias son distintas. En general, hay consenso en la comunidad científica en que si $p\text{-valor} < 0,05$ debe reconocerse que la diferencia de medias observada es estadísticamente significativa.

La tabla 4 resume el análisis estadístico llevado a cabo sobre los escenarios y los modos de desarrollo indicados

Tabla 4. Prueba de significancia estadística *t-Student*

	Apache Cordova \bar{T} (S)	Titanium \bar{T} (S)	p-valor	Resultado
Samsung Advance	228,07 (14,43)	237,40 (13,01)	0,01091	ES
Motorola Moto-G2	230,33 (14,22)	211,67 (24,95)	0,00087	EES
Samsung Tab 2	190,60 (9,36)	192,70 (16,80)	0,55271	ENS

ES = Estadísticamente Significativo. ENS = Estadísticamente No Significativo.
EES = Extremadamente Estadísticamente Significativo.

A partir de los datos presentados en la tabla precedente y demás pruebas *t-Student* realizadas, se puede afirmar que todas las diferencias en los promedios muestrales presentados en este trabajo son estadísticamente significativas a excepción de la correspondiente a Apache Cordova y Titanium en el escenario 5 (Samsung Tab2). En este último caso no se ha encontrado evidencia suficiente para sostener que las

aplicaciones realizadas con Apache Cordova mejoran el rendimiento de aquellas construidas con Titanium, o viceversa.

La figura 2 presenta en forma de diagrama de barras la información respecto del promedio muestral para cada escenario de prueba y modo de desarrollo multiplataforma evaluado. Estos valores se corresponden con los descriptos en la tabla 3.

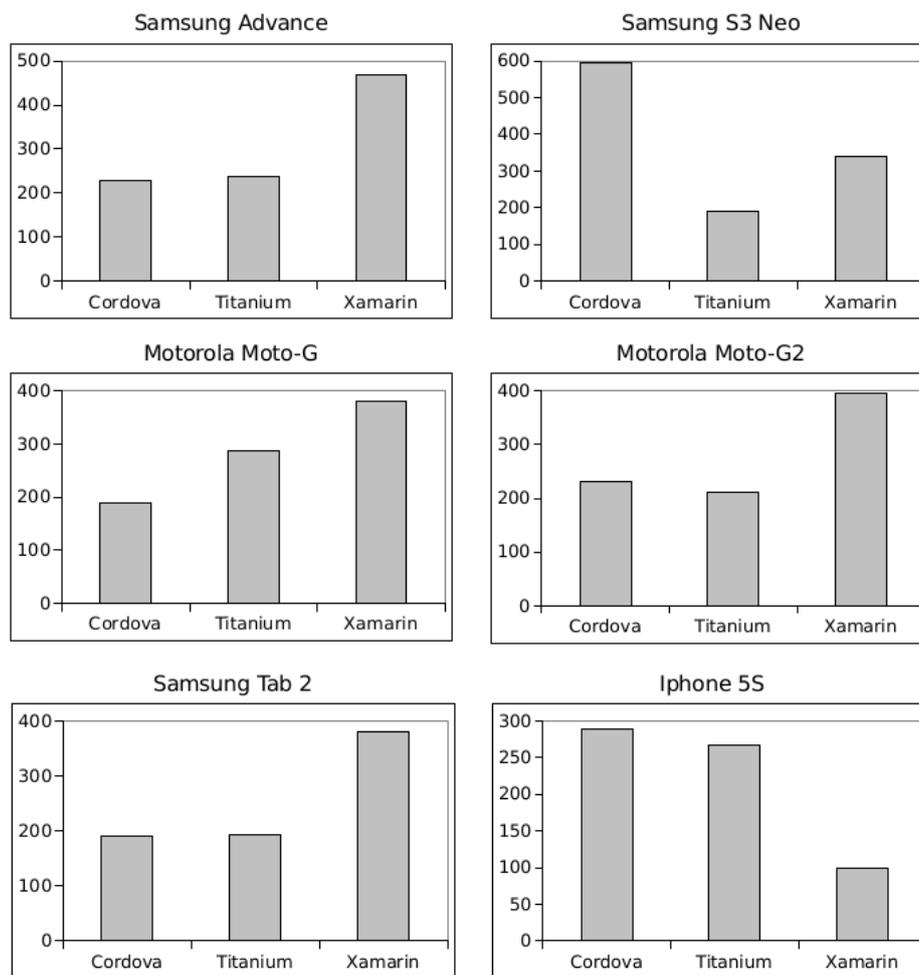


Figura 2. Diagrama de barras que visualiza la media del tiempo de ejecución empleado en el cálculo de la serie (1) observado en cada uno de los escenarios de prueba para los modos de desarrollo multiplataforma evaluados

Planteadas la competencia entre Apache Cordova (desarrollo híbrido), Titanium (aplicación interpretada) y Xamarin (compilación cruzada), la siguiente tabla resume los resultados generales de esta contienda.

Tabla 5. Resultados generales

Escenario	Posiciones		
	Apache Cordova	Titanium	Xamarin
Samsung Advance	1º puesto	2º puesto	3º puesto
Samsung S3 Neo	3º puesto	1º puesto	2º puesto
Motorola Moto-G	1º puesto	2º puesto	3º puesto
Motorola Moto-G2	2º puesto	1º puesto	3º puesto
Samsung Tab 2	1º puesto	1º puesto	3º puesto
Iphone 5S	3º puesto	2º puesto	1º puesto

4 Conclusiones

Se ha presentado un estudio comparativo sobre el rendimiento de distintos modos de desarrollo multiplataforma en relación con el tiempo de ejecución de las aplicaciones construidas. Los escenarios de prueba diseñados incluyeron dos sistemas operativos (cinco versiones de Android y una de iOS) y diversos dispositivos (una *Tablet* y cinco *smartphones* distintos). Sobre dichos escenarios fueron puestos a prueba Apache Cordova, Titanium, y Xamarin (desarrollo híbrido, aplicación interpretada y generada por compilación cruzada respectivamente).

Planteada la competencia entre estos tres modos de desarrollo multiplataforma, Titanium ha mostrado el mejor rendimiento global. De los 6 escenarios configurados, obtuvo el primer puesto tres veces (uno compartido con Apache Cordova) y el segundo puesto las otras tres.

Apache Cordova representa el segundo mejor rendimiento general obteniendo el primer puesto tres veces (uno compartido con Titanium) una vez el segundo puesto y dos veces el tercero.

Finalmente Xamarin mostró el rendimiento general más bajo obteniendo cuatro veces el tercer puesto, una vez el segundo y una vez el primer puesto.

Es destacable que la compilación cruzada, por medio de su representante Xamarin, sólo haya obtenido el mejor rendimiento sobre el sistema operativo iOS mientras que en todos los escenarios de prueba basados en dispositivos Android obtuvo el rendimiento más bajo, salvo en una ocasión que, aún así, fue superado por Titanium.

Se concluye que, entre los tres modos de desarrollo multiplataforma puestos a prueba, cuando se prioriza el rendimiento en relación al tiempo de ejecución de las aplicaciones construidas, Titanium representa la mejor opción resultando siempre uno de los mejores tiempos de ejecución y comportándose de manera estable, sin grandes variaciones de rendimiento entre distintos dispositivos.

5 Trabajo futuro

Se plantea como línea de trabajo futuro profundizar el estudio y análisis sobre el desarrollo multiplataforma examinando vía experimentación el rendimiento de las

aplicaciones así construidas. Se plantea incorporar en este análisis a otros modos de desarrollo multiplataforma y establecer las pertinentes comparaciones con las aplicaciones desarrolladas en forma nativa. Se propone también la evaluación de otros aspectos de rendimiento, para cuantificarlo en relación al acceso a disco, al consumo de batería y en general a otras capacidades del dispositivo.

Referencias

1. Hayes, I. S. *Just Enough Wireless Computing*. Prentice Hall Professional Technical Reference. 2002. ISBN:0130994618
2. Spyros Xanthopoulos, Stelios Xinogalos, *A Comparative Analysis of Cross-platform Development Approaches for Mobile Applications*, BCI' 2013, Greece
3. Yonathan Aklilu Redda, *Cross platform Mobile Applications Development*, Norwegian University of Science and Technology, Norwegian University of Science and Technology, Norwegian University of Science and Technology, Master in Information Systems, June 2012.
4. Dalmaso I., Datta S.K., Bonnet C. Nikaein N., *Survey, comparison and evaluation of cross platform mobile application development tools*, Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International.
5. Delia L., Galdamez N. Thomas P., Pesado P., *Un Análisis Experimental de Tipo de Aplicaciones para Dispositivos Móviles*, XVIII Congreso Argentino de Ciencias de la Computación, CACIC 2013.
6. Delia, L.; Galdamez, N.; Thomas, P.; Corbalan, L.; Pesado, P., *Análisis Experimental de desarrollo de Aplicaciones Móviles Multiplataforma*, XX CACIC, Universidad Nacional de La Matanza, Octubre 2014.
7. Delia, L.; Galdamez, N.; Thomas, P.; Corbalan, L.; Pesado, P., *Multi-platform mobile application development analysis*, Research Challenges in Information Science (RCIS), 2015 IEEE 9th International Conference on, Atenas, Grecia, Año 2015.
8. <http://gs.statcounter.com>
9. <http://cordova.apache.org>
10. <http://www.appcelerator.com>
11. <http://xamarin.com>