

Comparación de dos tecnologías de desarrollo de aplicaciones móviles desde la perspectiva del rendimiento como atributo de calidad

Comparison of two technologies mobile application development from a performance perspective as Quality attribute

Álvaro Javier Durán Sanjuán¹, Jorge Luis Peinado Rodríguez², Albeiro Alonso Rosado³
^{1, 2, 3}*Ingeniero de sistemas y computación, Universidad Francisco de Paula Santander, Santander, Colombia*
ajdurans@ufpso.edu.co
jlpeinador@ufpso.edu.co
aarosadog@ufpso.edu.co

Resumen— Este trabajo se ocupa de confrontar dos tecnologías para el desarrollo de aplicaciones móviles: tecnología de desarrollo nativo y tecnología basada en web, buscando determinar cuál de ellas es la mejor con respecto al atributo de calidad del rendimiento. Para esto se construyeron dos aplicaciones en las tecnologías mencionadas: una nativa y la otra basada en web, y entre éstas se llevó a cabo una comparación que se basó en el proceso de evaluación propuesto por el estándar ISO/IEC 14598-1 junto con el modelo de calidad del producto de ISO/IEC 9126-1, resultando de mayor rendimiento la aplicación construida con tecnología de desarrollo nativa.

Palabras clave— Android, atributos de calidad, dispositivos móviles, evaluación de software, ISO/IEC 14598, ISO/IEC 9126.

Abstract— This research is focused to confronting two technologies for developing mobile applications: technology of native development and web-based technology, seeking to determine which one is the best regarding to quality attribute performance. For this two applications were built on the technologies mentioned: one native and other web based technology, and among these was conducted a comparison based on the assessment process proposed by the ISO / IEC 14598-1 standard, along with quality product model ISO / IEC 9126-1, resulting in higher performance the application built with technology of native development.

Key Word — Android, quality attributes, mobile devices, evaluation of software, ISO/IEC 14598, ISO/IEC 9126.

I. INTRODUCCIÓN

En la actualidad existe un campo creciente en el desarrollo de software, se trata del mercado de las aplicaciones para dispositivos móviles, que hoy representa grandes oportunidades en este sector [1]. Si bien el desarrollo de dichas aplicaciones es relativamente nuevo, ya se cuenta con herramientas que lo facilitan, y que además cobran alta importancia debido a la serie de plataformas móviles existentes, de las cuales, *Android* es la más sobresaliente, entre otras razones, porque está implantado fuertemente en el ámbito mundial.

Así las cosas, el desarrollador de aplicaciones móviles se encuentra frente a la necesidad de definir cuál tecnología de desarrollo emplear, si bien la tecnología de desarrollo nativo o si debe ser la basada en web.

El método para dar solución al problema planteado fue la comparación de las dos tecnologías utilizadas en el desarrollo de aplicaciones móviles, desde la perspectiva del atributo de calidad del rendimiento, y basándose en los parámetros establecidos por los estándares internacionales ISO/IEC 14598 e ISO/IEC 9126, además, haciendo uso de una herramienta de monitorización y análisis de rendimiento [3].

II. METODOLOGÍA

Para comenzar con el proyecto fue necesario establecer los requisitos de la evaluación de los atributos de calidad conforme al modelo de calidad del estándar ISO/IEC 9126, y para lo cual se escogió una única característica; posteriormente, se especificó la evaluación, tomando como

base el proceso planteado por ISO/IEC 14598 y considerando el contexto de las aplicaciones móviles. En este sentido, la comparación de las tecnologías de desarrollo exigió construir dos versiones de la misma aplicación móvil bajo tecnología nativa y basada en web, al producto generado, se le efectuó la evaluación propuesto por los estándares.

A. Establecer los requisitos de la evaluación

En este trabajo se escogió el estándar ISO/IEC 9126 como modelo de calidad del producto software[4] junto con el ISO/IEC 14598 como modelo de evaluación[5].

El modelo de calidad, establecido en la primera parte de dicho estándar (ISO/IEC 9126-1), estructura la calidad del software dividiéndola en tres partes: calidad interna, calidad externa y calidad en uso. La calidad interna y externa son abordadas especificando seis características: funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad; mientras que la calidad en uso comprende cuatro características: efectividad, productividad, seguridad y satisfacción[4].

Para realizar la evaluación de calidad no se tuvo en cuenta la totalidad de los atributos[6], de acuerdo a dos razones que se explican a continuación:

- a. No es posible optimizar todos los atributos para un sistema. Dada la amplia variedad de atributos de calidad a considerar, no es posible abarcar todos los atributos para un sistema[7].
- b. Considerar los atributos más importantes de acuerdo al sistema a evaluar. Los atributos de calidad a evaluar dependen de su aplicación; de manera que el contexto de implantación de la aplicación determina cuáles características son más relevantes que otras[7][8].

Por lo anterior, se han de considerar las características más significativas de acuerdo al tipo de software y al caso particular en que se ha de aplicar el modelo de calidad. Es importante utilizar las características que vienen a ser más relevantes por tratarse de aplicaciones para dispositivos móviles. En este orden de ideas, se presenta a continuación una serie de enfoques por varios autores para argumentar por qué se seleccionaron las características de calidad que hicieron parte de la evaluación.

Dentro del mundo de los móviles se considera que desde la construcción o desarrollo es necesario probar las aplicaciones teniendo en cuenta su propio contexto y considerando el cumplimiento de los atributos de calidad. En concreto, se habla de cuatro características para las aplicaciones móviles: funcionalidad, usabilidad, seguridad y rendimiento[9]. Por otro lado, el usuario es quien debe definir las características a tener en cuenta en el proceso de evaluación de forma que este permita determinar las métricas más relevantes de acuerdo a la situación o aplicación[10]. Además del usuario y del contexto de uso, entre los aspectos importantes para el desarrollo de aplicaciones móviles están: la experiencia de usuario –

interfaz gráfica, entrada de usuario, rendimiento, seguridad y privacidad[11]; esto apunta a tres características de calidad presentes en el modelo ISO/IEC 9126, en donde permite establecer una correspondencia entre la usabilidad y la experiencia de usuario – interfaz gráfica y entrada de usuario. Además, el rendimiento coincide con la característica eficiencia del modelo de calidad, por último, con respecto a seguridad y privacidad hay correspondencia con la seguridad en el estándar ISO/IEC 9126.

1. Selección de la eficiencia (rendimiento) como única característica de calidad a evaluar

Aunque se viene exponiendo que las características de calidad indispensables en las aplicaciones son: funcionalidad, seguridad, usabilidad y rendimiento[7], en el contexto de las aplicaciones móviles la eficiencia o también llamada rendimiento, se destaca notablemente, y por tanto fue la única característica de calidad que se escogió en este estudio. Según una encuesta elaborada por Capgemini, Sogeti y HP el rendimiento es la cualidad más buscada; dicha encuesta encontró que el 64% de los usuarios opina que el rendimiento es la cualidad más importante y está por encima de otros atributos como la funcionalidad con 48%, la portabilidad con 46% e interfaz de usuario con 36%[12].

Otro estudio realizado por Apigee, encontró que el 96% de los usuarios manifiestan frustración cuando encuentran problemas como: aplicaciones congeladas, capacidad de respuesta lenta, demasiado uso de la batería y cantidad exagerada de publicidad. El mismo estudio también mostró que el 98% de los encuestados basan su satisfacción en el rendimiento de la aplicación; cuando no funcionan como se esperaba o como se ha anunciado el 99% de los encuestados dijeron que tomarían medidas como: eliminar la aplicación inmediatamente: el 44%, eliminar la aplicación si se bloquea durante más de 30 segundos: el 38%, comentar a sus amigos y colegas su pésimo funcionamiento: el 32%, y quejarse de la aplicación en *Facebook* y *Twitter*: el 21%[13].

Entre los principales problemas que han experimentado los usuarios se encuentra las fallas por: bloqueo o error con un 62%, tiempos de lanzamiento lentos: el 47%, y el 40% ha probado una aplicación que simplemente no ha arrancado[14]. Estos problemas causan en los usuarios sentimientos negativos, por eso casi nueve de cada diez estadounidenses se decepcionan de las marcas que tienen aplicaciones móviles con un bajo rendimiento. Dentro de las cualidades que se buscan en una aplicación móvil, se encuentran la rapidez y el tiempo de espera, con lo cual se ratifica el interés de los usuarios en el rendimiento de las aplicaciones móviles[15].

En definitiva, es posible identificar que el rendimiento de las aplicaciones móviles es un aspecto clave en la actualidad, es lo que más interesa, tanto así que se habla del rendimiento como la clave del éxito[12]. Esta característica se ubica por encima de otras[14] puesto que los usuarios de las aplicaciones móviles exponen un inconformismo muy significativo en aspectos asociados con la eficiencia de

ellas[13][15]. Finalmente, dados los estudios realizados y siguiendo las características definidas[8] es posible establecer que los usuarios manifiestan inconvenientes de las aplicaciones móviles relacionados con el rendimiento, lo que en el modelo de calidad de ISO/IEC 9126 corresponde a la característica de calidad Eficiencia.

Dentro del contexto de movilidad con el que se determinó el rendimiento como la única característica de calidad a evaluar, los requerimientos de la evaluación corresponden solamente a los atributos de calidad relacionados con tal característica. En este orden de ideas, los requerimientos no funcionales[16] son los siguientes:

- a) Verificar el tiempo de respuesta ante una solicitud por parte del usuario.
- b) Comprobar el tiempo de espera que el usuario experimenta.
- c) Verificar la utilización de recursos de memoria RAM en el cumplimiento de una función.
- d) Verificar la utilización de CPU en el cumplimiento de una función.
- e) Inspeccionar el comportamiento del consumo de batería.

2. Descripción del producto

La aplicación fue construida con el objetivo de realizar un caso de estudio para comparar las tecnologías de desarrollo que se vienen estudiando, se le dio el nombre de NewsApp. Se diseñó para ser instalada en un *Smartphone Android* con sistema operativo 4.0 *Ice Cream Sandwich*[2], y se preparó para ser ejecutada sin inconvenientes en una pantalla de resolución 1280 x 720 píxeles. La aplicación consiste en un Really Simple Syndication (RSS), que es un sencillo lector o recopilador de contenidos que presenta cinco fuentes, de las cuales se puede tener acceso a las últimas noticias, obteniéndose el título de cada una junto con una breve descripción y el enlace para ver la noticia completa en el sitio web de la misma. En la figura 1, se aprecia la arquitectura de la aplicación, en ella se muestra cómo la aplicación instalada en el celular envía y recibe mensajes [16] al servidor que contiene los contenidos en formato RSS.

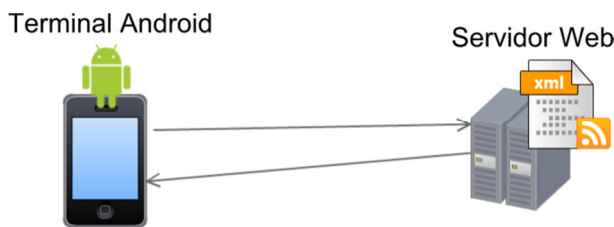


Figura 1. Arquitectura de la aplicación móvil NewsApp.

B. Especificación de la evaluación

El estándar ISO/IEC 9126-1 presenta tres vistas de calidad: interna, externa y calidad en uso; debido a que es necesario efectuar el proceso de evaluación de calidad una vez el producto software esté terminado, se seleccionaron las métricas de la vista de calidad externa, que es la medición

cuando el software es ejecutado, la cual es evaluada mientras se prueba en un ambiente simulado con datos simulados[4].

El estándar ISO/IEC 9126, propone un modelo de calidad con una visión general aplicable a todo tipo de software. Por lo tanto, deja abierta la posibilidad de modificar y hacer consideraciones sobre las métricas definidas con el propósito de adaptarse al contexto al que pertenece el producto software a evaluar, además, es posible usar métricas que no están presentes en el estándar. Para este caso el software es una aplicación para dispositivos móviles, de acuerdo a ese contexto es preciso enfatizar en aspectos que proponen retos en el diseño de dichas aplicaciones:

- **El consumo de batería.** Este consumo es relevante puesto que la energía es limitada y requiere de administración apropiada[17]. Además, uno de los errores que debe evitarse es el alto consumo de batería [9]; aproximadamente la mitad de los usuarios de aplicaciones móviles manifiestan tener problemas por el alto consumo de batería [13]; la satisfacción con los *smartphones* se ve afectada en la duración de la vida de la batería antes de que se requiera recarga; por esta razón, el rendimiento de la batería es el aspecto menos satisfactorio de estos teléfonos [14].

- **Consumo de CPU.** También se requiere la consideración de la CPU puesto que, en general, la capacidad de procesamiento de los dispositivos móviles en la actualidad es limitada, si bien existen terminales que poseen características de CPU avanzadas, estas también tienen su límite.

En la tabla 1, se presenta la característica eficiencia desglosada en subcaracterísticas y métricas.

Característica	Subcaracterística	Métrica
Eficiencia	Comportamiento temporal	Tiempo de respuesta
		Tiempo de respuesta medio
		Tiempo de respuesta en el peor caso
	Utilización de recursos	Utilización máxima de memoria
		Utilización máxima de CPU
		Utilización máxima de batería

Tabla 1. Desglose de la característica de calidad eficiencia.

1. Establecer niveles de puntuación para las métricas

La puntuación para las métricas se llevó a cabo interpretando los resultados de las mediciones con el uso de los siguientes criterios:

Para las métricas del comportamiento temporal: $0 < T$, Cuanto más pronto es mejor, donde T corresponde al tiempo.

Para la utilización de recursos: $0 \leq R$, Cuanto más pequeño es el mejor, donde R es el recurso, que puede ser: memoria RAM, CPU o batería.

se observa la versión final de NewsApp nativa y en la figura 3 la de basada en web.

2. Establecer criterios para la valoración

Para el proceso de evaluación no se consideró el uso de un modelo calidad diferente al de ISO/IEC 9126, que es el sugerido por el estándar ISO/IEC 14598.

Diseño de la evaluación. Esta etapa consiste en especificar los métodos, procedimientos y herramientas necesarias en la evaluación.

Descripción de la herramienta de evaluación. Para la medición se seleccionó un software denominado Little Eye, una herramienta de monitoreo que puede analizar diversos tipos de datos relacionados con el rendimiento de una aplicación para dispositivos *Android*[3].

Esta herramienta permite medir el consumo de recursos de una aplicación determinada, obteniendo estadísticas sobre cantidad de energía, datos, *CPU*, memoria *RAM* y espacio en disco que la aplicación está utilizando. Little Eye monitorea en tiempo de ejecución, y se presentan los resultados mediante gráficas que se actualizan mientras es realizado el monitoreo, puesto que captura el video de la pantalla del teléfono, permitiendo observar las acciones que se llevan a cabo en el mismo. Una vez finalizada la evaluación del rendimiento es posible acceder a la grabación del monitoreo y analizar cualquier punto de interés en la línea de tiempo.

C. Construir una aplicación móvil tanto en tecnología nativa como una basada en web

1. **Descripción de la aplicación desarrollada.** Se construyó una aplicación móvil con el nombre NewsApp tanto en la versión nativa como en la basada en web; ambas se desarrollaron buscando la mayor similitud entre ellas.
2. **Herramientas de desarrollo empleadas.** Para la aplicación nativa. Se escogió el kit de desarrollo de software (SDK, software development kit) de *Android* y el *IDE* Eclipse con el plugin ADT, teniendo en cuenta que el SDK de *Android* es la herramienta oficial y además es *open source*.

La aplicación basada en web. Se eligió Phonegap porque es la plataforma más utilizada por los desarrolladores, además es compatible con jQuery Mobile, que fue el framework seleccionado para la construcción de interfaces gráficas[18].

3. **Metodología de desarrollo.** Las metodologías denominadas ágiles [19] son las que mejor se adecúan para el caso de aplicaciones móviles[20], de acuerdo a esto se hizo uso del modelo ágil Extreme Programming (XP) [21] en la construcción de NewsApp; se elaboraron las historias de usuario y diarios de actividades[22]. Durante el desarrollo se realizaron dos iteraciones, siendo la segunda la versión final de la aplicación. En la figura 2

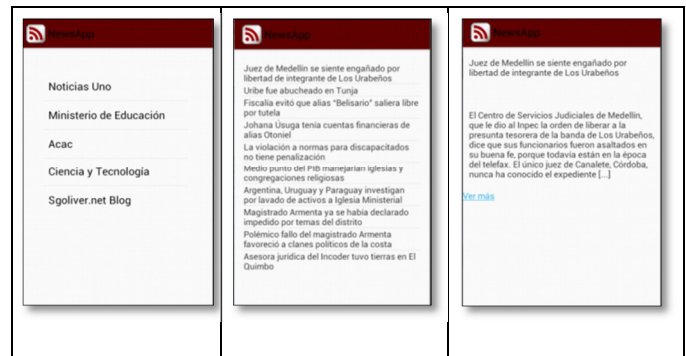


Figura 2. Versión final de NewsApp nativa.



Figura 3. Versión final de NewsApp basada en web.

D. Efectuar el proceso de evaluación de los atributos de calidad del software

1. **Ejecución de la evaluación.** Se tomaron medidas de comportamiento temporal (Tiempo de respuesta) y utilización de recursos (*CPU*, memoria *RAM*, batería) mediante el uso de la herramienta Little Eye, instalada en una computadora de escritorio con especificaciones de hardware que se detallan en la tabla 2.

Computador para monitoreo	
Procesador	AMD Phenom II X6 2.7 Ghz
Memoria RAM	4 GB
Disco duro	1 TB
Pantalla	Samsung 21"
SO	Windows 7 Professional 64 bits

Tabla 2. Características de hardware de computador de monitoreo.

La aplicación NewsApp se ejecutó en un *Smartphone Android* de características que se indican en la tabla 3.

Smartphone Sony Xperia S (Lt26i)	
Procesador	Dual-core de 1,5 GHz
Chipset	Qualcomm MSM8260 Snapdragon
GPU	Adreno 220
Memoria RAM	1 GB
Memoria interna	32 GB
Pantalla	720 x 1280 píxeles 4.3 pulgadas
Sensores	Acelerómetro, giroscopio, proximidad,

	brújula
Navegador	Compatible HTML5
Versión SO	Android 4.0 Ice Cream Sandwich

Tabla 3. Características de hardware de Sony Xperia S.

También se verificaron las condiciones en el teléfono antes de ejecutar la evaluación con el fin de obtener información con respecto al uso de memoria RAM, almacenamiento interno y estado de la batería como se muestra en la tabla 4.

Condiciones iniciales en el teléfono	
Almacenamiento interno	25 GB libres
Uso de memoria RAM	330MB usada; 308MB libre
Carga de batería	100%
Brillo de la pantalla	50%
Aplicaciones en ejecución	- NewsApp - Little Eye App Monitor
Intensidad señal WiFi	Aceptable
Velocidad de descarga*	0.96 Mbps

Tabla 4. Condiciones del teléfono previas la evaluación.

III. RESULTADOS

Luego de haber tabulado los resultados, se realizó el respectivo análisis que a continuación se presenta. Los resultados están organizados a través de gráficos de barras, los cuales muestran los resultados obtenidos en las diferentes pruebas. La barra de color rojo corresponde a la versión desarrollada con herramientas basadas en web y el color azul para la desarrollada con código de forma nativa.

Una de las primeras pruebas que se realizó fue la del tiempo de inicio de la aplicación, en donde NewsApp basada en web tardó 4,75 segundos, mientras que la nativa demoró sólo un segundo, lo cual indica que NewsApp basada en web tarda 3,75 segundos más en iniciar que la nativa, es decir, esta última es aproximadamente 4 veces más rápida que la basada en web; a pesar de esta situación, los tiempos de inicio son aceptables dado que éste debe estar por debajo de los seis segundos [17].

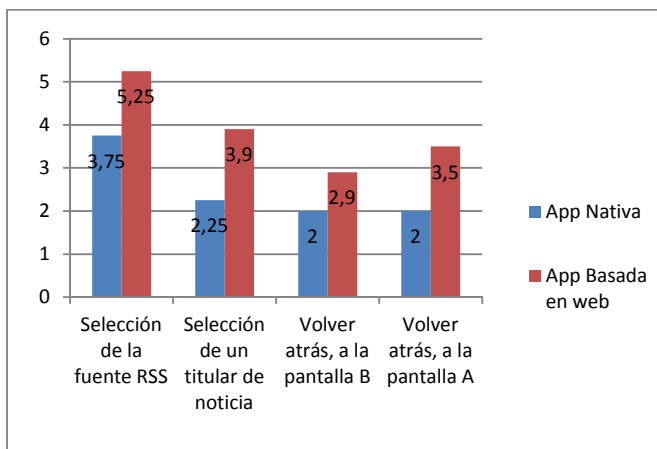


Figura 4. Tiempo de respuesta en segundos para NewsApp nativa VS. Tiempo de respuesta en NewsApp basada en web (s).

En la figura 4, se observa como la versión nativa de NewsApp tiene un mejor tiempo de respuesta con respecto a la basada en web. Se nota que el tiempo de respuesta medio no supera los 2 segundos de diferencia entre ambas versiones de NewsApp; además, se puede decir que estos tiempos son aceptables teniendo en cuenta que un tiempo de respuesta inferior a los 5 segundos es tolerable para el usuario[13]. Como se observa en la figura 4, la versión nativa de NewsApp se mantuvo por debajo de dicho margen de tiempo aún en el peor caso, mientras que la versión basada en web lo sobrepasó notablemente.

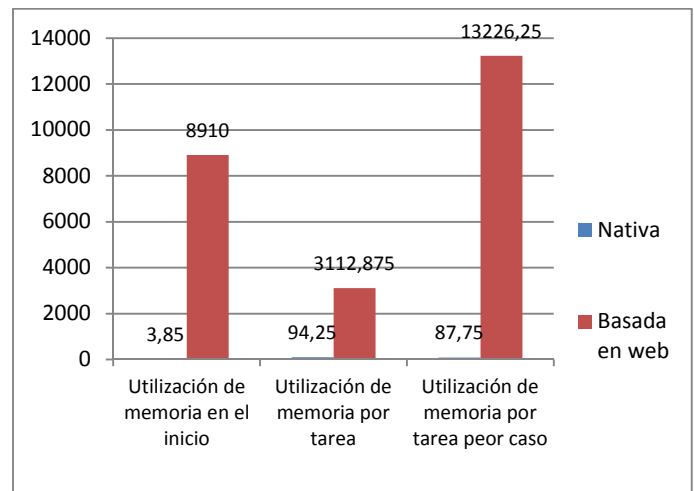


Figura 5. Utilización máxima de memoria en NewsApp nativa VS. Utilización máxima de memoria en NewsApp basada en web (kB).

Según la figura 5, la utilización máxima de memoria en el inicio de la aplicación, para el caso de NewsApp nativa consumió menor cantidad de memoria RAM que la versión basada en web tanto al inicio de la misma, como también por tarea ejecutada y en el peor caso; en otras palabras, en los tres casos mencionados la utilización de memoria de NewsApp nativa es menor frente al uso de este recurso en la otra versión de la aplicación.

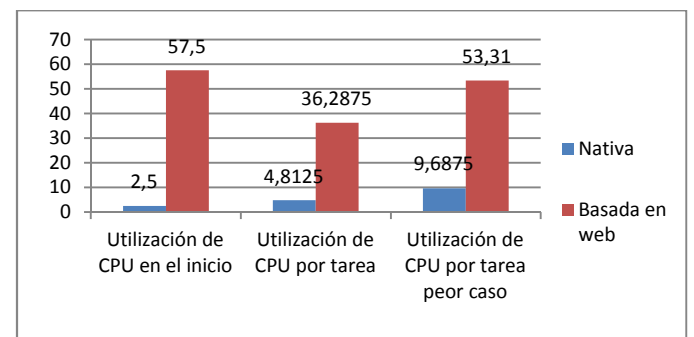


Figura 6. Utilización máxima de CPU en NewsApp nativa VS. Utilización máxima de CPU en NewsApp basada en web (kB).

En la figura 6, se aprecia que NewsApp basada en web tanto en el inicio de la aplicación como en el peor caso, hace una

* Velocidad de descarga medida con la aplicación móvil OOKLA Speed Test.

utilización de *CPU* por encima del 50%. Los picos de uso de *CPU* a los que se llegó en la versión basada en web son tan elevados que podrían generar problemas de rendimiento en el sistema. Al iniciar la NewsApp nativa se alcanzó a utilizar únicamente un 2,5% de *CPU*.

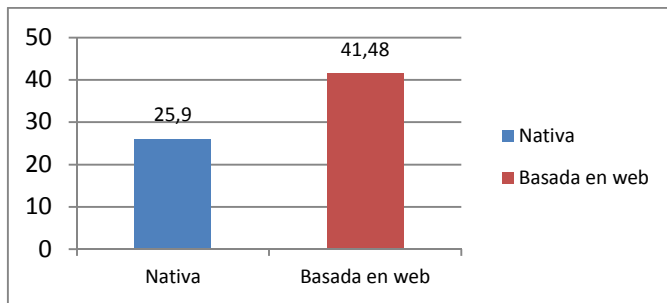


Figura 7. NewsApp nativa VS. NewsApp basada en web en cuanto a utilización máxima de batería (mAh).

Como indica la figura 7, NewsApp nativa consumió menos cantidad de miliamperios que la versión basada en web, de modo que esta última consumió 1,6 veces más de batería que la nativa. Teniendo en cuenta que la herramienta Little Eye usa como modelo de comparación el *Samsung Galaxy Nexus*, que posee una batería de 1750 mAh, idéntica a la del *Sony Xperia S* usado en la evaluación, se determinó que NewsApp basada en web consumió el 2,37% del total de la batería, mientras que NewsApp nativa utilizó el 1,48%.

IV. CONCLUSIONES

La confrontación de las dos tecnologías de desarrollo de aplicaciones móviles: nativa y basada en web, fue necesario soportarla en directrices que hicieron del presente trabajo un documento técnico, serio y de contenido apoyado en la realización de un procedimiento que se basó en los criterios de los estándares internacionales ISO/IEC 14598 e ISO/IEC 9126, de los cuales, el primero proveyó el modelo para el proceso de evaluación, y el segundo se empleó como modelo de calidad del producto. De acuerdo a las características de calidad que plantea el modelo ISO/IEC 9126, y considerándolas en el contexto de los *smartphones* y las aplicaciones móviles, se determinó la eficiencia como única característica a evaluar.

En la evaluación del rendimiento se analizaron: el tiempo de respuesta (métrica de comportamiento temporal), y utilización máxima de: memoria, *CPU* y batería, métricas pertenecientes a la subcaracterística utilización de recursos. Los resultados de esta evaluación sugieren que: el tiempo de respuesta es menor en la aplicación nativa que en la versión basada en web y la utilización de recursos tanto de memoria como de *CPU* y de batería, también es menor en la versión nativa de la aplicación móvil. En cuanto a la utilización de batería, cabe resaltar que el uso de este recurso en ambas versiones de la aplicación, es el que presenta una diferencia más pequeña.

Tras la confrontación de las dos tecnologías de desarrollo de aplicaciones móviles mediante los atributos de la calidad relacionados con la eficiencia, se confirmó que el desarrollo nativo genera aplicaciones de mejor rendimiento que el desarrollo basado en tecnologías web.

REFERENCIAS

- [1] Tommi Mikkonen and Antero Taivalsaari, "Cloud computing and its impact on mobile software development: Two roads diverged," *Journal of Systems and Software*, vol. 86, no. 9, pp. 2318-2320, Septiembre 2013.
- [2] Android Open Source Project. Android Developer. [Online].
<http://developer.android.com/guide/webapps/index.html>
- [3] Little Eye Labs. (2014) Little Eye. [Online].
<http://www.littleeye.co/index.html>
- [4] ISO/IEC Organization, Software engineering -- Product quality -- Part 1: Quality model, 2001.
- [5] ISO/IEC Organization, Information technology -- Software product evaluation -- Part 1: General overview, 1999.
- [6] Roger Pressman, *Ingeniería del Software un Enfoque Practico*. Mexico, D.F: McGraw-Hill, 2010.
- [7] Ian SommerVille, *Ingenieras de Software Séptima edición*. Madrid: Pearson Educación, 2005.
- [8] Ian Gorton, *Essential Software Architecture*, Segunda edición ed. Richland, USA: Springer, 2011.
- [9] El Cuera. MTP Software Quality Assurance. [Online].
<http://www.mtp.es/noticias/265-pruebas-calidad-aplicaciones-moviles>
- [10] Andrés Vivanco, Evaluación de la calidad del sistema integrado para casas de valores SICAV de la Bolsa de Valores de Quito utilizando la norma ISO/IEC 14598, Agosto 2011.
- [11] Android Open Source Project. Android Developer. [Online].
<http://developer.android.com/training/index.html>

- [12] Capgemini; Sogeti ; HP, "World Quality Report," Capgemini Sogeti y HP, Reporte 2012.
- [13] Apigee, "Apigee Survey: Users Reveal Top Frustrations That Lead to Bad Mobile App Reviews," Apigee, the API company, Palo Alto, Encuesta de opinión 2012.
- [14] Compuware Corporation, "Mobile Apps vs. Mobile Websites -- and the Winner Is?," Compuware Corporation, Deroit, Estudio 2013.
- [15] SOASTA. (2013, Junio) 88% of Americans Have Negative Feelings about Brands with Poorly Performing Websites and Mobile Apps. [Online].
<http://www.soasta.com/press-releases/467413/>
- [16] Craig Larman, *UML y Patrones*. Madrid: Pearson Educacion, S.A, 2002.
- [17] Darío Yorio, "Identificación y Clasificación de Patrones en el Diseño de Aplicaciones Móviles.," Universidad Nacional de La Plata, La Plata, Tesis Magíster 2006.
- [18] Developer Economics. (2013, Enero) PhoneGap and Appcelerator lead developer mindshare across tens of CPTs. [Online].
<http://www.developereconomics.com/phonegap-and-appcelerator-lead-developer-mindshare-across-tens-of-cpts/>
- [19] Simon Bennett, Steve McRobb, and Ray Farmer, *Análisis y Diseño Orientado a Objetos de Sistemas*. Madrid: McGraw-Hill, 2006.
- [20] Paco Blanco, Julio Camarero, Antonio Fumero, Adam Woterski, and Pedro Rodríguez. (2009) Metodología de desarrollo ágil para sistemas móviles. [Online].
http://www.adamwesterski.com/wp-content/files/docsCursos/Agile_doc_TemasAnv.pdf
- [21] Kenneth E Kendall and Julie E Kendall, *Análisis y diseño de sistemas. Sexta edición*. México: Pearson Educación, 2005.
- [22] Don Wells. (1999) user stories. [Online].
<http://www.extremeprogramming.org/rules/userstories.html>
- [23] Anywhere Software. (2014) Basic4android. [Online].
<http://www.basic4ppc.com/>
- [24] xamarin. (2014) Xamarin.Android. [Online].
<http://xamarin.com/android>
- [25] Android Open Source Project. Android NDK. [Online].
<https://developer.android.com/tools/sdk/ndk/index.html>
- [26] David Ehringer. (2010, Marzo) The dalvik virtual machine architecture. [Online].
http://davehringer.com/software/android/The_Dalvik_Virtual_Machine.pdf