

# Comparación de la Calidad de una Aplicación Móvil Desarrollada para Android y iOS: un Caso de Estudio

Sehyris Campos Arce<sup>1</sup>, Alejandro Mora<sup>1</sup> y Marcelo Jenkins<sup>2</sup>

<sup>1</sup> Posgrado de Computación e Informática Universidad de Costa Rica, San Pedro, Costa Rica  
{sehyris.campos, alejandro.moracastro}@ucr.ac.cr

<sup>2</sup> Escuela de Computación e Informática Universidad de Costa Rica, San Pedro, Costa Rica  
marcelo.jenkins@ucr.ac.cr

**Abstract.** Una empresa costarricense desarrolladora de software creó una aplicación móvil con funcionalidad equivalente para Android y iOS. Debido a ello surgió el deseo de evaluar la calidad del software desarrollado en cada plataforma. Objetivo: Este artículo compara dos plataformas de desarrollo con el fin de determinar el impacto que el uso de éstas tiene en la calidad del software en una aplicación desarrollada para Android y iOS desde el punto de vista de la administración del proyecto. Método: Se recopilieron tiquetes registrados en el sistema de administración de proyectos durante más de dos años para cada aplicación. Posteriormente, se realizaron mediciones de los datos, las cuales permitieron identificar de forma cuantitativa y cualitativa si la calidad del software desarrollado se vio influenciada por la plataforma de desarrollo. Resultados: El caso de estudio presenta como resultado principal que en la aplicación desarrollada en iOS se percibe una calidad menor que en Android. Además llama la atención la diferencia en la cantidad de defectos con severidad *Major* entre las plataformas. Conclusiones: Este estudio permitió realizar la evaluación de calidad de dos versiones de una aplicación móvil, sin embargo sus resultados no pueden ser generalizados por lo que se recomienda extender el estudio a más aplicaciones y más compañías, así como realizar una investigación para determinar las causas asignables de los resultados obtenidos.

**Keywords:** Calidad del software, plataforma móvil, aplicación móvil, Android, iOS.

## 1 Introducción

### 1.1 Definición del Problema

Una compañía costarricense de desarrollo de software cuenta en la actualidad con un cliente que posee una aplicación base sobre la cual se implementan nuevas características, y a partir de la misma se generan nuevas aplicaciones pero con diferentes personalizaciones. A nivel general, esta aplicación provee acceso a una serie de artículos científicos publicados en varias revistas médicas. El usuario puede leer y/o guardar

los artículos, compartirlos en redes sociales y visitar enlaces de interés para las revistas, entre otras tareas.

La aplicación ha sido desarrollada para las plataformas Android y iOS, y aunque ha estado vigente por varios años, no se han realizado estudios para determinar la calidad de la misma. Es por ello que surgió el interés por parte del administrador de calidad de proyectos de determinar si la calidad del software se ve afectada por la plataforma de desarrollo.

Este estudio corresponde a uno de los primeros análisis de la calidad de las aplicaciones desarrolladas y le permitirá a la empresa determinar la cantidad de incidentes y la severidad de los mismos durante el proceso de desarrollo, de ahí podrá tomar las medidas necesarias para mejorar la calidad de la aplicación dependiendo de la plataforma de desarrollo, como por ejemplo impartir cursos de programación de plataforma donde se haya encontrado una mayor cantidad de defectos.

En este caso de estudio se analizó la cantidad de defectos y mejoras registradas en cada una de las plataformas, con el objetivo de obtener una respuesta a la interrogante planteada. A partir de la información obtenida luego del análisis, se podrá apoyar los procesos de desarrollo y toma de decisiones en el futuro, cuando corresponda seleccionar y trabajar con plataformas móviles.

## **1.2 Objetivo de Investigación**

El objetivo de la investigación es comparar dos plataformas de desarrollo para determinar el impacto que el uso de éstas tiene en la calidad del software en una aplicación móvil desarrollada tanto para Android como iOS desde el punto de vista del administrador de calidad del proyecto.

## **1.3 Contexto de la Investigación**

La investigación analizó una misma aplicación para dos plataformas cuya funcionalidad es equivalente, creada en una empresa costarricense desarrolladora de software. Se analizaron defectos y mejoras registrados a lo largo de 38 meses (del 1/4/2011 al 31/5/2014). Se eligió ese periodo porque fue el que correspondió a la información registrada en el sistema de administración de proyectos de la organización. Por tratarse de datos históricos a los que los investigadores tuvieron acceso, no se requirió intervención de otras personas. Únicamente se hicieron las consultas necesarias en el sistema de administración de proyectos y se exportaron los datos para su posterior análisis.

## **2 Trabajo Relacionado**

En el contexto de aplicaciones móviles existen múltiples plataformas de desarrollo, tales como Android, iOS, Windows Phone y Blackberry, entre otras. Ante tal variedad de ambientes de desarrollo, en la literatura se han realizado comparaciones en distin-

tos ámbitos, que permiten conocer el comportamiento de las plataformas y determinar cuál se adapta mejor según las circunstancias del caso de estudio.

Zhang et al. [8] comparan Android, iOS y Windows Phone en términos de defectos relacionados con la administración de la energía del dispositivo. El artículo muestra que las tres plataformas son susceptibles a la introducción de defectos de este tipo por parte de los desarrolladores. En Android y Windows Phone existe mayor control en el uso de energía en las aplicaciones, mientras que iOS es más restrictivo y provee mínima flexibilidad al desarrollador.

Ahmad et al. [1] realizan una comparación entre Android y iOS en términos de seguridad, según las características: 1) *Sandboxing* de las aplicaciones, 2) Aleatorización de la memoria, 3) Encriptación, 4) Formato de almacenamiento de datos, y 5) Antivirus incorporado. Aunque es difícil hablar a favor de una plataforma u otra, los autores concluyen que iOS tiene ventaja en cuanto a seguridad se refiere.

Por otra parte, Alhamed et al. [2] comparan métodos para el control de la privacidad en Android y iOS, basándose en un *framework* de cuatro componentes: 1) Componente de captura, 2) Componente de construcción, 3) Componente de actividad, y 4) Componente de propósitos. Los resultados indican que ninguna de las dos plataformas desarrolla la totalidad del modelo, por lo que es posible encontrar defectos relacionados con el control de la privacidad en aquellas áreas donde un componente no ha sido implementado.

Syer et al. [6] comparan tres pares de aplicaciones cuya funcionalidad entre las plataformas Android y Blackberry es equivalente. Lo hacen en términos de las métricas: 1) Tamaño de código fuente, 2) Cantidad de dependencias, y 3) *Code churn*. De acuerdo con las aplicaciones analizadas en el caso de estudio, el tamaño del código Blackberry es más del doble que su equivalente Android, Blackberry tiene una mayor dependencia de librerías de terceros, y ambas plataformas presentan un alto grado de *code churn* como resultado de la rápida evolución de las aplicaciones.

En el presente trabajo se considera la calidad del software como "el grado en que un sistema, componente o proceso satisface los requerimientos especificados" [4] y se realiza una nueva comparación, esta vez evaluando la calidad del software producido con base en los defectos y mejoras reportados para una misma aplicación desarrollada en dos plataformas, Android y iOS.

La comparación se realizó mediante el análisis de tiquetes reportados en el sistema administración de proyectos, Jira (<https://www.atlassian.com/es/software/jira>) en este caso. Un tiquete es un ítem almacenado en la base de datos del sistema, que tiene estados de inicio y finalización, estados opcionales de transición y que almacena información a lo largo de su ciclo de vida [3].

Jira permite administrar distintos tipos de tiquetes, tales como defectos, mejoras, nuevas funcionalidades, tareas y tipos personalizados. Se tomaron en cuenta dos tipos de tiquetes: defectos y mejoras. Un defecto es un problema que dificulta o impide el correcto funcionamiento del software, mientras que una mejora es el progreso de una funcionalidad existente.

Asimismo, es posible asignar a los tiquetes distintos niveles de severidad, que por defecto son: *Trivial*, *Minor*, *Major*, *Critical* y *Blocker*. El nivel de severidad de cada

tiquete es asignado por el *product owner* y/o el administrador del proyecto, dependiendo del impacto que el incidente genere en el funcionamiento de la aplicación.

### **3 Diseño del Estudio Empírico**

#### **3.1 Preguntas de Investigación**

Para desarrollar este estudio se plantearon las siguientes preguntas de investigación:

- P1: ¿Existe una diferencia significativa entre la cantidad de defectos registrados en una plataforma y la otra?
- P2: ¿Existe una diferencia significativa entre la cantidad de mejoras registradas en una plataforma y la otra?
- P3: ¿Cuál es la diferencia en la densidad de defectos entre una plataforma y la otra?
- P4: ¿Cuál es la diferencia en la cantidad total de defectos registrados por nivel de severidad en una plataforma y la otra?

#### **3.2 Selección del Tipo de Estudio y los Sujetos de Investigación**

Para llevar a cabo esta investigación se desarrolló un caso de estudio, en el cual los sujetos de investigación fueron dos versiones de una aplicación móvil con funcionalidades idénticas, pero desarrolladas en dos plataformas diferentes. Ambas versiones se encuentran en el mismo estado, es decir, cuentan con la misma funcionalidad. Además, el periodo analizado en este estudio corresponde al inicio y fin de las versiones tanto de iOS como Android.

Las herramientas para el desarrollo de cada versión dependen del sistema operativo, para iOS se usa Xcode y para Android inicialmente se utilizaba Eclipse con los respectivos *plugins* posteriormente se pasó a utilizar Android Studio. Las diferentes herramientas para cada plataforma son un factor de variabilidad que puede afectar el resultado final, sin embargo, no fue considerado de esta manera dado que todos los desarrolladores manejan de igual forma dichas herramientas.

El proceso de desarrollo en la empresa a la que pertenecen las aplicaciones tiene la particularidad de que los integrantes de cada proyecto son intercambiados periódicamente, con el fin de que el conocimiento y la experiencia en cada plataforma no se concentren en determinadas personas, por lo que se busca que todos los desarrolladores de aplicaciones móviles tengan un conocimiento similar de cada ambiente y así estén en capacidad de encarar en cualquier momento los proyectos actuales o bien futuros.

#### **3.3 Procedimiento de Recolección de Datos**

Para responder a las preguntas de investigación se extrajeron los datos del sistema de administración de proyectos Jira. A continuación se detallan las tareas realizadas para la recolección de los datos:

**Consultas al Sistema de Administración de Proyectos.** Ambas versiones de la aplicación (de cada plataforma) se encuentran registradas en el mismo sistema. Para la obtención de los tickets se utilizaron consultas JQL (*Jira Query Language*).

**Ajuste de Filtros.** Dadas las restricciones de confidencialidad requeridas para la investigación, se aplicaron filtros en las columnas para no obtener información sensible. En este caso se usaron los filtros de la búsqueda avanzada del sistema, y solo se extrajo la siguiente información por ticket: identificador, tipo (defecto o mejora), fecha de creación, nivel de severidad y plataforma.

**Limpieza de Datos.** Adicional a las consultas y filtros se realizó una revisión manual de los tickets para limpiarlos. Por ejemplo, se descartaron aquellos tickets resueltos como *Duplicate*, ya que esto implicaba la existencia de otro ticket que trataba el mismo defecto o mejora. Asimismo, se descartaron registros que no contenían información necesaria como tipo (defecto o mejora) o plataforma, siempre y cuando el desarrollador no fuera capaz de identificar dicha información basado en el contexto del ticket, como se describe en la sección de procedimiento de validación de datos. También se realizó una inspección sobre el nivel de severidad, y cuando un ticket no lo especificó, se le asignó por defecto el nivel de *Trivial*.

Las consultas al sistema de administración de proyectos se realizaron por un investigador del caso de estudio y se exportaron los resultados a archivos de Excel sobre los cuales se realizó el análisis de datos. Las consultas estuvieron limitadas por el período del 1/4/2011 al 31/5/2014.

### 3.4 Procedimiento de Validación de Datos

Los datos fueron extraídos del sistema de administración de proyectos, sin embargo, como esta información es introducida por humanos es probable que existan errores, por lo que se aplicaron los siguientes pasos para validar los datos:

- En caso de que se omitiera información como la plataforma o tipo de incidencia (defecto o mejora), se utilizó el criterio experto de uno de los desarrolladores que conoce el contexto de la aplicación y posee acceso al sistema, por lo que se podría revisar el registro y completar los datos de acuerdo con la información suministrada.
- En caso de que no se suministrara el nivel de severidad de un ticket, se asumió con el nivel más bajo (*Trivial*), para no afectar significativamente las demás estadísticas.
- Los demás datos extraídos del sistema como el identificador y fecha de creación, se asumieron como válidos para todos los casos, debido a que son generados por el sistema.

## 4 Descripción de Resultados

### 4.1 Análisis de Resultados

A continuación se describe el procedimiento de análisis de resultados para cada una de las preguntas de investigación de este caso de estudio.

**Medición de cuán significativas son las comparaciones.** Se obtuvo una medición para cada plataforma y luego se procedió a determinar si la diferencia fue lo suficientemente significativa para establecer una conclusión. Al ser una comparación entre dos números, no existe un método estadístico que permita determinar cuán significativa es esa diferencia, por lo que se utilizó el juicio experto de acuerdo con los intereses de la organización. Según la experiencia de uno de los *QA Manager* de la empresa (a quien se le consultó antes de realizar el estudio), para este caso en particular se considera que hay una diferencia significativa si el resultado obtenido al efectuar la operación:

$$\text{diferencia} = 1 - \frac{\text{valor de la medición menor}}{\text{valor de la medición mayor}}$$

es mayor o igual a 0,25, según el razonamiento que se describe a continuación. Si las cantidades medidas fueran muy similares, la diferencia calculada tendería a 0 y la coincidencia de las mediciones se incrementaría. Si las cantidades fueran considerablemente distintas, la diferencia calculada tendería a 1 y la coincidencia de las mediciones disminuiría. En este caso, el umbral 0,25 indica que si el valor más alto medido en una plataforma es  $n$ , en la otra se midieron  $3/4$  de  $n$ , es decir, diferencias mayores a una cuarta parte se consideran significativas.

**P1: ¿Existe una diferencia significativa entre la cantidad de defectos registrados en una plataforma y la otra?.** Esta pregunta permitió medir la calidad de forma cuantitativa. A mayor cantidad de defectos menor es la calidad. Para responder a esta pregunta se contabilizó la cantidad total de defectos registrados en cada una de las plataformas de desarrollo y para determinar cuán significativa es la diferencia, se calculó la proporción de los defectos registrados entre ambas plataformas de la siguiente manera:

$$\text{diferencia} = 1 - \frac{\text{menor cantidad de defectos}}{\text{mayor cantidad de defectos}}$$

La **Tabla 1** muestra un resumen de la cantidad de defectos registrados en Android y iOS, así como el cálculo de la proporción. Dado que la proporción de defectos encontrada fue de 0,34 se considera que sí existe una diferencia significativa entre ambas plataformas. Por cada  $n$  defectos que se encuentran en iOS, en Android se reportan cerca de  $2/3$  de  $n$ , por lo que se percibe que la calidad de la aplicación Android es mayor.

**Tabla 1.** Cálculo de la proporción de los defectos registrados entre las plataformas.

<i>Cantidad de defectos en Android</i>	346
<i>Cantidad de defectos en iOS</i>	523
<i>Diferencia de los defectos registrados</i>	$1 - (346 / 523) = 0,34$

**P2: ¿Existe una diferencia significativa entre la cantidad de mejoras registradas en una plataforma y la otra?.** Esta pregunta permitió medir la calidad de forma cuantitativa. Mientras mayor sea la cantidad de mejoras en una plataforma, menor es la calidad percibida. Para responder a esta pregunta se contabilizó la cantidad total de mejoras registradas en cada una de las plataformas de desarrollo y para determinar cuán significativa es la diferencia, se calculó la proporción de las mejoras registradas entre ambas plataformas de la siguiente manera:

$$diferencia = 1 - \frac{\text{menor cantidad de mejoras}}{\text{mayor cantidad de mejoras}}$$

La Tabla 2 presenta la proporción de las mejoras registradas en la plataforma Android y iOS, así como la cantidad de mejoras registradas en cada una de ellas. Dado que la proporción de las mejoras registradas fue de 0,38 se considera que sí existe una diferencia significativa, lo que permite percibir la calidad de la aplicación de Android como de menor grado que iOS.

**Tabla 2.** Cálculo de la proporción de las mejoras registradas entre las plataformas.

<i>Cantidad de mejoras en Android</i>	104
<i>Cantidad de mejoras en iOS</i>	65
<i>Proporción de las mejoras registradas</i>	$1 - (65 / 104) = 0,38$

Con la pregunta de investigación P1 se concluye que en iOS la cantidad de defectos es considerablemente mayor que en Android. Sin embargo, con esta pregunta de investigación se obtiene lo contrario, es decir, se realiza una mayor cantidad de mejoras al código en Android en comparación con iOS. Podría resultar de interés de indagar los motivos por los cuales ocurren estas diferencias y proponer mejoras en el proceso de desarrollo para disminuir la cantidad de mejoras sin impactar negativamente la calidad del producto independientemente de la plataforma.

**P3: ¿Cuál es la diferencia en la densidad de defectos entre una plataforma y la otra?.** Esta pregunta permitió medir la calidad de forma cuantitativa. Se compararon las densidades de defectos entre las plataformas, como una medida de la calidad.

Según Thomas [7], contar el número de defectos en un software no es suficiente para medir su calidad, mientras que la densidad de defectos es un estándar de la industria, y permite conocer la proporción entre el número de defectos y el tamaño del software. Para medir el tamaño de cada aplicación, se utilizaron KLLOCs (miles de líneas de código lógicas por sus siglas en inglés). Se empleó una utilidad llamada Project Code Meter ([http://www.projectcodemeter.com/cost\\_estimation/index.html](http://www.projectcodemeter.com/cost_estimation/index.html)) para automatizar el proceso. La densidad de defectos se calculó de la siguiente manera:

$$\text{densidad} = \frac{\text{cantidad de defectos reportados en el proyecto } i}{\text{número de KLLOCs del proyecto } i}$$

Con las densidades de defectos medidas, el cálculo de la diferencia se hizo de la siguiente manera:

$$\text{diferencia} = 1 - \frac{\text{menor densidad de defectos}}{\text{mayor densidad de defectos}}$$

La **Tabla 3** muestra las densidades de defectos calculadas para cada plataforma. Se contabilizaron 10,08 y 13,83 defectos por cada mil líneas de código en Android y iOS, respectivamente. Según la fórmula mencionada anteriormente para determinar si la diferencia es significativa se obtiene 0,27.

El valor 0,27 es mayor al umbral definido (0,25), por lo que la diferencia se considera significativa (la menor densidad de defectos es poco menos de 3/4 de la mayor). Por lo tanto, a partir de ambas mediciones se percibe que la aplicación iOS es más propensa a fallos que su equivalente Android.

**Tabla 3.** Cálculo de las densidades de defectos de las plataformas.

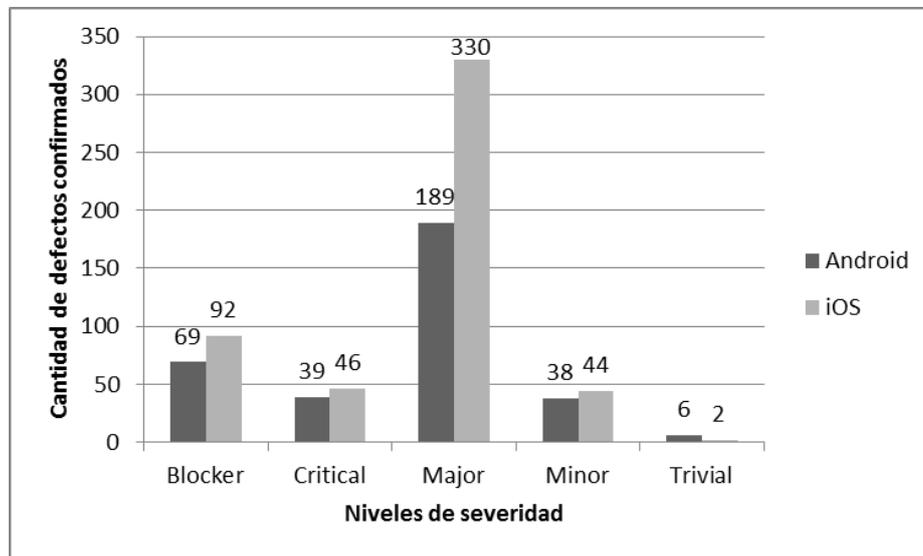
	<b>Android</b>	<b>iOS</b>
<i>Cantidad de archivos</i>	391	379
<i>Miles de líneas de código lógicas (KLLOCs)</i>	33.827	37.172
<i>Cantidad de defectos confirmados</i>	341	514
<b><i>Densidad de defectos</i></b>	<b>10,08</b>	<b>13,83</b>

**P4: ¿Cuál es la diferencia en la cantidad total de defectos registrados por nivel de severidad en una plataforma y la otra?** Esta pregunta permitió medir la calidad de forma cualitativa y cuantitativa. Aún cuando puede haber cantidades relativamente bajas de defectos, se debe considerar el nivel de severidad de éstos como una medida de la calidad. Se utilizó un gráfico de columnas agrupadas, el cual permite la comparación de categorías utilizando rectángulos verticales. El eje X representa las categorías de severidad de los defectos, mientras que el eje Y contiene el número de defectos

en esa categoría. Cada par de columnas está dado por el número de defectos por categoría en cada aplicación.

La Figura 1 muestra la cantidad de defectos confirmados según los niveles de severidad para cada plataforma. En cuatro de los cinco niveles de severidad, la aplicación iOS presenta mayor cantidad de defectos, la única excepción se presenta en los defectos de tipo *Trivial*. En las categorías de menor severidad (*Trivial* y *Minor*) los defectos se encuentran distribuidos de forma casi idéntica entre las plataformas (Android tiene 44 defectos y iOS tiene 46), sin embargo, en las categorías de severidad más alta la diferencia se hace considerablemente mayor. Sumando los defectos *Mayor*, *Critical* y *Blocker*, se tiene que hay 297 defectos en Android y 468 en iOS. Al calcular la diferencia según la fórmula se obtiene 0,365, por lo tanto, la diferencia para esos niveles de severidad se considera significativa. Los datos indican entonces que los defectos reportados en iOS no solo son más (punto de vista cuantitativo), sino que son más severos (punto de vista cualitativo).

**Fig. 1.** Comparación de cantidades de defectos confirmados según los niveles de severidad por plataforma.



#### 4.2 Asuntos y/o Problemas con el Análisis e Interpretación de los Datos

Durante el desarrollo de este caso de estudio se encontró el siguiente problema de análisis e interpretación de datos:

- Para el cálculo de las líneas de código de cada aplicación se notó que podían haber problemas entre las plataformas, por ejemplo los comentarios no se manejan igual o incluso el código está formateado de diferente forma por lo que se podía afectar el conteo. Para solucionar estos problemas se buscó una herramienta que permitiera contar líneas de código lógicas y no físicas, con el fin de disminuir el ruido genera-

do por este factor. En este estudio se utilizó la herramienta Project Code Meter en ambas plataformas. Según Nguyen et al. [5], usar líneas de código lógicas tiene la ventaja de que se contabilizan sentencias, las cuales son independientes del formato físico del código analizado (múltiples sentencias podrían localizarse en una línea física o una sentencia podría estar distribuida en varias líneas físicas).

### 4.3 Evaluación de la Validez del Estudio

*Validez de Conclusion.* Para realizar el análisis se utilizó sólo un par de versiones de la misma aplicación, por lo que la generalización de los datos no es posible.

Además, la evaluación de qué tan significativas fueron las diferencias de las mediciones entre plataformas está basada en el criterio experto según el interés de la empresa donde se desarrolló el estudio, por lo que los resultados podrían no ser trasladables a otras organizaciones.

*Validez Interna.* Para realizar el caso de estudio los sujetos seleccionados se escogieron porque eran los disponibles en el momento. Es posible que si se hubieran seleccionado otros sujetos los resultados habrían variado.

*Validez de Constructo.* Al iniciar el experimento, los investigadores establecieron como hipótesis que la calidad percibida en iOS sería menor, basados en el juicio experto. Es posible que tales consideraciones hayan afectado el análisis de los resultados de forma inconsciente.

*Validez Externa.* Los resultados se obtuvieron a partir de aplicaciones desarrolladas en una empresa en particular, por lo que no pueden ser necesariamente generalizadas para todas las organizaciones.

## 5 Conclusiones y Trabajo Futuro

### 5.1 Sumario de Hallazgos

El caso de estudio muestra como resultado principal que en la aplicación desarrollada en iOS se percibe una calidad menor que en su equivalente Android. iOS presenta no sólo una cantidad de defectos significativamente mayor, sino que la severidad de éstos también es más alta.

Llama particularmente la atención la diferencia en la cantidad de defectos con severidad *Major*, en la que se registran 189 defectos en Android y 330 en iOS. Luego de analizar el proceso de recolección de tickets en Jira, se notó que en ambas plataformas los defectos se reportan con severidad *Major* por omisión, así que la diferencia tan grande en ambas mediciones podría verse afectada por ello.

En una de las cuatro preguntas de investigación (P2) se concluyó que la calidad percibida en la aplicación Android es menor debido a que el número de mejoras introducidas en ella fue significativamente mayor. Sin embargo, esto bien podría ser un factor que por el contrario permitió que en esa plataforma aparecieran menos defec-

tos. Se necesita recopilar más información relacionada con el fin de establecer una conclusión válida.

Al realizar la investigación no se encontró un estudio igual, por lo que se concluye que el caso de estudio desarrollado es novedoso y que abre camino para plantear futuros trabajos sobre este tema. No se encontraron otros casos de estudio en los que se compare la calidad entre las plataformas estudiadas. Otros estudios en la literatura existente realizan comparaciones de plataformas móviles en términos de administración de la energía [8], seguridad [1] y controles de privacidad [2], los cuales no se relacionan directamente con el presente trabajo. El estudio que más se asemeja es [6], pues los autores comparan tres pares de aplicaciones cuya funcionalidad entre las plataformas Android y Blackberry es equivalente, sin embargo lo hacen en términos de otras métricas (tamaño de código fuente, cantidad de dependencias y *code churn*).

Los resultados son útiles para la empresa porque le permiten identificar áreas con mayor propensión a fallos durante el proceso de desarrollo. En este caso, la aplicación iOS presenta una cantidad de defectos significativamente mayor que su contraparte Android, por lo que resultaría útil conocer a qué se debe tal disparidad para corregirla, tomando en cuenta que la funcionalidad desarrollada es la misma.

Por otro lado, los resultados arrojan evidencia suficiente para que otros investigadores en el área de Ingeniería de Software se vean motivados a desarrollar más trabajo con el fin de generalizar o refutar los hallazgos, es decir, se abrió espacio para desarrollar trabajo futuro.

Este caso de estudio se limitó al análisis de dos versiones de una aplicación móvil con funcionalidades idénticas con el objetivo de determinar el impacto que tiene el uso de dos plataformas diferentes de desarrollo en la calidad del software. El estudio se llevó a cabo en una empresa de software costarricense ya que era donde se tenía la autorización para la extracción de la información anteriormente mencionada. Por consiguiente, este estudio no presenta una muestra aleatoria de los sujetos de investigación sino que se escogieron a conveniencia de los investigadores.

Las conclusiones de este estudio son resultados preliminares de un primer estudio empírico, en el cual solo se analizaron los incidentes reportados en el sistema de administración de proyectos durante un periodo de tiempo donde tanto la aplicación de Android como iOS contaban con la misma funcionalidad. Otros factores como el tiempo de desarrollo de determinadas funciones, complejidad de desarrollo de las tareas dependiendo de la plataforma, diferentes ambientes y herramientas de desarrollo estuvieron presentes, pero no consideramos que hayan influido considerablemente en los resultados obtenidos.

## 5.2 Trabajo Futuro

Sería interesante realizar otra investigación que incluya una mayor cantidad de pares de aplicaciones provenientes de diferentes compañías. Con ello se tendría una muestra mayor que permitiría confirmar (y eventualmente generalizar) o refutar los resultados obtenidos. En la actualidad es muy común que las organizaciones que desarrollan una aplicación en una plataforma también lo hagan en la otra, por lo que conseguir más pares de aplicaciones es factible.

Finalmente, para tres de las cuatro preguntas de investigación se obtuvo que en la plataforma iOS se generan más defectos que en la plataforma Android. Puede resultar de interés para futuras investigaciones o de la misma empresa donde se realizó el estudio tratar de determinar las causas asignables de estos resultados, para así buscar medidas que permitan mitigar dichas diferencias.

## Referencias

1. Ahmad, M., Musa, N., Nadarajah, R., Hassan, R., & Othman, N. (2013). Comparison Between Android and iOS Operating System in terms of Security. *2013 8th International Conference on Information Technology in Asia (CITA)* (pp. 1 - 4). Kota Samarahan: IEEE.
2. Alhamed, M., Amiri, K., Omari, M., & Le., W. (2013). Comparing Privacy Control Methods for Smartphone Platforms. *2013 1st International Workshop on the Engineering of Mobile-Enabled Systems (MOBS)* (pp. 36 - 41). San Francisco, CA: IEEE.
3. Fogel, K. (2013). *Producing Open Source Software*. Retrieved from Bug Tracker: <http://producingoss.com/en/bug-tracker.html>
4. IEEE. (1990). IEEE Standard Glossary of Software Engineering . *IEEE Std. 610.12-1990*. Institute of Electrical .
5. Nguyen, V., Dees-Rubin, S., Tan, T., & Boehm, B. (2007). A SLOC Counting Standard. *22nd International Annual Forum on COCOMO® and System/Software Cost Modeling*. Los Angeles, CA.
6. Syer, M., Adams, B., Zou, Y., & Hassan, A. (2011). Exploring the Development of Micro-apps: A Case Study on the BlackBerry and Android Platforms. *2011 11th IEEE International Working Conference on Source Code Analysis and Manipulation (SCAM)* (pp. 55 - 64). Williamsburg, VI: IEEE.
7. Thomas, P. (2011). *Defect Density Measurement*. Retrieved from <http://www.ifpug.org/ISMA6/Thomas-Defect%20Density%20Measurement-Sept14.pdf>
8. Zhang, J., Musa, A., & Le., W. (2013). A Comparison of Energy Bugs for Smartphone Platforms. *2013 1st International Workshop on the Engineering of Mobile-Enabled Systems (MOBS)* (pp. 25-30). San Francisco, CA: IEEE.