

**APRENDIENDO A DESARROLLAR APLICACIONES PARA  
ANDROID CON LA METODOLOGÍA ÁGIL SCRUM:  
UN CASO DE ESTUDIO**

**Área de investigación: Educación en Contaduría,  
Administración e Informática**

**Ramón Ventura Roque Hernández**

Facultad de Comercio, Administración y Ciencias Sociales  
Universidad Autónoma de Tamaulipas  
México  
[rvhernandez@uat.edu.mx](mailto:rvhernandez@uat.edu.mx)

**Eloy Negrete Hoz**

Facultad de Comercio, Administración y Ciencias Sociales  
Universidad Autónoma de Tamaulipas  
México  
[negrete.oz@hotmail.com](mailto:negrete.oz@hotmail.com)

**Juan Manuel Salinas Escandón**

Facultad de Comercio, Administración y Ciencias Sociales  
Universidad Autónoma de Tamaulipas  
México  
[jmsalinas@uat.edu.mx](mailto:jmsalinas@uat.edu.mx)

**XVIII**  
**CONGRESO**  
**INTERNACIONAL**  
**DE**  
**CONTADURÍA**  
**ADMINISTRACIÓN**  
**E**  
**INFORMÁTICA**



Octubre 2, 3 y 4 de 2013 ♦ Ciudad Universitaria ♦ México, D.F.



**ANFECA**  
Asociación Nacional de Facultades y  
Escuelas de Contaduría y Administración

## APRENDIENDO A DESARROLLAR APLICACIONES PARA ANDROID CON LA METODOLOGÍA ÁGIL SCRUM: UN CASO DE ESTUDIO

### Resumen

En este artículo se presenta un caso de estudio en el que se realizó un ejercicio experimental de desarrollo de software con la finalidad de que los programadores involucrados, todos ellos estudiantes de carreras profesionales de sistemas e informática, pusieran en práctica las herramientas necesarias para el desarrollo básico de aplicaciones móviles utilizando una metodología ágil. En la investigación se estudió a los programadores en su proceso de aprendizaje y aceptación de la metodología ágil Scrum durante el desarrollo de software. La implementación de Scrum como metodología ágil ampliamente adaptable en la teoría se llevó al desarrollo práctico de una aplicación móvil con fines didácticos creada con el lenguaje de programación Java para el sistema operativo Android. Al final de este experimento se aplicó una encuesta a cada uno de los participantes con el objetivo de analizar su postura, su grado de motivación sobre la metodología y la posible aplicación de lo aprendido en proyectos futuros. En este trabajo se presentan los resultados obtenidos y también se reflexiona sobre las ventajas y desventajas percibidas por los participantes e investigadores.

**Palabras clave:** Educación, Software, Agilidad

### Abstract

This paper presents a case study in software development that was conducted to implement practical tools for basic development of mobile applications using an agile methodology. The involved programmers were bachelor students with major in Computer Systems or Computing. The research studied the programmers in their learning process and acceptance of the Scrum agile methodology for software development. Scrum, widely adaptable in the theory, was implemented in practice to develop a mobile application with teaching purposes; Java programming language and the Android operating system were used. At the end of this experiment, a survey was conducted with the participants with the aim of analyzing their posture, degree of motivation on the methodology and the application of Scrum in future projects. In this paper, results are presented on the advantages and disadvantages perceived by participants and researchers.

**Key words:** Education, Software, Agility



## APRENDIENDO A DESARROLLAR APLICACIONES PARA ANDROID CON LA METODOLOGÍA ÁGIL SCRUM: UN CASO DE ESTUDIO

### 1. Introducción

En este artículo se presenta un caso de estudio en el que se realizó un desarrollo experimental de software con la finalidad de que los programadores involucrados, todos ellos estudiantes de carreras profesionales de sistemas e informática, pusieran en práctica las herramientas necesarias para el desarrollo básico de aplicaciones móviles utilizando una metodología ágil. El grupo de programadores participantes con habilidades diversas trabajaron en conjunto para la obtención de una aplicación móvil en un corto tiempo de entrega. El desarrollo se enfocó en la programación de una aplicación con fines didácticos utilizando Java para el sistema operativo Android siguiendo la metodología ágil denominada Scrum. El artículo está estructurado de la siguiente manera: los antecedentes se abordan en el apartado 2. La descripción del desarrollo realizado, las características del producto y el cuestionario que se aplicó se presentan en la sección 3. Los resultados obtenidos desde las perspectivas de investigadores y desarrolladores, así como la discusión de los mismos se exponen en el apartado 4. Finalmente se abordan las conclusiones y el trabajo futuro en la sección 5.

### 2. Antecedentes

#### 2.1 Los casos prácticos de estudio como herramienta de aprendizaje

Un caso práctico es una herramienta muy útil para el aprendizaje ya que muestra un panorama representativo de la situación que está siendo estudiada e involucra elementos que retan al alumno a poner en práctica sus conocimientos y habilidades para la resolución de problemas específicos. Se ha identificado a los casos prácticos como un medio de investigación (Kitchenham, Pickard, & Lawrence Pfleeger, 1995) pero también de enseñanza eficiente de la Ingeniería de Software que permite conocer la industria real del desarrollo de sistemas (Varma & Garg, 2005).

Contrario al enfoque tradicional de enseñanza de la Ingeniería de Software, en donde el maestro es la fuente principal de conocimientos, los casos prácticos plantean situaciones propicias para el aprendizaje en un terreno multidimensional en donde el alumno se involucra en escenarios que le permiten aplicar sus conceptos teóricos a nuevas situaciones para generar nuevas experiencias. Con el uso de casos prácticos se busca formar estudiantes capaces de unir sus conocimientos y habilidades personales con los sucesos que acontecen en su entorno; también se pretende motivarles a emprender actividades de investigación en diversos niveles.



## 2.2 Scrum: una metodología ágil

La palabra Scrum proviene del nombre de una jugada que ocurre en los partidos de rugby (Pressman, 2010). Scrum (Sims & Johnson, 2011) es una metodología ágil en la cual se llevan a cabo una serie de prácticas iterativas cuyo objetivo es que el grupo de desarrolladores trabaje unido, contribuyendo con sus habilidades individuales para la obtención de un software de buena calidad. Una de las características de Scrum es la entrega de porciones incrementales del producto final al término de cada iteración; de esta manera el cliente puede ir haciendo modificaciones o continuar con el desarrollo del software tal como se tenía previsto originalmente. Scrum es una metodología diseñada para el desarrollo de productos en ambientes complejos en donde se requiere un producto funcional rápidamente, con cambios constantes o con especificaciones ambiguas.

En un equipo de Scrum existen tres diferentes roles: el propietario del producto, el Scrum master y los miembros del equipo (Ver Tabla I). Dentro del marco general de Scrum existen reuniones que ayudan a la planeación del desarrollo en donde las experiencias adquiridas por los desarrolladores son muy relevantes, por lo que la retroalimentación y la recopilación de historias son fundamentales (Ver Tabla II).

El Sprint es el ciclo fundamental (iteración) del proceso de Scrum (Ver Figura 1). Scrum no especifica la longitud del sprint de su equipo, pero se considera que cuatro semanas es el tiempo máximo. Dos semanas es la longitud más frecuente, aunque también es común encontrar sprints con longitud de entre una y tres semanas.

Tabla I

Rol	Descripción
Propietario del producto (Product Owner)	Se asegura de que las necesidades de los clientes y usuarios finales son comprendidas por el equipo. Hace esto directamente creando, perfeccionando y comunicando los requisitos
Maestro Scrum (Scrum Master)	Una de sus principales funciones es la implementación de cada una de las características de Scrum, también es un facilitador el cual se asegura de eliminar los problemas que pueda llegar a tener el equipo.
Equipo de desarrollo	Los equipos son altamente colaborativos, también se auto-organizan. Los miembros del equipo tienen autoridad total sobre cómo se hace el trabajo. Las personas que hacen el trabajo son las más autorizadas para decidir la mejor forma de hacerlo.

Tabla I. Roles principales en Scrum



Tabla II

Reunión	Descripción
Scrum diario (Daily Scrum)	Es una reunión diaria generalmente al inicio del periodo de trabajo, que se puede adaptar a las necesidades del equipo. De manera breve cada participante comparte : <i>Lo que ha completado desde el último Daily Scrum.</i> <i>Lo que espera completar para el siguiente Daily Scrum.</i> <i>Los obstáculos que ha tenido durante el desarrollo.</i>
Scrum de Scrum	Esta reunión ocurre al finalizar el <i>Daily Scrum</i> ; acuden los líderes de grupos especializados en un área de desarrollo. Cada participante comparte: <i>Lo que ha completado su equipo desde la última reunión.</i> <i>Lo que espera que complete su equipo para la siguiente reunión.</i> <i>Los obstáculos que ha tenido su equipo.</i>
Reunión de planeación del Sprint (Sprint Planning Meeting)	El <i>Sprint Planning Meeting</i> marca el principio del sprint. Comúnmente tiene dos partes. El objetivo de la primera parte es que el equipo se comprometa a un conjunto de metas para el sprint. Durante la segunda parte de la reunión, el equipo identifica las tareas que deben realizarse en el orden de las historias de usuario acordadas.
Reunión de revisión del Sprint (Sprint Review Meeting)	Al final del sprint, el equipo tiene la oportunidad de mostrar su trabajo en el <i>Sprint Review</i> . El equipo muestra las historias que fueron completadas y las que faltaron por completar. En esta etapa es cuando el propietario del producto toma nota del trabajo realizado, revisando las características solicitadas por el cliente y evaluando si es necesario realizar cambios o agregar nuevas características.
Retrospectiva (Retrospective)	La retrospectiva se lleva a cabo al final de cada sprint, consiste en dedicar tiempo para que el equipo se enfoque en lo que se ha aprendido durante el sprint; esta retroalimentación puede ser útil para hacer algunas mejoras.

Tabla II. Reuniones en Scrum

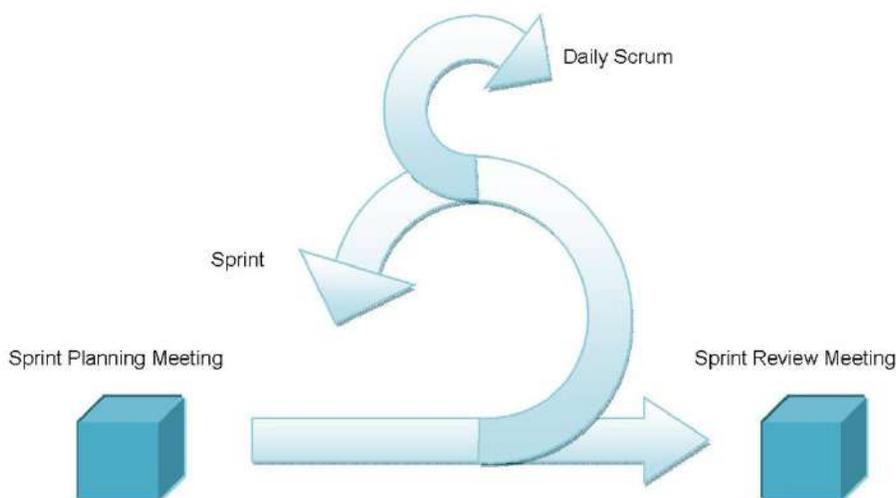


Figura 1. Visión general de Scrum.

### 2.3 Android como plataforma móvil

Android (Gironés, 2010) es una solución completa de software de código libre desarrollado sobre la plataforma Linux para teléfonos y dispositivos móviles: incluye un sistema operativo, un entorno de ejecución basado en Java, un conjunto de librerías de bajo y medio nivel y un conjunto inicial de aplicaciones destinadas al usuario final. Android es uno de los sistemas operativos móviles más utilizados actualmente en el mercado mundial. La versión

más reciente es la 4.2 y es conocida como Jelly Bean; fue lanzada para el final del año 2012 y posee una interfaz gráfica más funcional y un mejor desempeño con el hardware en comparación con sus antecesoras.

La programación de aplicaciones para el sistema operativo Android se realiza principalmente en Java y C++; éstos son los lenguajes oficiales de la plataforma propuestos desde su lanzamiento. El lenguaje Java, por su parte, permite la creación de aplicaciones que se ejecutan sobre la máquina virtual Dalvik, la cual fue especialmente optimizada para un buen desempeño en este sistema operativo al tomar en cuenta la limitante de recursos computacionales en los dispositivos móviles. Por otra parte, el Lenguaje C++ permite el desarrollo de aplicaciones nativas para Android. También existen otras alternativas de lenguajes de programación como por ejemplo, Basic para Android (Basic 4 Android, 2013) y Mono (Xamarin, 2013) con lenguajes de la plataforma .Net; las versiones completas de estos productos tienen costo y si bien cuentan con versiones gratuitas, éstas son limitadas en funcionalidad. Por el contrario, Java y C++ son alternativas de libre uso en cuanto a herramientas de desarrollo y compilación para Android se refiere.

#### **2.4 Implementación de metodologías ágiles en el desarrollo de aplicaciones móviles.**

El desarrollo de aplicaciones móviles puede beneficiarse del uso de metodologías ágiles ya que da respuesta a un mercado de usuarios con necesidades que cambian constantemente. El corto tiempo disponible para la obtención de un producto final es otra razón que justifica el uso de la agilidad pues pone a prueba la capacidad del equipo de trabajo para entregar software funcional de buena calidad. En el mercado, las aplicaciones móviles y sus actualizaciones deben estar disponibles rápidamente; en muchas ocasiones, incluso, cuidando de cerca los pasos de la competencia para lograr una ventaja estratégica sobre ella.

Diversos trabajos en los últimos años han identificado que el desarrollo de aplicaciones móviles no debería llevarse a cabo con una metodología tradicional centrada en la documentación o en los procesos tardados (Blanco, Camarero, Fumero, Warterski, & Rodríguez, 2009), (Abrahamsson P. , 2005). El desarrollo móvil debería buscar la obtención rápida de un producto funcional, para lo cual, los principios ágiles han prometido ser útiles. Aunque las metodologías ágiles (Beck, y otros, s.f.) son más antiguas que el desarrollo de software móvil con las plataformas actuales, sus principios fundamentales pueden ser directamente implementados a este tipo especial de software. En los últimos años han surgido metodologías ágiles como Mobile-D (Abrahamsson, Ihme, Kolehmainen, Kyllönen, & Salo, 2005), (Spataru, 2010), las cuales, combinan las principales prácticas ágiles que ya existen y las contextualizan para la obtención de aplicaciones móviles.



### 3. El trabajo realizado

#### 3.1 Descripción

Se realizó un desarrollo experimental de software con un grupo de ocho programadores. El objetivo era construir una aplicación didáctica para Android utilizando Java con el entorno Eclipse (Eclipse, 2013) y siguiendo la metodología ágil denominada Scrum.

La convocatoria para participar en este desarrollo se extendió a alumnos de licenciatura en informática y de ingeniería en sistemas computacionales cuyo nivel de programación fuera intermedio; se les solicitó que conocieran por lo menos un lenguaje de programación, preferentemente Java o C y que tuvieran las bases suficientes de diseño de aplicaciones para poder trabajar en el proyecto. Para seleccionarlos, se aplicó una encuesta y un examen de conocimientos de programación. A los solicitantes se les ofreció un diploma con valor curricular siempre y cuando no faltaran ni llegaran tarde a las sesiones. No hubo remuneración económica.

Al iniciar el trabajo, los participantes recibieron una capacitación de doce horas sobre el lenguaje Java para Android y los fundamentos de la metodología Scrum; el proceso fue muy interactivo y se contó con sesiones frecuentes de preguntas y respuestas para aclarar las dudas que surgían. Después de la capacitación, inició una segunda etapa de trabajo que duró ocho horas durante la cual se desarrolló el software solicitado siguiendo la metodología Scrum. Los programadores adoptaron el rol específico que debían cumplir y estuvieron conscientes de las tareas que debían realizar así como de los tiempos disponibles para el producto final. Una persona ajena al equipo tomó el rol de cliente y expuso sus necesidades; los programadores lo escucharon con atención. El cliente no proporcionó los requerimientos concretos de manera prescriptiva, sino simplemente expresó con una perspectiva de usuario final, unas necesidades previamente establecidas por el equipo de investigación (ver apartado 3.2). Los programadores tuvieron que organizarse de acuerdo a los principios de la metodología para extraer los requerimientos, tomar decisiones y asignar responsabilidades siempre tomando en cuenta los límites de tiempo establecidos.

#### 3.2 Requerimientos del software solicitado

El objetivo fue desarrollar una aplicación móvil en lenguaje Java para el sistema operativo Android que ayudara a ejercitar el razonamiento matemático de las personas que lo utilicen. La aplicación debería ofrecer tres niveles de dificultad: principiante, intermedio y avanzado; además, el usuario podría seleccionar entre realizar sumas, restas o multiplicaciones. Cada vez que el usuario acertara, la aplicación debería informar al usuario de su respuesta correcta mediante una imagen y un sonido distintivo; de la misma manera debería suceder cuando el usuario cometiera un error. Si el usuario dejaba alguna respuesta en blanco, el programa debería informarle que existen entradas vacías. La aplicación también tendría un contador para el número de intentos de solución. El usuario podría rendirse en cualquier momento y la aplicación le mostraría el resultado correcto; también se podría comenzar un juego nuevo en cualquier momento y por último la aplicación debería proporcionar al usuario información breve acerca del nombre del equipo que la programó y la fecha de desarrollo, así como una ayuda sobre la manera de utilizar el software.

Para la obtención del programa completo se realizaron dos iteraciones: en la primera se les



plantearon a los programadores los requerimientos previamente explicados con las siguientes excepciones: únicamente se tomarían en cuenta dos niveles: principiante y avanzado, no se les pidió incorporar ayuda para usar el programa y se les solicitó que el programa tuviera una sola pantalla para toda la aplicación. En la segunda iteración, se modificaron de manera intencional los requisitos planteados originalmente: ahora deberían utilizar algún tipo de menú y varias pantallas; también deberían incorporar la ayuda para el uso del programa y un nivel intermedio para la generación de los problemas aritméticos.

### 3.3 Cuestionario

Al finalizar el desarrollo experimental se aplicó un cuestionario a los integrantes del equipo (ver Tabla III) en el que se presentaron algunas afirmaciones acerca de la participación del equipo, de los problemas para organizarse, de la facilidad para la toma de decisiones, de la relación entre los integrantes y la comunicación entre ellos; también se cuestionaron aspectos sobre su nivel de motivación durante el tiempo de desarrollo y posterior a él para seguir aplicando la metodología en otros proyectos. Para estas afirmaciones se podían elegir una de las siguientes opciones: Completamente de acuerdo (4), De acuerdo (3), Neutral (2), En desacuerdo (1), Completamente en desacuerdo (0). También se les preguntó en un formato de respuesta libre cuál es el principal problema que tuvo el equipo durante el proyecto y cuál es la principal ventaja que percibieron en esta manera de trabajo. Así mismo se les pidió una evaluación cuantitativa de Scrum para la comunicación, motivación y organización del equipo durante el desarrollo experimental. Por su parte, los investigadores también reportaron los aspectos positivos y oportunidades de mejora que observaron durante este desarrollo experimental.

Tabla III.

En el desarrollo todos participamos activamente.
Tuvimos problemas para organizarnos
Se nos dificultó tomar decisiones
Todos los integrantes tuvimos una buena relación durante el desarrollo
Tuvimos problemas causados por la relación entre los integrantes del equipo
La comunicación entre los integrantes del proyecto fue deficiente
La metodología favoreció la participación de todos
La metodología utilizada contribuyó a la obtención rápida de un programa final de calidad
La metodología utilizada favorece sin duda el desarrollo rápido de aplicaciones móviles
Durante el desarrollo me mantuve motivado todo el tiempo
Creo que hicimos un proyecto de buena calidad
Aprendí cosas nuevas en el desarrollo de este proyecto
Estoy motivado a seguir aprendiendo más sobre esta metodología

Tabla III. Cuestionario.

## 4. Resultados obtenidos

### 4.1 Perspectiva de los participantes

Los datos estadísticos que se obtuvieron de este caso de estudio se muestran en la Tabla IV. Los integrantes del equipo de desarrollo destacan que tuvieron una buena relación entre ellos y que hubo un buen nivel de participación activa por parte de todos. A pesar de que reportaron haber tenido algunas dificultades para tomar decisiones en grupo y para organizarse conjuntamente, estos conflictos no llegaron a representar problemas graves en



la relación del equipo ni afectaron la comunicación entre los participantes.

El equipo coincidió por unanimidad en que Scrum favorece el desarrollo rápido de aplicaciones móviles. También de acuerdo a su percepción, la metodología contribuyó a la obtención rápida de un software de buena calidad y en un menor nivel, favoreció la participación de todos. A pesar de esto, algunos participantes se mantuvieron en una opinión neutral al preguntarles si aplicarían esta metodología a otros proyectos en el futuro.

Todos los participantes aceptaron haber aprendido cosas nuevas en este desarrollo al mismo tiempo que se mantuvieron motivados durante el proceso. También todos coincidieron en estar motivados a diferentes niveles para seguir aprendiendo más sobre la metodología Scrum y en percibir que el software resultante era de buena calidad.

En cuanto a la evaluación cuantitativa de Scrum, el aspecto con mejor calificado fue la motivación para trabajar, seguido de la comunicación entre los integrantes del equipo. Los aspectos evaluados con puntuaciones más bajas fueron la organización para trabajar y la eficiencia general de la metodología.

Las ventajas de Scrum que fueron identificadas por los participantes incluyen: participación conjunta en un mismo proyecto, integración del equipo con una manera ágil de trabajar, ayuda mutua, participación activa y eficiente en el área de fortaleza de cada persona, rapidez para terminar el proyecto, buen modelo de división de la carga de trabajo, buena organización y relación con el resto del equipo y desarrollo simultáneo de varias actividades.

Los principales problemas o desventajas que los participantes encontraron en este desarrollo de software en particular fueron: diferentes niveles de habilidades y conocimientos, poca paciencia de algunos integrantes, la manera de socializar entre el equipo, dificultades en la toma de decisiones, inconvenientes con el entorno de desarrollo y con algunos aspectos avanzados de la sintaxis del lenguaje utilizado.

Tabla IV.

Criterio	N	Evaluación Mínima	Evaluación Máxima	Media	Desviación
<b>Dinámica del equipo de trabajo</b>					
En el desarrollo todos participamos activamente.	8	2	4	3.38	.744
Tuvimos problemas para organizarnos	8	1	2	1.50	.535
Se nos dificultó tomar decisiones	8	1	3	1.63	.744
Todos los integrantes tuvimos una buena relación durante el desarrollo	8	3	4	3.88	.354
Tuvimos problemas causados por la relación entre los integrantes del equipo	8	0	2	.63	.916
La comunicación entre los integrantes del proyecto fue deficiente	8	0	2	.88	.835
<b>Percepción de la Metodología utilizada</b>					
La metodología favoreció la participación de todos	8	3	4	3.38	.518



Criterio	N	Evaluación Mínima	Evaluación Máxima	Media	Desviación
<b>Dinámica del equipo de trabajo</b>					
En el desarrollo todos participamos activamente.	8	2	4	3.38	.744
Tuvimos problemas para organizarnos	8	1	2	1.50	.535
Se nos dificultó tomar decisiones	8	1	3	1.63	.744
La metodología utilizada contribuyó a la obtención rápida de un programa final de calidad	8	3	4	3.63	.518
La metodología utilizada favorece sin duda el desarrollo rápido de aplicaciones móviles	8	4	4	4	0
<b>Percepción personal</b>					
Durante el desarrollo me mantuve motivado todo el tiempo	8	3	4	3.88	.354
Creo que hicimos un proyecto de buena calidad	8	3	4	3.63	.518
Aprendí cosas nuevas en el desarrollo de este proyecto	8	3	4	3.88	.354
Estoy motivado a seguir aprendiendo más sobre esta metodología	8	3	4	3.75	.463
En el futuro buscaré aplicar esta metodología a otros proyectos	8	2	4	3.63	.744
<b>Evaluación cuantitativa de Scrum</b>					
Comunicación entre los integrantes del equipo	8	2	4	3.50	.756
Motivación de los integrantes del equipo	8	2	4	3.63	.744
Organización para trabajar	8	2	4	3.25	.707
Eficiencia general de la metodología	8	0	4	3.25	1.389

Tabla IV. Resultados obtenidos: perspectiva de los participantes.

En entrevistas no estructuradas, los participantes reportaron haber aprendido nuevos aspectos del desarrollo de software de una manera amena e interactiva; manifestaron haberse sentido cómodos y estar contentos de haber podido conocer a nuevas personas con intereses afines. Algunos reconocieron que este ejercicio era su primer encuentro práctico con las metodologías ágiles o con el desarrollo móvil y estaban muy motivados a seguir aprendiendo sobre estos temas. La experiencia fue calificada como excelente por todos los entrevistados.

#### 4.2 Perspectiva de los investigadores

Siguiendo la metodología Scrum, se logró producir un software que cumplió con los requerimientos establecidos. La aplicación de la metodología fue positiva según la percepción de los investigadores. El grupo de desarrolladores cumplió con los tiempos establecidos para la entregas del software en cada iteración; se observó un buen ambiente de trabajo, en donde cada integrante del equipo aportó algo al software obtenido. El cliente aprobó el producto final, el cual cumplió con las pruebas especificadas. A petición de los conductores de la investigación, una persona completamente ajena al equipo que desconocía la aplicación que se había solicitado hizo una prueba de usabilidad al software



final; aunque este usuario no tuvo problemas en el manejo de la aplicación desarrollada, indicó que le pareció que la interfaz gráfica no era tan sencilla de utilizar y que pudo haber sido más intuitiva.

Los investigadores al observar al equipo y su organización para trabajar en este proyecto en concreto, destacaron los siguientes aspectos positivos: la integración del equipo, la rápida entrega del producto y la sinergia de habilidades diversas. Por otra parte, las principales desventajas que encontraron se refieren a la poca coordinación inicial que puede tener el equipo de desarrollo y al variado nivel de conocimientos y habilidades de los participantes.

### 4.3 Discusión

Los integrantes del equipo reportaron una experiencia positiva al aplicar Scrum al desarrollo de aplicaciones móviles; sin embargo, aceptaron haber tenido algunas dificultades relacionadas con la toma de decisiones, la organización para trabajar y la eficiencia general de la metodología. Esto puede estar relacionado con la naturaleza no prescriptiva de Scrum, en donde el equipo se debe auto organizar para alcanzar sus objetivos y con el hecho de que no todos los participantes se conocían entre sí. También se debe considerar que el tiempo de desarrollo fue corto (ocho horas) y que el equipo estaba formado por ocho personas. Llegar a acuerdos de equipo en estas condiciones no es tarea fácil. El equipo se demoró en tomar decisiones y adoptó una forma de trabajo centralizada en unas cuantas personas responsables, lo cual promovió algunos tiempos libres en algunos integrantes y dio la percepción de que la eficiencia de la metodología podría mejorarse. Quizá por esta razón algunos integrantes mostraron una postura neutral al considerar aplicar Scrum en otros proyectos futuros. Por otra parte, algunos participantes reportaron haber tenido inconvenientes con el entorno de desarrollo y algunos aspectos del lenguaje, ya que el conocimiento del lenguaje y de las herramientas utilizadas por parte del equipo se ubicó en un nivel básico – intermedio.

Desde la perspectiva del proceso enseñanza - aprendizaje, la experiencia fue positiva sin duda. Los alumnos pusieron en práctica conocimientos específicos del área de desarrollo de software relacionados con Android, Java y Scrum, desarrollaron habilidades como el trabajo en equipo y la administración del tiempo y valores como la responsabilidad y la honestidad. El proceso de trabajo conjunto permitió la retroalimentación continua de experiencias que enriqueció a cada uno de los integrantes e hizo crecer al equipo completo.

### 5. Conclusiones y trabajo futuro

En este artículo se presentó un caso de estudio en el que se desarrolló un software con el objetivo de que los programadores involucrados, todos ellos estudiantes de carreras profesionales de sistemas e informática, pusieran en práctica las herramientas necesarias para el desarrollo básico de aplicaciones móviles utilizando Java para Android y Scrum.

Los resultados obtenidos muestran que Scrum es una metodología que puede favorecer el desarrollo rápido de aplicaciones de buena calidad y que los casos prácticos de estudio son herramientas útiles en el proceso de enseñanza aprendizaje. Este ejercicio experimental en concreto produjo a tiempo un software que cumplió con los requerimientos funcionales especificados y contó con un equipo de personas motivadas que construyeron nuevos



conocimientos y los pusieron en práctica responsablemente.

Como trabajo futuro se plantea la replicación de este desarrollo experimental, primero con estudiantes y después con desarrolladores de software profesionales. Además de Scrum se propone utilizar otra metodología ágil y una metodología tradicional con el propósito de contrastar los resultados obtenidos.

## Referencias

- Abrahamsson, P. (2005). *Mobile software development - the business opportunity of today*. Iceland.
- Abrahamsson, P., Ihme, T., Kolehmainen, K., Kyllönen, P., & Salo, O. (2005). *Mobile-D for Mobile Software: How to Use Agile Approaches for the Efficient Development of Mobile Applications*. Finland: VTT Technical Research Centre of Finland.
- Basic 4 Android*. (2013). (Anywhere Software) Retrieved 03 01, 2013 from <http://www.basic4ppc.com/>
- Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (n.d.). *Portal web del manifiesto ágil*. Retrieved 03 01, 2013 from [agilemanifesto.org](http://agilemanifesto.org)
- Blanco, P., Camarero, J., Fumero, A., Werterski, A., & Rodríguez, P. (2009). *Metodología de desarrollo ágil para sistemas móviles. Introducción al desarrollo con Android y el iPhone*. Madrid: Universidad Politécnica de Madrid.
- Eclipse. (2013). *Eclipse*. Retrieved 05 20, 2012 from <http://www.eclipse.org>
- Gironés, J. (2010). *El gran libro de Android*. España: Marcombo.
- Kitchenham, B., Pickard, L., & Lawrence Pfleeger, S. (1995). Case Studies for Method and Tool Evaluation. *IEEE Software*, 52-62.
- Pressman, R. (2010). *Ingeniería del Software: un enfoque práctico*. México, D.F.: McGrawHill.
- Sims, C., & Johnson, H. (2011). *Elements of Scrum*. Foster City, California: Dymaxicom.
- Sims, C., & Johnson, H. (2012). *SCRUM: A breathtakingly brief and agile introduction*.
- Spataru, A. C. (2010). *Agile Development Methods for Mobile Applications*. University of Edinburgh.
- Varma, V., & Garg, K. (2005). Case Studies: The Potential Teaching Instruments for Software Engineering Education. *Proceedings of the Fifth International Conference on Quality Software (QSIC'05)*.
- Xamarin, S. (2013). *Mono for Android*. Retrieved 03 01, 2013 from <http://xamarin.com/monoforandroid>

