

Reporte de caso

Desarrollo Ágil de una Aplicación para Dispositivos Móviles. Caso de Estudio: Taxímetro Móvil

Agile Application Development for Mobile Devices. Case Study: Mobile Taximeter

Angélica Babativa¹ , Paula Briceño¹ , Consuelo Nieto¹ , Omar Salazar¹

¹Universidad Distrital Francisco José de Caldas. Correspondence email: acnietol@udistrital.edu.co

Recibido: 29-03-2016. Modificado: 13-06-2016. Aceptado: 18-07-2016

Resumen

Contexto: La globalización ha afectado todos los sectores productivos y, en particular, la industria del software, lo cual ha requerido el desarrollo de nuevas metodologías que se adapten a las velocidades de los cambios y permitan construir rápidamente productos que satisfagan los requerimientos de los clientes. Por otro lado, la tecnología GPS, la conectividad 4G y la integración de redes sociales con las que cuentan la mayoría de teléfonos móviles actuales han abierto un campo grande de aplicación, en particular en el área del transporte, la movilidad y la denuncia ciudadana, cuyo desarrollo puede ser abordado con éxito mediante una metodología ágil.

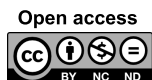
Método: Se utilizó la metodología ágil Scrum para el desarrollo de una aplicación móvil sobre el sistema operativo Android y tecnología GPS, que le permite a un usuario de taxi en Bogotá monitorear el recorrido y enviar una denuncia a la red social Twitter en caso de inconformidad. Para el análisis y diseño se utilizaron algunos modelos de UML y se aplicó un intervalo de confianza para validar los resultados.

Resultados: Prototipo de un taxímetro móvil desarrollado con una metodología ágil, que satisface características de calidad, extensibilidad y mantenibilidad. Se utilizó una distribución t-student para validar la medición del prototipo sobre 50 muestras, concluyendo que la diferencia entre la medición de un taxímetro real y la del taxímetro móvil es en promedio dos unidades con una desviación estándar de 1,39 unidades.

Conclusiones: Se muestra que con el desarrollo ágil se pueden combinar herramientas de modelado UML y técnicas de validación estadísticas para obtener productos de calidad, que no violan, sino que, por el contrario, reafirman los principios del desarrollo ágil.

Palabras clave: Android, dispositivos móviles, redes sociales, Scrum, taxímetro GPS.

Idioma: Español



©The authors; licensee: Revista INGENIERÍA. ISSN 0121-750X, E-ISSN 2344-8393

Cite this paper as: Babativa, A., Briceño, P., Nieto, C., Salazar, O.: Desarrollo Ágil de una Aplicación para Dispositivos Móviles. Caso de Estudio: Taxímetro Móvil. INGENIERÍA, Vol. 21, Num. 3, 2016 pp. 260:275. En línea DOI: <http://dx.doi.org/10.14483/udistrital.jour.reveng.2016.3.a01>

Abstract

Context: Globalization has affected all productive sectors and in particular the software industry, which has required the development of new methodologies to suit the speed of the changes and allow quickly build products that meet the requirements of the customers. On the other hand, the GPS technology, 4G connectivity and integration of social networks that have the most current mobile phones have opened a large field of application, particularly in the area of the transport, mobility and citizen complaint, whose development can be successfully addressed through an agile methodology.

Method: Agile methodology Scrum was used for the development of a mobile application on the Android operating system and GPS technology, which allows a Bogota taxi user to monitor the route and send a complaint to the social network Twitter in case of nonconformity. Some UML models were used for analysis and design of the application, and a confidence interval was used to validate the results.

Results: Prototype of a mobile taximeter developed with an agile methodology that meets quality characteristics, extensibility and maintainability. T-student distribution was used to validate the measurement of the prototype on 50 samples, concluding that the difference between the measurement of a real taximeter and our mobile taximeter is on average 2 units with a standard deviation of 1,39 units.

Conclusions: It is shown that with the agile development can be combined with UML modeling tools and statistical validation techniques for quality products that do not violate, but on the contrary, that reaffirm the agile development principles.

Keywords: Android, GPS taximeter, mobile devices, Scrum, social network.

1. Introducción

Hoy en día hay en el mundo más dispositivos móviles que computadores [1], [2] y muchas personas los utilizan no solo como medio de comunicación, sino también para acceder a servicios comerciales, financieros, de entretenimiento, de transporte, de denuncia ciudadana, entre otros. Es por esto que el desarrollo de aplicaciones móviles se ha expandido en los últimos años debido, por un lado, a la masificación de los teléfonos móviles y al acceso a servicios de datos, y por otro, a la aparición de herramientas de software que facilitan su desarrollo.

Además de tener un amplio campo de aplicación, las aplicaciones móviles tienen unas características que las diferencian de una aplicación web o de escritorio [3], [4]: uso de menos interfaces físicas y más interfaces gráficas para la interacción, uso limitado de memoria, almacenamiento y procesamiento; desarrollo para múltiples plataformas como *Android*, *BlackBerry*, *iPhone/iPad*; procesos de despliegue y de mantenimiento flexibles.

Las múltiples características que incluyen los dispositivos móviles (pantalla táctiles, cámara, GPS, conectividad Bluetooth, Wifi, 3G, 4G) han motivado una gran demanda de aplicaciones en una variedad de escenarios, en particular en el campo del tráfico, la movilidad y el servicio de taxis [5]–[10], pero la demanda de nuevas aplicaciones y la versatilidad de algunas herramientas de software ha llevado a que el desarrollo se haga sin un marco metodológico formal que asegure su calidad y mantenimiento.

En la literatura se encuentran varias propuestas para el desarrollo de aplicaciones móviles, que van desde algunas modificaciones a las metodologías clásicas hasta la adopción de metodologías ágiles. En [3] se presenta el marco de trabajo denominado Mobile Application Development Life Cycle (MADLC, por sus siglas en inglés) que considera las etapas de identificación, diseño, implementación, prototipado, pruebas, despliegue y mantenimiento, pero en el que no se hace énfasis en el tratamiento ágil del desarrollo. En [8] se presenta una propuesta para la construcción rápida de aplicaciones móviles mediante un generador de código que facilita la implementación de tareas repetitivas (ejecución de ciertos comandos con el teclado, manejo de menús, conexión a una base de datos, envío de SMS), pero no se abordan los aspectos metodológicos. En [4] se propone un *framework* genérico para desarrollar aplicaciones móviles sobre diferentes sistemas operativos, pero tampoco discute el tema metodológico.

En este artículo se presenta el desarrollo de una aplicación móvil sobre el sistema operativo *Android* que le permitirá a un usuario del servicio público de taxi en la ciudad de Bogotá D.C., monitorear el recorrido mediante la tecnología *GPS*, determinar la tarifa del servicio y enviar una denuncia a la red social Twitter en caso de inconformidad. Se ilustra el uso detallado de la metodología ágil *Scrum*, el empleo de algunos modelos de *UML* para el análisis y diseño, y se muestra la aplicación de técnicas estadísticas para validar el requerimiento no funcional de confiabilidad en casos en que el software puede entregar resultados diferentes, dadas las mismas entradas debido a variables del entorno (calidad de señal de GPS debido al clima, edificios altos y otros obstáculos, u otras razones externas a la aplicación) que afectan el comportamiento del sistema.

1.1. Descripción del caso de estudio

El servicio público de taxis ha adquirido mucha importancia en los últimos años, dado los beneficios de movilidad que ofrece a la ciudadanía. Un 42.3 % [11] de la población bogotana prefiere este servicio para desplazarse a su trabajo, sitio de estudio o lugar de vivienda, dada la comodidad, rapidez y servicio puerta a puerta que ofrece. Sin embargo, a medida que ha aumentado la demanda de este servicio también se ha incrementado el número de usuarios inconformes con el cobro de la tarifa por parte de los taxistas [12]. Lo anterior ha generado altercados, desconfianza y zozobra en los usuarios, poniéndose en riesgo la demanda de este tipo de transporte.

Una de las opciones que ha surgido para el cálculo de la tarifa en taxis es el taxímetro con GPS [13]. Aunque dicha tecnología presenta ciertas limitaciones para la medición precisa debido a que la señal se debilita ante la presencia de edificios muy altos o zonas de escasa cobertura, los simuladores de taxímetros con GPS permiten calcular el valor de la tarifa aproximándose bastante a la tarifa total a pagar [14]. Para calcular el valor del recorrido, el GPS indica la velocidad y el tiempo transcurrido de una carrera, de esta manera el taxímetro no va conectado a ningún sensor de velocidad ni de movimiento y el pasajero viaja más tranquilo, pues la probabilidad de que el taxímetro esté adulterado es mínima. Algunas aplicaciones móviles como *Taxímetro GPS* [9] y *Taxiando* [15] utilizan la tecnología GPS y si bien simulan el funcionamiento de un taxímetro, presentan algunos inconvenientes como demora al iniciar la aplicación, las unidades corren más rápido que el taxímetro real, manejan mucha publicidad, no permiten guardar datos de los recorridos ni manipularlos para generar reportes, entre otros.

El objetivo de este trabajo fue implementar una aplicación móvil utilizando la tecnología GPS que viene incorporada en los dispositivos móviles con sistema operativo *Android*. No solo se propuso resolver las limitaciones de las aplicaciones anteriormente mencionadas, sino que además se consideraron las tarifas específicas definidas anualmente por la Secretaría Distrital de Movilidad para la ciudad de Bogotá D.C., [16] y se extendió la funcionalidad para enviar un mensaje a la red social *Twitter* con la placa del taxi, la empresa a la que está vinculado, el número de unidades del recorrido, el valor calculado por la aplicación y el cobrado por el taxista, como una forma de denuncia social.

1.2. Cálculo del servicio de taxi

Para el cálculo del cobro del servicio de taxi, se hace uso del taxímetro como instrumento de medición, el cual le indica al pasajero la cantidad total que debe pagar según las unidades marcadas basándose en la distancia recorrida y el tiempo transcurrido. Para calcular el valor a pagar, cada una de las unidades que marca tiene un equivalente en distancia y en tiempo: se marca una unidad por cada 100 metros recorridos o por cada 30 segundos de espera [17]. La cantidad fija es un valor económico acreditado legalmente por la Secretaría Distrital de Movilidad [16] para el pago de los servicios del transporte retribuido a los taxistas, la cual se compone de:

1.2.1. Costo inicial

También llamado *banderazo*; se refiere al valor en el que inicia el taxímetro al momento de ser puesto en servicio. De acuerdo a la Alcaldía Mayor de Bogotá [16], se estipuló que para el año 2014-2015, el *banderazo* sería de 25 unidades, que corresponden a \$2000 pesos.

1.2.2. Costo del incremento

Se refiere al valor económico habitual y constante en el que va aumentando de las unidades del taxímetro a partir del costo inicial.

1.2.2.1 Costo por función tiempo

Es un valor monetario que se calcula a partir de la siguiente fórmula [18]:

$$c.p.f.t = \frac{[(segundos) * (costo por hora de servicio)]}{3600}$$

En donde *segundos* corresponde a la cantidad de tiempo en la que el taxi ha registrado una velocidad menor o igual a la velocidad de cambio de arrastre, y *costo por hora de servicio* es el valor fijado por la Alcaldía Mayor de Bogotá [16]. Se estipuló que para el año 2014-2015, el costo por hora sería de 225 unidades, que corresponden a \$17600 pesos. El costo por la función tiempo es sumado al acumulador interno de costo a medida que se vaya generando.

1.2.2.2 Costo por función de distancia

Es un valor monetario que se calcula a partir de la siguiente fórmula [18]:

$$c.p.f.d = \frac{[(metros\ recorridos) * (costo\ Kilometro)]}{1000}$$

En donde *metros recorridos* corresponde a la distancia recorrida durante el servicio, y *costo por kilómetro* es el resultado de la suma de los costos fijos, variables y de capital calculados por la autoridad competente. De acuerdo a la Alcaldía Mayor de Bogotá [16], se estipuló que para el año 2014-2015, el costo por kilómetro corresponde a \$780 pesos. El costo por la función distancia es sumado al acumulador interno de costo a medida que se vaya generando.

2. Metodología

El desarrollo de aplicaciones móviles debe, por un lado, considerar las características particulares de los dispositivos como las restricciones de memoria, procesamiento y almacenamiento, implementación de plataforma cruzada, procesos de despliegue y mantenimiento [3], [4], y por otro, debe responder rápidamente a la creciente demanda de soluciones móviles. Si bien las metodologías ágiles han sido empleadas tradicionalmente en aplicaciones web y de escritorio, las características anteriores las hacen apropiadas en el campo del software móvil. Si bien las metodologías ágiles han sido empleadas tradicionalmente en aplicaciones web y de escritorio, las características anteriores también las hacen apropiadas en el campo del software móvil, como se muestra en [19] con el uso de la metodología de Proceso Unificado Ágil (AUP) para el desarrollo, tanto de una aplicación de escritorio, como para una aplicación móvil. Este documento presenta un caso práctico del uso de la metodología ágil *Scrum* [20] en el desarrollo de una aplicación móvil para el control de la tarifa del servicio público de taxi en la ciudad de Bogotá D.C., en el que se muestra que para alcanzar altos estándares de calidad y mantenibilidad se pueden integrar algunos artefactos de UML en el análisis y diseño, y utilizar técnicas estadísticas para la validación de los resultados, sin violar los principios del manifiesto ágil [21].

2.1. Planeación del proyecto–ProductBacklog

Para la planeación, seguimiento y gestión del proyecto se hizo uso de *IceScrum* [22], aplicación web de código abierto que soporta la metodología *Scrum*. Inicialmente se construyó el *Product-Backlog*, documento en el que se agruparon las historias de usuario funcionales y no funcionales; se establecieron los *Sprints* y se asociaron las historias a cada *Sprint*. Para manejar el cambio, el proceso de desarrollo fue iterativo e incremental, pues a medida que se iba avanzando en el desarrollo se encontraron algunas historias de usuarios que eran redundantes, otras en las que se debía especificar con mayor detalle la lógica de negocio y otras que debían desarrollarse a lo largo de varios *Sprints*.

Se definieron cuatro tipos de historias de usuario para su adecuada visualización y seguimiento en *IceScrum* (Figura 1): Gestión de tarifas, Medición, Notificación y Validación. Se establecieron

Sprint 1 From 17/04/2015 to 01/05/2015	Sprint 2 From 02/05/2015 to 16/05/2015	Sprint 3 From 17/05/2015 to 31/05/2015	Sprint 4 From 01/06/2015 to 15/06/2015	Sprint 5 From 16/06/2015 to 30/06/2015	Sprint 6 From 01/07/2015 to 15/07/2015
1. Construcción de las historias de usuario	5. Diagrama de componentes	9. Cargar modelo de tarifa	13. Contabilizar recargos	16. Almacenar información del servicio	18. Notificar y actualizar modelo de tarifa
2. Prototipo de las historias de usuario	6. Modelo de base de datos	10. Iniciar contadores	14. Seleccionar recargos	17. Generar informe con los datos del servicio	19. Notificar queja en Twitter
3. Modelo de Requerimientos	7. Verificar la solicitud de activación del GPS	11. Iniciar medición	15. Calcular valor de la carrera		
4. Diagrama de clases	8. Mostrar la ubicación del usuario en el mapa	12. Actualizar mapa			

Figura 1. Sprints e historias de usuario por tipo

siete *Sprints*, cada uno con más o menos un mes de duración, incluyendo el *Sprint 0* en el que se adelantaron tareas de investigación y prototipado acerca del uso de las herramientas tecnológicas.

2.2. Documentación de las historias de usuarios

Debido a que *IceScrum* solo permite incluir cierta información para cada historia de usuario (nombre, tipo y descripción), como parte de este proyecto se elaboró una plantilla para registrar información adicional (Tabla I) que resultó muy útil, tanto para la implementación como para las pruebas.

Tabla I. Plantilla para documentar historias de usuario

Historia de usuario	
Código: Hu_Not_01	Nombre: Notificar queja en Twitter
Tipo HU: Funcional	Complejidad: A
Actor: Usuario	Hu.Relacionadas: Hu_Med_03, Hu_Med_04, Hu_Med_05, Hu_Med_06.
Modulo: Notificación	
Descripción: Si el usuario quiere reportar una situación anómala, puede difundirlo por medio de la red social Twitter	
Criterios de aceptación	
Condición:	Resultado:
Al finalizar el recorrido el usuario da click en la opción "Reportar"	Publica en la red social Twitter un reporte con los siguientes datos: Placas del taxi, empresa, unidades de diferencia, motivo del reporte y comentarios

2.2.1. Dependencia entre las historias de usuario

La matriz de trazabilidad [23], [24] es una técnica bastante útil en el proceso de desarrollo de software, permite gestionar de manera adecuada los cambios que se pueden presentar a lo largo de un proyecto. Es decir, al momento de querer hacer una modificación, la gestión de cambios debe tener conocimiento previo de los requerimientos que pueden verse afectados directa e indirecta-

Tabla II. Matriz de trazabilidad de historias de usuario

		HISTORIAS DE USUARIO (A) DEPENDIENTES									
HU vs. Hu		Hu_Med.01	Hu_Med.02	Hu_Med.03	Hu_Med.04	Hu_Med.05	Hu_Med.06	Hu_Med.07	Hu_Med.08	Hu_Tar.01	Hu_Not.01
INDEPENDIENTES	Hu_Med.01		x	x	x						
	Hu_Med.02			x	x						
	Hu_Med.03				x	x	x	x	x		x
	Hu_Med.04					x	x	x	x		x
	Hu_Med.05						x	x	x		x
	Hu_Med.06							x	x		x
	Hu_Med.07								x		
	Hu_Med.08										
	Hu_Tar.01										x
	Hu_Tar.02										
	Hu_Not.01										

mente. De esta manera se evita que las historias de usuario queden inconsistentes y ambiguas a la hora de hacer un cambio.

En la Tabla II se muestra la matriz de trazabilidad de las historias de usuario asociadas al proyecto: en ésta matriz las filas representan las historias de usuario independientes y las columnas las historias que dependen de las primeras. Así, se puede observar que las historias de usuario HU_Med_003, HU_Med_004 y HU_Med_005 son las que más generan dependencias.

2.3. Definición y seguimiento de los Sprints del proyecto

Como se mencionó anteriormente, se definieron siete *Sprints* (Tabla III) con cerca de un mes de duración cada uno, con los que se cubrió todo el ciclo de vida del desarrollo de la aplicación, incluidas tareas de planificación, prototipado, análisis, diseño, implementación, validación y despliegue. Al inicio de cada *Sprint* se hizo la reunión de planeación (*Sprint Planning Meeting*) tomando como base la planeación inicial registrada en el *ProductBacklog* y al final del cada uno de ellos se hizo la reunión de revisión y retrospectiva (*Sprint Review Meeting* y *Sprint Retrospective*) con el fin de validar el resultado de las tareas propuestas y los ajustes que se debían hacer en el plan de los siguientes *Sprints*.

Tabla III. Sprints del proyecto con HU asociadas

<i>Sprint</i>	Historias Planificadas
<i>Sprint 0</i>	Documentación previa del desarrollo móvil.
	Desarrollo de un prototipo inicial de bajo nivel.
<i>Sprint 1</i>	Construcción de las historias de usuario.
	Prototipo de las historias de usuario.
	Modelo de requerimientos.
	Modelo de casos de uso.
<i>Sprint 2</i>	Arquitectura de la aplicación.
	Diagrama de clases.
	Modelo de base de datos.
	Verificar y solicitar la activación del GPS.
	Mostrar la ubicación del usuario en el mapa.
<i>Sprint 3</i>	Iniciar aplicación.
	Cargar modelo de tarifa.
<i>Sprint 4</i>	Iniciar medición.
	Contabilizar y seleccionar recargos.
<i>Sprint 5</i>	Calcular valor de la carrera.
	Almacenar información del servicio.
	Generar informe con los datos del servicio.
<i>Sprint 6</i>	Notificar y actualizar modelo de la tarifa.
	Notificar queja en <i>Twitter</i> .
	Manual del usuario de la aplicación móvil: <i>TwTaxi</i> .
	Manual del usuario de la aplicación web: Administrador <i>TwTaxi</i> .

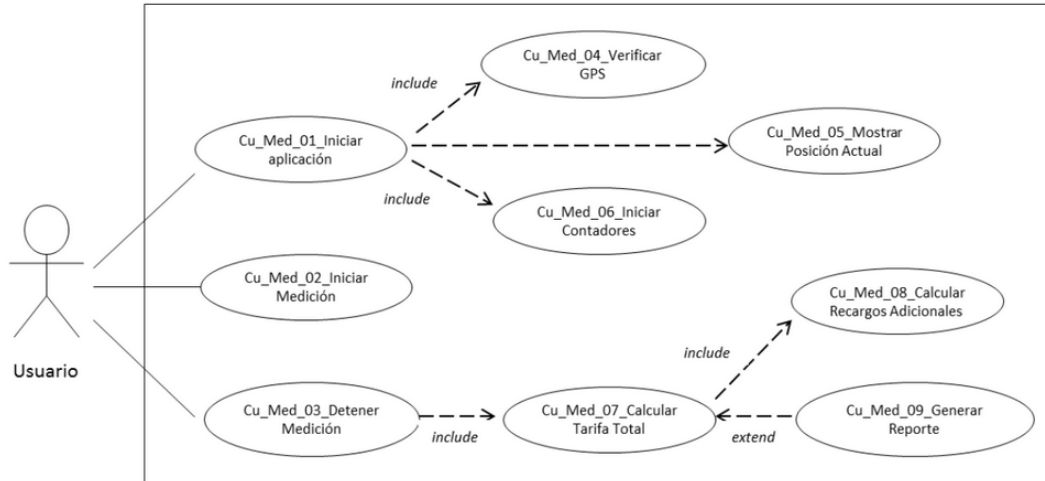


Figura 2. Diagrama de casos de uso para el módulo de medición

2.4. UML para el modelado ágil

Si bien las metodologías ágiles se centran en la comunicación entre el cliente y el equipo de desarrollo como criterio de éxito de un proyecto [21] y no enfatizan en la documentación, ello no implica que el uso de herramientas formales de modelado estén excluidas; por el contrario, pueden asegurar la calidad y mantenibilidad de una aplicación al no sacrificar las buenas prácticas del desarrollo de software.

En este trabajo se utilizaron diagramas de actividades para identificar las tareas del taxímetro y el flujo de datos entre ellas, los diagramas de casos de uso combinados con la descripción de las historias de usuario como una forma ágil de documentación (Figura 2), diagramas de clases y paquetes para agrupar la funcionalidad y diagramas de secuencia para modelar el flujo de eventos desde que se inicia el taxímetro hasta que se envía la notificación a la red social *Twitter*. Aquí solo se presenta el diagrama de paquetes de la aplicación (Figura 3); los diagramas completos se pueden consultar en [25].

3. Resultados

Con el desarrollo de este proyecto se implementó una aplicación móvil para controlar el cobro de la tarifa de taxi en la ciudad de Bogotá D.C., conforme a la normatividad vigente y haciendo uso de la red social *Twitter* como medio de denuncia, pero también se documentó la experiencia práctica del uso de una metodología ágil para el desarrollo de aplicaciones móviles y el empleo de la estadística como forma de validación de los resultados.

3.1. Taxímetro móvil – TwTaxi

Se implementó un taxímetro móvil sobre el sistema operativo *Android* (Figura 4) con las siguientes funcionalidades: a) Cargar las tarifas al iniciar el taxímetro y actualizar periódicamente las tarifas vigentes para la ciudad de Bogotá D.C. b) Calcular la localización geográfica del móvil mediante coordenadas espaciales y medir el recorrido con respecto al tiempo y la distancia, c) Calcular

el costo total de la carrera con la opción de ingresar cargos adicionales; d) Notificar cada vez que hay un nuevo modelo de tarifa, e) Permitir el envío del reporte del recorrido a *Twitter*.

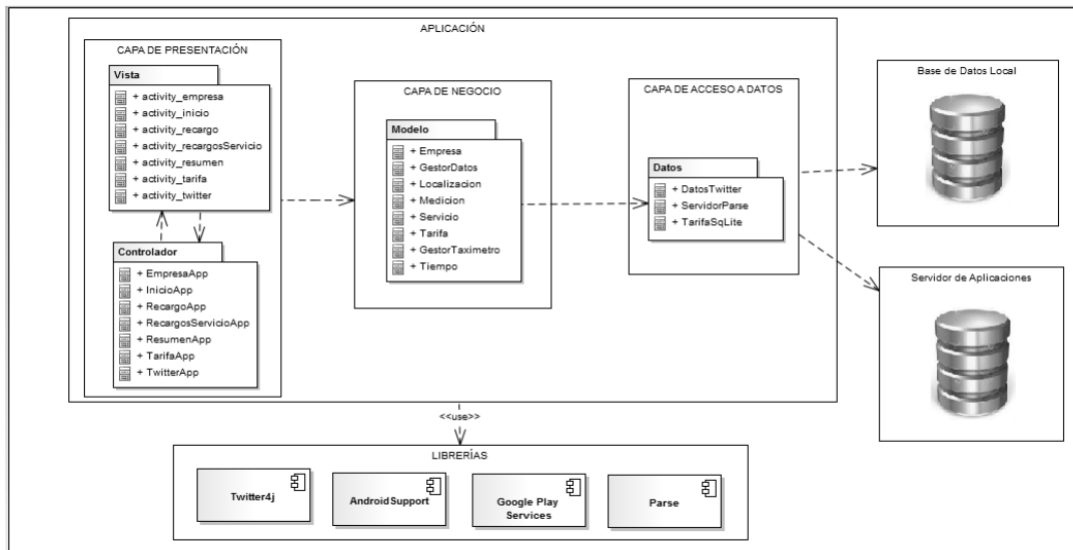


Figura 3. Diagrama de paquetes del taxímetro



Figura 4. Interfaz del taxímetro móvil desarrollado

La aplicación móvil se construyó utilizando el patrón MVC [26] sobre una arquitectura en tres capas (presentación, lógica y persistencia) [27]. Se utilizaron las siguientes tecnologías: a) Internet inalámbrico b) *Android* versión 4.4 *KitKat* y *Java Standard Edition* versión 8 c) *Justinmind* versión 6.4 d) *AndroidAsset Studio* e) *Google Maps* f) *Eclipse Luna Service Release 2* versión 4.4.2 g) *BizagiModeler* h) *Enterprise Architect* i) *Twitter4j* j) *AndroidSupport v7 appcompat* k) *BackEndParse* versión 1.7.1 l) *SQLite*. Se escogió *Android* versión 4.4 *Kit kat* como sistema operativo

Tabla IV. Herramientas de software utilizadas en la implementación del taxímetro

CAPA	Característica	Herramienta Utilizada
Presentación	Permitió la comunicación de la aplicación móvil con otras tecnologías.	Internet inalámbrico.
	Lenguaje de Programación.	<i>Android</i> versión 4.4 <i>KitKat</i> y <i>Java Standard Edition</i> versión 8.
	Herramienta para crear prototipos de interfaz.	<i>Justinmind</i> versión 6.4.
	Herramienta que permite crear componentes gráficos.	<i>AndroidAsset Studio</i> .
Negocio	Librería utilizada para manipular y utilizar los mapas de Google.	<i>Google Maps</i> .
	Entorno de desarrollo y compilación.	<i>Eclipse Luna Service Release 2</i> versión 4.4.2.
	Herramienta utilizada para el análisis y diseño de procesos.	<i>BizagiModeler</i> .
	Herramienta utilizada para el análisis y diseño de la aplicación.	<i>Enterprise Architect</i> .
	Librería que permite la interacción de una aplicación móvil con <i>Twitter</i> .	<i>Twitter4j</i> .
Persistencia	Librería utilizada para administrar la compatibilidad de las diferentes versiones de <i>Android</i> .	<i>AndroidSupport v7 appcompat</i> .
	Persistencia de Datos en la nube	<i>BackEndParse</i> versión 1.7.1.
	Persistencia de Datos local.	<i>SQLite</i> .

y *Eclipse Luna Service Release 2* versión 4.4.2 como IDE. Para el desarrollo y pruebas de la aplicación se utilizó: un portátil *DELL* procesador *intelcore i5*, una *Tablet Samsung GalaxyTab4, Android 4.4.2* y un celular *Huawei Y511 Android 4.4*. En la *Tabla IV* se resume el uso de las herramientas tecnológicas por capas.

3.2. Validación

Una de las razones que justificó la utilización de la metodología ágil para el desarrollo del proyecto fue la restricción de tiempo. Sin embargo, ágil no implica dejar de lado el uso disciplinado de una metodología ni sacrificar los estándares de calidad en el proceso. Tal como lo afirma Humphrey [28], para obtener un producto de calidad se debe seguir un proceso de calidad.

La estrategia adoptada para asegurar la calidad de software durante el desarrollo del proyecto fue una combinación de: a) revisión e inspección de los artefactos desarrollados en cada Sprint, pues es la mejor estrategia para identificar y remover defectos antes de las pruebas; b) pruebas unitarias y pruebas del sistema de acuerdo con los lineamientos de Sommerville [27] para validar los requerimientos definidos en las Historias de Usuario y c) validación estadística aplicando Montgomery, para certificar la confiabilidad de la medición del taxímetro.

En este artículo se presenta el resultado de la validación estadística por considerarla de especial interés en un proceso de pruebas de software, en el que dadas las mismas entradas (hora del servicio, distancia recorrida, tarifa) es posible encontrar diferentes resultados, debido por un lado, a la tecnología de GPS que incorporan los dispositivos móviles y que se ve afectada por condiciones espaciales, de comunicación y de interferencia, y por otro, a condiciones de movilidad en diferentes horas del día. El proceso de validación estadística permitió establecer la métrica de confiabilidad del taxímetro para una combinación dada de diferentes escenarios.

3.2.1. Metodología

Para validar la confiabilidad de la medición del taxímetro se siguieron las pautas que propone Montgomery [29] las cuales proporcionan una adecuada planificación en el diseño de experimentos.

Las etapas que se mencionan a continuación se ejecutaron en forma secuencial:

- **Identificación y exposición del problema:** El GPS, como instrumento de medida, en algunas ocasiones suministra datos imprecisos e inexactos; en la mayoría de los casos, estas situaciones se presentan por interrupciones en el envío y recepción de señales entre el GPS del móvil y el satélite encargado de recibir las coordenadas del dispositivo móvil.
- **Elección de los factores :** Los factores que alteraron e influyeron en el desempeño del sistema, fueron: a) Factor constante: Para realizar las medidas se tomó de referencia siempre el mismo taxímetro como instrumento de referencia y como instrumento de medida siempre se tomó el mismo teléfono celular. b) Factor variable: Los días y zonas fueron los factores que se permitieron variar durante el experimento. En el mes de julio se tomaron los datos los días comprendidos entre el 13 y el 24 de julio, en el mes de agosto se tomaron los datos entre el 11 y 19 de agosto.
- **Elección de la técnica estadística :** Distribución *t-student*: esta distribución representada por una campana, modela diversos fenómenos que suceden en la naturaleza, en la investigación y en la industria. Una de las aplicaciones que más se asemejan al comportamiento de esta distribución, es el tratamiento de errores en la toma de medidas, es por ello que se escogió esta técnica para validar el prototipo. Además, el tamaño de muestra, junto con el uso de la media y varianza muestral, hacen que se pueda determinar un intervalo de confianza para la aplicación.

3.2.2. Materiales, desarrollo y resultado del experimento

Los datos se tomaron con un celular Huawei Y511 Android 4.4 con 4GB de Internet. Para tomar los datos se siguieron los siguientes pasos:

- Se verificó que el dispositivo móvil tuviera conexión a Internet.
- Se abrió la aplicación segundos antes de subir al taxi, para que el sistema ubicara el dispositivo y cargara el mapa.
- Luego de estar en el taxi, se iniciaron al mismo tiempo: el instrumento de referencia (taxímetro) y el instrumento experimental (aplicación).
- Al llegar al lugar de destino se detuvieron al mismo tiempo: el taxímetro y la aplicación.
- Una vez se finalizó el recorrido se registraron las unidades que marcaba la aplicación y las que marcaba el taxímetro.
- Luego de realizar 50 muestras se encontró que las unidades registradas por la aplicación versus las registradas por el taxímetro diferían en -2 unidades promedio con una desviación estándar de 1,39 unidades (Figura 5 en el apéndice). Un tamaño de muestra igual a 50 se considera aceptable, desde el punto de vista económico y logístico [29].

Intervalo de confianza

El intervalo de confianza N de un instrumento ésta determinado por: la media muestral (x), la desviación estándar (s), el tamaño de la muestra (n) y por una constante de cobertura (t) llamada *T-Student*. Esta constante se saca a partir de: el nivel de confianza que se quiere, por lo general del 95 % y el número de grados de libertad.

$$N = x' \pm t \times \frac{s}{\sqrt{n}}$$

Dado que el tamaño de la muestra fue 50, los grados de libertad son 49 y el nivel de confianza el 95 %. Se obtuvo de una tabla de la distribución *T-student* una constante de cobertura igual a: 2,096. Remplazando la constante T- Student, la media muestral, la desviación estándar y el tamaño de la muestra en la ecuación anterior se tiene un intervalo de confianza entre -2,39 unidades y -1,61 unidades. Es decir, para una confianza del 95 %, se obtuvo que la aplicación registró entre 1,61 a 2,39 unidades menos que el taxímetro real.

4. Conclusiones

El uso masivo de dispositivos móviles y de teléfonos “inteligentes” ha generado una gran demanda de aplicaciones de software en diversos campos de aplicación. El desarrollo de aplicaciones móviles debe considerar ciertas características de hardware y software como el uso de menos interfaces físicas y más interfaces gráficas para la interacción, uso limitado de memoria, almacenamiento y procesamiento; desarrollo para múltiples plataformas como Android, BlackBerry, iPhone/iPad; y procesos de despliegue y mantenimiento flexibles. Hoy en día existen varias herramientas que consideran estas características y que además permiten integrar capacidades de software, hardware y conectividad para dar respuesta rápida a la gran demanda de aplicaciones, y es aquí donde las metodologías ágiles aportan gran valor en el desarrollo.

En este trabajo se ha mostrado el uso la metodología ágil *Scrum* para gestionar eficientemente el desarrollo del taxímetro mediante la definición de *sprints* en los que se implementaron las historias de usuario de manera iterativa e incremental. La metodología facilitó el manejo de los cambios en la funcionalidad ya que algunas de ellas se modificaron, otras se crearon e incluso, algunas que se eliminaron. Se ha mostrado también que empleando UML para el análisis y diseño se puede hacer desarrollo ágil, buscando la calidad y mantenibilidad del software.

Dado que los sistemas de software son cada vez más complejos, no solo por la misma funcionalidad sino por el entorno en que deben ejecutarse (sistemas distribuidos, en red, multiplataforma, en la nube) y por la variedad de soluciones tecnológicas y metodológicas disponibles, las pruebas de software se enfrentan al problema de una gran combinación de escenarios que requieren de una combinación de las mejores prácticas en cada caso [30]. En el proyecto presentado se utilizaron varias estrategias de validación y verificación que incluyeron: revisiones, inspecciones, pruebas de unidad, pruebas del sistema y técnicas estadísticas. La combinación de estrategias permitiódetectar y corregir oportunamente los defectos en cada etapa de desarrollo, entregar un producto con menos defectos en pruebas evitando el *re-work*, reafirmando así que agilidad en el desarrollo no significa

sacrificar la calidad ni del proceso, ni del producto.

Con el uso de la estadística se pudo validar la confiabilidad del taxímetro sin requerir ejecutar innumerables casos de prueba, dado que el resultado de la medición se ve afectado por variables externas a la propia aplicación. Mediante un intervalo de confianza se pudo comprobar que la medición del taxímetro móvil es fiable en un 95 % en condiciones normales, pero con la presencia de factores de ruido (condiciones atmosféricas desfavorables, presencia de edificios muy altos y capacidad limitada de la tarjeta de red del dispositivo móvil) la señal del GPS es baja reduciendo así la exactitud y precisión del resultado.

Se espera que este taxímetro móvil, al cual hemos denominado *TwTaxi*, sea útil para los usuarios de taxi en la ciudad de Bogotá D.C. Se puede descargar gratuitamente de la tienda de aplicaciones *Aptoide*. Cabe mencionar que durante los primeros quince días de publicación la aplicación se descargó veintitrés veces y recibió comentarios positivos porque no maneja publicidad, se inicia en menos de cinco segundos y está integrada con Twitter, permitiendo así que el usuario haga pública su denuncia en caso de inconformidad en el cobro del servicio. Los modelos detallados del proyecto se encuentran en el portal web del grupo de investigación *ARQUIISOFT* <http://arquisoft.udistrital.edu.co/finalizados>.

5. Trabajo Futuro

La aplicación puede ser extendida para:

a) Hacer la implementación para otros sistemas operativos, es decir, que no solo esté disponible en *Android* sino que además se encuentre en *iOS*, *Windows Phone*, *Black Berry6*, *Symbian*, *Firefox O.S* y *Ubuntu Touch*.

b) Buscar tecnologías alternativas al GPS. Las medidas que se obtuvieron con el GPS presentaban algunas veces variaciones con respecto a las del taxímetro real en condiciones atmosféricas desfavorables, presencia de edificios muy altos y capacidad limitada de la tarjeta de red del dispositivo móvil. Aunque esta diferencia no fue significativa en las pruebas realizadas, se pueden buscar otras tecnologías que reduzcan el margen de error en la medición, por ejemplo *GLONASS* [31], un complemento del GPS para mejorar la precisión y que ya viene incorporado en algunos celulares de alta gama.

c) A partir de la información recolectada en el servidor *Parse* se puede evaluar el grado de satisfacción y conformidad por parte del usuario con relación al servicio prestado.

d) Evaluar nuevas prácticas y estrategias de pruebas para el desarrollo ágil con el fin de mejorar la velocidad y efectividad de las mismas, tomando como base el trabajo de Tulasi y Mani [32].

e) Evaluar el impacto de metodologías no ágiles para el desarrollo de aplicaciones móviles que consideren aspectos de calidad, restricciones de tiempo, integración de múltiples plataformas tecnológicas y agilidad en el desarrollo, con el fin de evaluar ventajas y desventajas frente a las metodologías ágiles.

Referencias

- [1] N. Amanquah and O. T. Eporwei, "Rapid application development for mobile terminals". in *2009 2nd International Conference on Adaptive Science & Technology (ICAST)*, 2009, pp. 410–417.
- [2] C.-C. Teng and R. Helps, "Mobile Application Development Essential New Directions for IT". in *IEEE*, 2010, pp. 471–475.
- [3] T. Vithani and A. Kumar, "Presentation 5. A comprehensive mobile application development and testing lifecycle". in *2014 IT Professional Conference*, 2014, pp. 1–3.
- [4] D. Sambasivan, N. John, S. Udayakumar, and R. Gupta, "Generic framework for mobile application development". in *2011 Second Asian Himalayas International Conference on Internet (AH-ICI)*, 2011, pp. 1–5.
- [5] Waze, Aplicación de mapas, tráfico y navegación gratuita, basada en la comunidad. [En línea]. Disponible en: <https://www.waze.com/es-419>. [Tomado el: 30-Nov-2015].
- [6] Sitp, "Muévete en el SITP con Moovit". [En línea]. Disponible en: http://www.sitp.gov.co/publicaciones/muevete_en_el_sitp_con_moovit_pub. [Tomado el: 30-Nov-2015].
- [7] Tappsi, "Tappsi - Taxi Seguro - Aplicaciones Android en Google Play". [En línea]. Disponible en: https://play.google.com/store/apps/details?id=com.tappsi.passenger.android&hl=es_419. [Tomado el: 30-Nov-2015].
- [8] Uber, "Uber - Aplicaciones Android en Google Play". [En línea]. Disponible en: https://play.google.com/store/apps/details?id=com.ubercab&hl=es_419. [Tomado el: 30-Nov-2015].
- [9] Play Google, "Taxímetro GPS - Aplicaciones Android en Google Play". [En línea]. Disponible en: https://play.google.com/store/apps/details?id=com.seeit.android.taximeter&hl=es_419. [Tomado el: 30-Nov-2015].
- [10] Play Google, "Bici-Guía - Aplicaciones Android en Google Play". [En línea]. Disponible en: https://play.google.com/store/apps/details?id=com.lacreativeria.taxiapp&hl=es_419. [Tomado el 30-Nov-2015].
- [11] A. Cardenas, "Hay cerca de 675 taxis por cada mil habitantes". *Diario ADN*, Bogotá D.C., p. 1,9, Aug-2012
- [12] M. Reyes, "Noticias Radio Ver". p. 1, Aug-2014.
- [13] "Global Positioning System: Signals, Measurements, and Performance (Revised Second Edition): Pratap Misra, Per Enge: 9780970954428: Amazon.com: Books". [En línea]. Disponible en: http://www.amazon.com/Global-Positioning-System-Measurements-Performance/dp/0970954425/ref=pd_sim_14_1?ie=UTF8&dpID=51BOJdeU9JL&dpSrc=sims&preST=_AC_UL160_SR112%2C160_&r_eFRID=0TXK4C2MCHQV6QQY7T53. [Tomado el: 03-Dec-2015].
- [14] D. Cuadrado, "Las pruebas del nuevo taxímetro con GPS obtienen buenas notas". *Autopista.es*, Dec-2000.
- [15] L. Creativería, "Taxiando - Aplicaciones Android en Google PlayNo Title". [En línea]. Disponible en: https://play.google.com/store/apps/details?id=com.lacreativeria.taxiapp&hl=es_419. [Tomado el: 08-Dec-2015].
- [16] Secretaría Distrital de Movilidad, "DECRETO 400 DE 2014". in *Registro Distrital 5439 de 2014*, Bogotá D.C. 2014.
- [17] Alcaldía Mayor de Bogotá, "Decreto No. 237 de 2006". 2006.
- [18] V. autores, "NTC 3679. TAXÍMETROS ELECTRÓNICOS". ICONTEC.
- [19] H. Higuera, S.R., Ramírez, M.E., y Arguello, "Prototipo de Práctica, herramienta software orientado a dispositivos móviles para apoyar la práctica médica en el área de la práctica ortodoncia," *Ingeniería*, vol. 16, no. No.1, pp. 36–49, 2011.
- [20] "Scrum Methodology". [En línea]. Disponible en: <http://scrummethodology.com/>. [Tomado el: 02-Dec-2015].
- [21] "Manifiesto por el Desarrollo Ágil de Software". [En línea]. Disponible en: <http://agilemanifesto.org/iso/es/>. [Tomado el: 02-Dec- 2015].
- [22] "iceScrum, Open Source Scrum & Agile project management tool". [En línea]. Disponible en: <https://www.icescrum.com/>. [Tomado el: 07-Dec-2015].
- [23] M. Silvia Tabares and J. D. Barrera, Andrés Felipe Arroyave, Juan David Pineda, "Un método para la trazabilidad de requisitos en el proceso unificado de desarrollo". *EIA*, pp. 69–82, 2007.
- [24] J. Conejero and J. Hernández, "Analysis of Crosscutting Features". *ACM*, pp. 3–10, 2008.
- [25] A. Babativa and P. Briceño, "Modelos funcionales y estructurales TwTaxi". [En línea]. Disponible en: <http://arquisoft.udistrital.edu.co/finalizados>. [Tomado el: 07-Dec-2015].
- [26] F. A. Amo, L. M. Normand, and F. J. S. Pérez, *Introducción a la ingeniería del software*. Delta Publicaciones, 2005.
- [27] I. Sommerville, *Software Engineering*. 2011.
- [28] W. S. Humphrey, *Introduction to the Team Software Process*. Massachusetts: Addison Wesley, 2000.

- [29] D. C. Montgomery, *Diseño y análisis de experimentos*. México: LIMUSA WILEY, 2014.
- [30] C. Nie and H. Leung, "A survey of combinatorial testing". *ACM Comput. Surv.*, vol. 43, no. 2, pp. 1–29, 2011.
- [31] C. Hackman, S. M. Byram, V. J. Slabinski, J. C. Tracey, U. States, and N. Observatory, "USNO GPS / GLONASS PNT Products : Overview , and GPS + GLONASS vs GLONASS Only PPP Accuracy". pp. 793–803, 2007.
- [32] T. Anand and V. S. Mani, "Practices to make agile test teams effective : challenges and solutions". pp. 7–11, 2015.

Apéndice: muestras tomadas para validación

La Figura 5 muestra los resultados que se obtuvieron al realizar las pruebas, donde:

Unidades Taxímetro: Unidades registradas por el taxímetro del vehículo.
Unidades TwTaxi: Unidades registradas por la aplicación TwTaxi.

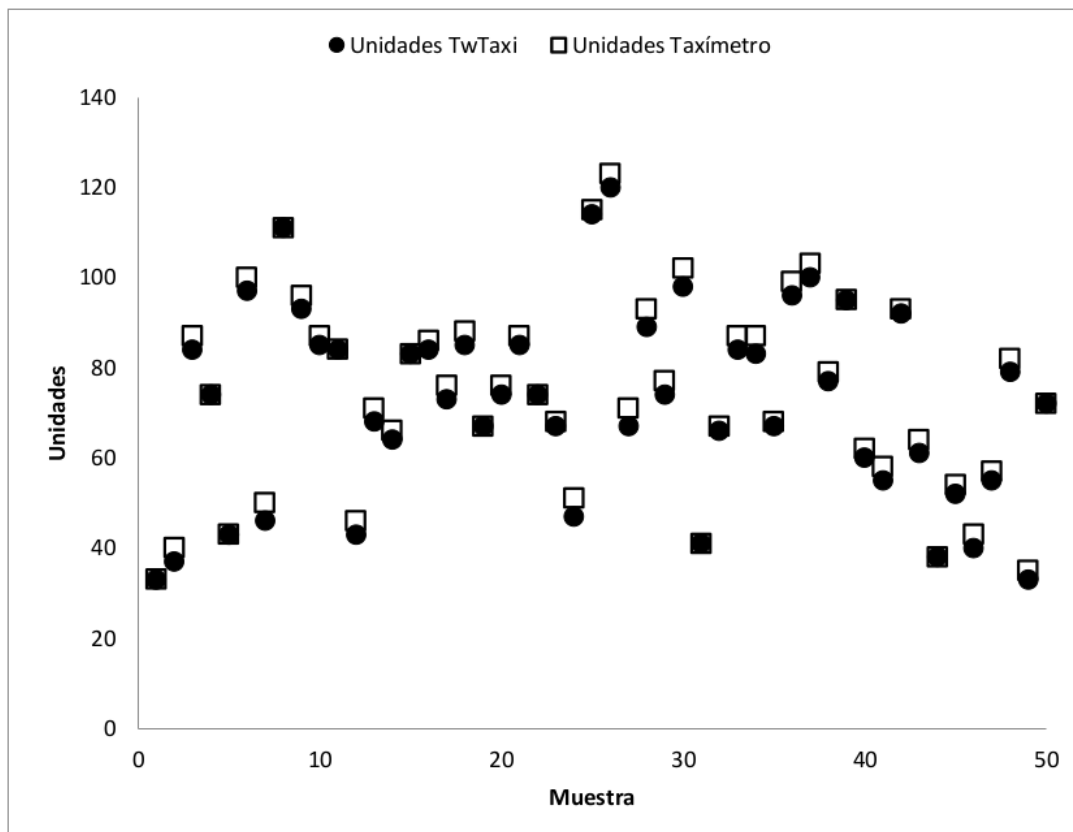


Figura 5. Unidades registradas por TwTaxi versus Taxímetro real

Angélica Babativa.

Ingeniera de sistemas, Universidad Distrital Francisco José de Caldas; estudiante Especialización en Proyectos Informáticos, Universidad Distrital Francisco José de Caldas; analista especializado en Banco Av Villas; sus intereses son las bases de datos y aplicaciones móviles. Correo electrónico: ambabativag@correo.udistrital.edu.co

Paula Briceño.

Ingeniera de sistemas, Universidad Distrital Francisco José de Caldas; analista Técnica en la Empresa TGO Ltda. para el Banco de Bogotá; sus intereses son las bases de datos, aplicaciones móviles, desarrollo de aplicaciones Web. Correo electrónico: pdbricenon@correo.udistrital.edu.co

Consuelo Nieto.

Ingeniera de sistemas, Universidad Nacional de Colombia; especialista en Transmisión de Datos, Cámara Argentina de Telecomunicaciones; magíster en Ingeniería de Sistemas y Computación, Universidad de los Andes; docente de la Facultad de Ingeniería de la Universidad Distrital Francisco José de Caldas. Correo electrónico: acnietol@udistrital.edu.co

Omar Salazar.

Ingeniero Electrónico, Universidad Distrital Francisco José de Caldas; magíster en Ingeniería Industrial, Universidad Distrital Francisco José de Caldas; magíster en Ciencias Matemáticas, Universidad Nacional de Colombia; docente de la Facultad de Ingeniería de la Universidad Distrital Francisco José de Caldas. Correo electrónico: osalazarm@correo.udistrital.edu.co