

# La Ingeniería de Software en el desarrollo de aplicaciones para dispositivos móviles

## Software engineering in the development of applications for mobile devices

**Fabio Alberto Vargas Agudelo**  
Magíster en ingeniería de sistemas,  
e-mail: fvargas@tdea.edu.co,  
Tecnológico de Antioquia

### Resumen

El artículo presenta la aplicación del ciclo de vida del software en el desarrollo de aplicaciones para dispositivos móviles, teniendo muy en cuenta las diferencias existentes cuando el desarrollo se hace pensando en aplicaciones cliente-servidor o inclusive en aplicaciones web. Se describe en cada etapa del ciclo de vida el proceso que se debe seguir a partir de la concesión de una construcción móvil.

**Palabras claves:** dispositivos móviles, ciclo de vida del software, desarrollo de software.

### Abstract

This paper shows the application of software lifecycle in the development of applications for mobile devices, considering the differences when development is intended for client-server and/or web applications. The process to be followed, after a mobile construction being granted, is described for each stage of the lifecycle.

**Keywords:** Mobile devices, software life cycle, software development.

## Introducción

El desarrollo de software requiere de un proceso planeado y estandarizado si se quiere generar productos de alta calidad, tanto en su documentación como en su aplicativo final. Para ello existen varios métodos de ingeniería de software que establecen un conjunto de entregables con el ánimo de dar una trazabilidad al producto, y asegurar que pueda ser interpretado, actualizado y adaptado fácilmente por los usuarios finales. Los métodos de ingeniería de software RUP (Kruchten, 1999), CDM (Oracle 2000), FDD (Coad y Lefebvre, 1999), XP (Beck, 2000) y UN-Método (Zapata *et al.*, 2006) fueron concebidos para el desarrollo de aplicaciones cliente-servidor, sin tener muy en cuenta el avance de los diferentes ambientes de desarrollo móvil y web, los cuales requieren un sinnúmero de particularidades que hacen insuficientes las etapas de ciclo del vida del software para responder a las necesidades de los interesados (Zapata y Vargas, 2009). El artículo aborda el ciclo de vida del desarrollo de software sugerido en los métodos de desarrollo de software mencionados anteriormente y plantea un conjunto aspectos que deben tenerse en cuenta en cada fase (definición de requisitos, análisis, diseño, desarrollo, pruebas e implementación) para crear un aplicativo móvil de alta calidad.

Este artículo tiene la siguiente estructura: en la sección 2 se presenta una revisión de los trabajos referentes a la utilización de la ingeniería de software en aplicaciones móviles; posteriormente, en la sección 3 se realiza una descripción del proceso de cada fase de la ingeniería de software para una aplicación móvil y en la sección 4 se presentan las conclusiones.

## Antecedentes

La ingeniería de software se ha aplicado al desarrollo de software desde hace treinta años aproximadamente, lo cual ha contribuido directamente a mejorar la calidad y a garantizar el cumplimiento de las necesidades de los usuarios o interesados en el producto (Lamsweerde, 2000).

Recientemente, los dispositivos móviles —como teléfonos, asistentes personales, *Ipod*, *smartphones*, computadores portátiles, relojes y otros— han cobrado gran popularidad en diferentes ámbitos sociales, económicos y políticos. Estos dispositivos han permiti-

do construir aplicaciones de gran alcance, adaptadas a cambios de contexto, como las variaciones en el ancho de banda, batería, conectividad, accesibilidad de los servicios, diseño de interfaces, capacidad de memoria, tamaño, etc.

Capra *et al.* (2003) describe un *middleware* de computación móvil que explota el principio de reflexión para mejorar la adaptación y sensibilidad al contexto en la construcción de aplicaciones móviles. Sin embargo, Mascolo *et al.* (2002) plantea que la aplicación de la ingeniería de software para dispositivos móviles es tediosa y propensa a errores, debido a que se debe tener en cuenta muchos aspectos en la definición de requisitos no funcionales, tales como la heterogeneidad y tolerancia a fallos, así como aspectos de seguridad y de rendimiento.

Los desarrollos alrededor de la computación ubicua requieren el despliegue de aplicaciones soportadas en la ingeniería de software que puedan sobrevivir el uso diario, se acomoden a las necesidades del entorno, cumplan realmente funciones productivas sin convertirse simplemente en una moda (Abowd & Mynatt, 2000). El reto para la ingeniería de software es diseñar notaciones, técnicas, artefactos, métodos y herramientas para construir sistemas para dispositivos móviles (Emmerich, 2000).

## Ciclo de vida del software para dispositivos móviles

Cuando se piensa desarrollar una aplicación para un dispositivo móvil en cualquiera de las plataformas y para cualquier entorno es de vital importancia reconocer y establecer condiciones que garanticen la pertinencia, la calidad, la seguridad, la eficiencia y el rendimiento del programa que se desea construir y utilizar por medio del dispositivo móvil (Fling, 2009). Por tal razón es de suma importancia seguir en forma clara las etapas generales del ciclo de vida del software (definición y análisis de requisitos, diseño, desarrollo, pruebas, y mantenimiento), pero teniendo muy presentes las grandes diferencias que existen entre el desarrollo de una aplicación para ejecutar en un PC de escritorio y el de una aplicación para ejecutar en un dispositivo móvil.

Cada etapa debe ir enfocada a establecer muy claramente qué es lo que se busca en una pieza de

software para un dispositivo móvil, que garantice la movilidad, el fácil uso y el aprovechamiento de los limitados recursos de memoria.

## Definición de requisitos

La primera fase para desarrollar una aplicación móvil es establecer una buena especificación de requisitos, ya que esta es la base para el buen rendimiento, la eficiencia y la calidad de la pieza de software por desarrollar. La claridad sobre todos los requisitos de los usuarios y el problema que se desea abordar permite tomar decisiones fundamentales acerca del tipo de aplicación requerida, así como también sobre el tipo de dispositivo por utilizar, pues hay claras diferencias entre una aplicación web móvil y una residente en el dispositivo. Además, se determina si la aplicación justifica un desarrollo móvil o si es recomendable una aplicación de servidor que se ejecute en un computador de escritorio normal. Así mismo, es muy importante definir los requisitos no funcionales con respecto a la seguridad, la funcionalidad, la capacidad y la cobertura.

Lo anterior permitirá establecer y definir en un documento inicial la viabilidad de la aplicación móvil que se va a desarrollar, tras determinar su inicio o en caso de cambios de planes, y a partir de allí descartar una aplicación móvil como solución para la organización.

## Análisis de requisitos

Una vez determinada la viabilidad de la aplicación móvil y definidos los requisitos del futuro sistema, se deben identificar y representar las entradas, los procesos, las salidas y los usuarios que intervienen directamente en la aplicación móvil, así como las estructuras de datos requeridas. La representación de estos aspectos se debe realizar mediante diagramas de UML (Larman, 2002), utilizando el diagrama de procesos (Villanueva *et al.*, 2005), el diagrama de casos de uso (Larman, 2002) y el diagrama de clases (Larman, 2002).

## Diseño

En la fase de diseño, también conocida como de desarrollo de prototipos, se comienza a reflejar de

forma muy clara los aspectos visuales y técnicos que dan vida a la aplicación móvil. Durante esta etapa, se genera una retroalimentación por parte del usuario, quien se encarga de hacer cumplir los requisitos de funcionalidad y diseño esperados —es muy importante definir aspectos funcionales con el cliente en cuanto a la navegabilidad, desplazamientos por el dispositivo y, lo más importante, la forma de carga, almacenamiento y presentación de los datos solicitados y necesarios para el proceso requerido del aplicativo móvil—.

Para representar las interfaces de usuario propuestas en el diseño se debe utilizar emuladores de dispositivos móviles presentes en un conjunto de herramientas libres, los cuales permitan ir realizando modificaciones solicitadas y sugeridas en el diseño de la interfaz, pero también se debe ir probando en dispositivos reales, de tal manera que se garantice un mayor grado de adaptabilidad a las diferentes tecnologías y dispositivos existentes.

## Desarrollo

Una vez el usuario defina los requisitos y se establezcan las condiciones de diseño aprobadas en las fases anteriores se pasa a la etapa en la cual se escribe el código fuente necesario para dar funcionamiento a la pieza de software móvil. Para ello se debe tener conocimiento de un lenguaje de programación móvil adaptable a las necesidades de diseño previamente concretadas con los usuarios directos de la aplicación.

## Pruebas

Una vez culminado el desarrollo de la pieza de software definida y establecida con el usuario es de vital importancia verificar y evaluar el funcionamiento y la calidad de la pieza de software resultante con el fin de evitar sobrecostos por ajustes posteriores al software. Además, debe verificarse que la pieza de software cumpla con los requisitos definidos en etapas anteriores y genere los resultados esperados por los usuarios. Para lograr este objetivo es necesario ejecutar la aplicación en el dispositivo móvil seleccionado. Igualmente, se debe realizar un conjunto de pruebas al software móvil desarrollado, que vayan desde las pruebas unitarias hasta las pruebas funcionales con el fin de garantizar la satisfacción total del usuario.

Es muy importante aclarar que un programador jamás debería entregar una pieza de software móvil sin haberlo probado, y a su vez, un usuario no debería recibir la aplicación si haber verificado dicha prueba. Para realizar la entrega de la pieza de software se deben llevar a cabo las siguientes pruebas:

*Pruebas unitarias:* cuando se construye un aplicativo móvil, es de vital importancia comprobar que el código fuente escrito funcione correctamente y se adapte al funcionamiento permanente del programa.

*Pruebas de integración:* se debe probar no solo el funcionamiento individual de la pieza de software móvil, sino también su integración con los demás componentes de la aplicación, especialmente con el servidor de datos con el cual se realiza el proceso de sincronización de la información utilizada en el dispositivo.

*Pruebas de validación:* este proceso tiene como objetivo determinar si la aplicación móvil cumple los objetivos para los que se construyó el producto.

Estas pruebas sirven para determinar que se hayan logrado todas las metas definidas en las fases anteriores, muy especialmente los requisitos funcionales especificados por el usuario, así como también en las características de diseño establecidas.

*Prueba de recuperación:* aquí se examina la capacidad del aplicativo móvil de recuperar el sistema luego de bloqueos, fallas en el dispositivo y descargas.

*Prueba de seguridad:* evalúa la capacidad de la aplicación móvil para protegerse contra accesos no autorizados y de limitar sus operaciones generales a un conjunto de usuarios determinados en las fases anteriores.

*Prueba de compatibilidad:* aquí se revisa el desempeño de la pieza de software móvil entre diferentes dispositivos que manejen diferente tecnología y apariencia a fin de garantizar su diseño y rendimiento. Es muy importante que un aplicativo móvil se acomode a una gran cantidad de dispositivos, formas de navegación y tecnologías existentes.

## Instalación y mantenimiento

Luego de cumplir con todos las fases anteriores, es decir, de haber realizado una buena definición de requisitos, un análisis, un diseño acorde a las nece-

sidades y de codificar y probar la aplicación —por ende garantizando la satisfacción de las necesidades de los usuarios del aplicativo móvil—, se procede a la instalación del software en el dispositivo (en caso de ser necesario) y a la puesta en marcha del mismo, así como a la capacitación de los usuarios.

## Conclusiones

La computación móvil permite, independientemente del lugar o tiempo, realizar diferentes actividades de procesamiento en los dispositivos móviles. Se realizan desde aplicaciones de entretenimiento, mensajería, comercio electrónico móvil (*m-commerce*), entre otras que apenas comienzan a desplegarse en nuestro país, como servicios de multimedia, de vigilancia, y colaboración en servicios médicos.

La construcción de aplicaciones de software para dispositivos móviles de alta calidad demanda de los desarrolladores el uso de métodos de ingeniería de software para lograr la satisfacción total de los requisitos de los usuarios y permitir a la vez una fácil actualización y mantenimiento del software por un largo periodo de tiempo.

El software móvil es un campo de acción que aún no ha alcanzado el grado de uso que esto genera, debido a que todavía las personas y las organizaciones desconfían de la seguridad de estos recursos para administrar información personal y, especialmente, organizacional.

Es muy importante crear métodos de ingeniería de software adaptables al desarrollo móvil, que tengan en cuenta sus particularidades y permitan al desarrollador el crear productos de alta calidad.

## Referencias

- Abowd, G. y Mynatt, E. (2000). "Charting past, present, and future research in ubiquitous computing". *ACM Trans. Comput.-Hum. Interact.* 7, 1, pp. 29-58.
- Beck, K. (2000). *Extreme Programming Explained: Embrace Change*. Boston, Addison-Wesley.
- Capra, L., Emmerich, W., Mascolo, C. (2003). "CARISMA: Context-Aware Reflective middleware System for Mobile Applications". *IEEE Transactions on Software Engineering*, vol. 29, no. 10, pp. 929-945.

Coad, P. y Lefebvre, E. (1999). *Java Modeling in Color with UML*. Nueva York, Prentice Hall.

Emmerich, W. (2000). Software engineering for middleware: a roadmap. En A. Finkelstein, editor, *The Future of Software Engineering*. Edición especial publicada en asociación con ICSE 2000.

Fling B. (2009). *Mobile Design and Development*. O'Reilly, ISBN: 978-0-596-15544-5.

Kruchten, Ph. (1999). *Rational Unified Process—An Introduction*. Boston, Addison-Wesley.

Larman, C. (2002). *UML y patrones*, 2ª ed. Madrid, Prentice-Hall.

Lamsweerde A. (2000). Requirements engineering in the year 2000: a research perspective. Proceeding of the 22nd International Conference on Software Engineering, Limerick (Irlanda), 5-19.

Mascolo, C., Capra, L. y Emmerich, W. (2002). Middleware for Mobile Computing (A Survey). En: E. Gregori, G. Anastasi y S. Basagni, editores, *Networking 2002 Tutorial Papers*, LNCS 2497, pp. 20-58.

Oracle® Corporation (2000). *Oracle MethodSM CDM Quick Tour*. Redwood City, Oracle Corporation.

Villanueva, I., Sánchez, J., Pastor, O. (2005). "Elicitación de requisitos en sistemas de gestión orientados a procesos", *VIII Seminario sobre Ingeniería de Requerimientos (WER'05)*, Porto, Portugal.

Zapata, C., Vargas, F. (2009). Una revisión de la literatura en consistencia entre problemas y objetivos en ingeniería de software y gerencia organizacional. *Revista ELA*, Número 11, pp. 117-129.

Zapata, C., Villegas, S. y Arango F. (2006). Reglas de consistencia entre modelos de requisitos de UN-Método. *Revista Universidad Eafit*, 42(141):40-59.