



## EVALUACIÓN DE LAS CARACTERÍSTICAS DE UN ENFOQUE METODOLÓGICO PARA DESARROLLAR SOFTWARE EN CURSOS UNIVERSITARIOS

Ramón Ventura Roque Hernández<sup>1</sup>, Adán López Mendoza<sup>2</sup>,  
Carlos Manuel Juárez Ibarra<sup>3</sup>, Salvador Mota Martínez<sup>4</sup>

*Universidad Autónoma de Tamaulipas. Facultad de Comercio, Administración y Ciencias Sociales  
rvHernandez@uat.edu.mx<sup>1</sup>, Lerdo de Tejada 1218 Sector Centro, Nuevo Laredo, Tamaulipas, México, (867)  
129 0303, aLopez@uat.edu.mx<sup>2</sup>, Ayuntamiento Sur S/N Col. Infonavit Fundadores, Nuevo Laredo,  
Tamaulipas, México, (867) 719 42 50, cJuarez@uat.edu.mx<sup>3</sup>, Ayuntamiento Sur S/N Col. Infonavit  
Fundadores, Nuevo Laredo, Tamaulipas, México, (867) 711 13 13, Ayuntamiento Sur S/N Col. Infonavit  
Fundadores, Nuevo Laredo, Tamaulipas, México, (867) 711 13 13.*

*Fecha de envío: 08/Abril/2016*

*Fecha de aceptación: 16/Mayo/2016*

### Resumen

Este artículo presenta una investigación que se realizó con el objetivo de evaluar el Proceso Unificado en el desarrollo de software que se realiza como parte de los cursos universitarios. Se tomaron en cuenta las percepciones de veintiocho estudiantes del programa académico de licenciatura en informática. Se estudiaron los siguientes aspectos: acerca del enfoque metodológico utilizado (facilidad de entenderlo, facilidad de implementarlo, facilidad de adaptarlo, eficiencia general) y acerca de la dinámica de trabajo de los participantes (organización, comunicación, motivación). Se utilizó un cuestionario con el cual se pudieron caracterizar las evaluaciones a estos aspectos y conocer cuáles de ellos resultaron mejor y peor evaluados. En el análisis de datos se condujo una prueba de Friedman con comparaciones posteriores para establecer diferencias estadísticamente significativas. Comunicación, organización y motivación fueron los aspectos que recibieron las evaluaciones más altas; el resto obtuvo las puntuaciones más bajas.

### Palabras clave:

Educación, Evaluación, Programación, Software, Universidad

## **Introducción**

Desarrollar software no es una labor sencilla (Roque, Salinas, & López, 2015). Es común que la complejidad proveniente de las áreas técnica, administrativa y del dominio de la aplicación superen la capacidad del equipo de desarrollo para manejar simultáneamente varias tareas. Esto trae como consecuencia que los proyectos se retrasen, sobrepasen el presupuesto que fue establecido para ellos, o bien, resulten inadecuados o insuficientes para responder a los requerimientos que los originaron. En resumen, las situaciones pueden salirse de control fácilmente y provocar serias consecuencias.

Por esta razón, los enfoques metodológicos resultan útiles para conceptualizar y organizar las principales actividades relacionadas con el desarrollo de software (análisis, diseño, programación, pruebas, puesta en marcha, mantenimiento). De esta manera, la probabilidad de fracaso del proyecto puede verse disminuida. Si bien es cierto que un enfoque metodológico no es una garantía absoluta de solución a los problemas mencionados anteriormente, sí proporciona recursos con los cuales los integrantes del equipo pueden gestionar su trabajo de una manera menos caótica que si no siguieran ninguna metodología.

En los cursos universitarios de desarrollo de software, los estudiantes deben aprender sobre la creación de programas de cómputo y sobre el proceso para realizarlos. Los enfoques metodológicos aportan a los estudiantes recursos que enriquecen su formación en estos dos aspectos. Existen muchos de estos enfoques que podrían abordarse en las aulas desde una perspectiva teórica exclusivamente; sin embargo, los modelos académicos actuales basados en competencias exigen en los alumnos el desarrollo de tres componentes: conocimientos, habilidades y valores. Por esta razón, un enfoque metodológico debería ser abordado con

teoría, práctica y trabajo colaborativo que promoviera las buenas prácticas, la comunicación entre los integrantes, así como una buena organización para trabajar.

La investigación que se presenta en este trabajo se realizó con el objetivo de describir la aplicación del Proceso Unificado de desarrollo de software en el entorno universitario en relación a diversos aspectos tales como: la facilidad de entenderlo, la facilidad de implementarlo, la facilidad de adaptarlo, así como la organización, la comunicación y la motivación que tuvieron los estudiantes durante su uso en una materia de su carrera profesional. Para lograr este objetivo, se analizaron las evaluaciones que los participantes otorgaron a cada uno de estos aspectos a través de un cuestionario.

Este artículo tiene la siguiente estructura: primero se presenta el marco teórico, en donde se exponen las actividades generales del proceso de desarrollo de software, así como los fundamentos del Proceso Unificado, el uso de metodologías de desarrollo de software en los cursos universitarios y la filosofía de la investigación empírica dentro del área de ingeniería de software. Posteriormente, se explica la metodología utilizada en esta investigación. Luego se exponen los resultados que se obtuvieron y finalmente se presentan las conclusiones.

## **Marco teórico**

### **El proceso de desarrollo de software**

El desarrollo de sistemas de software se realiza a través de un conjunto de actividades compuestas por: análisis, diseño, programación, pruebas, conversión, producción y mantenimiento (Laudon & Laudon, 2011). Cada enfoque metodológico puede organizar estas actividades de manera diferente para lograr una configuración apropiada a su filosofía. Se debe tomar en cuenta que la aplicación de un enfoque metodológico en distintos escenarios puede producir resultados distintos.

La primera actividad es el análisis de sistemas, que se encarga de la definición del problema que se va a resolver con el software que se va a desarrollar; esta tarea incluye también la extracción y determinación de los requerimientos del usuario, los cuales son la columna vertebral del programa de cómputo que se está elaborando. El diseño de sistemas, por su parte, se encarga de especificar las características del software; a través de él se conocen los detalles que describen entradas, salidas, procesamientos, interfaces, y almacenamiento. La programación, por el contrario, es la actividad de escribir código fuente utilizando un lenguaje específico; para esta tarea, es fundamental contar con los detalles de diseño. De esta manera, se van escribiendo una a una las funciones que se espera que el software realice. La programación está encaminada a la obtención de un sistema de calidad con el cual el usuario pueda interactuar.

Las pruebas permiten determinar si el programa entrega los resultados esperados bajo condiciones concretas. Se deben probar todos los componentes del software, primero como unidad y posteriormente en conjunto para comprobar el efecto de sus interacciones. También se debe conducir una prueba de aceptación, en donde el usuario evalúa el

programa construido. Después de esta actividad, se continúa con un proceso denominado conversión en el que el nuevo software es introducido en las operaciones cotidianas del usuario, reemplazando así al sistema anterior. Cuando el nuevo programa se encuentra totalmente instalado, se dice que el programa está en producción, en donde se realizan revisiones y ajustes; estas tareas conforman las actividades de mantenimiento. El mantenimiento puede corregir errores, optimizar procesos, o bien, adaptar el software a nuevos requerimientos del usuario.

### **El Proceso Unificado (PU)**

El Proceso Unificado (PU) (Jacobson, Booch, & Rumbaugh, 2005) es un marco de trabajo para el desarrollo de software. Su objetivo principal es la transformación de un conjunto de requerimientos de usuario en un programa de cómputo plenamente funcional que los satisfaga. Este enfoque se apoya en el uso del UML (Lenguaje Unificado de Modelado) (Rumbaugh, Jacobson, & Booch, 2000), (Schach, 2005) como herramienta de comunicación, especificación y documentación durante todo el tiempo de creación del software. En la literatura se identifica que el Proceso Unificado es: 1) dirigido por casos de uso, 2) centrado en la arquitectura, 3) iterativo, 4) incremental.

Los casos de uso son las características funcionales que el programa debe implementar. Cada caso de uso se relaciona directamente con los requerimientos que el usuario plantea al inicio del proceso. Por esta razón, los casos de uso son la directriz principal del proceso completo. Al centrar la atención en ellos, se pueden focalizar los esfuerzos de los participantes hacia la producción de un software que cubra las necesidades por las que fue creado.

Por otra parte, el PU encamina todas sus actividades hacia la creación de la arquitectura del sistema, que es una representación de todo el software que se está desarrollando y contiene únicamente detalles relevantes. La arquitectura tiene componentes estructurales y dinámicos. Los primeros definen los elementos estáticos del sistema, tales como clases u objetos; los segundos especifican las relaciones e interacción entre los elementos estáticos en escenarios concretos dentro del sistema.

Todas las actividades del PU se enmarcan en varias secuencias de tiempo definidas como iteraciones. Con este enfoque, el alcance global del programa es fragmentado en varios objetivos de menor alcance, los cuales se van logrando paulatinamente en cada iteración. Cada iteración agrega nuevo valor al software, ya que las versiones intermedias que se obtienen son cada vez más cercanas a la versión final que se entregará al usuario al término del proceso. Por esta razón se dice que el proceso es iterativo e incremental.

El total de iteraciones del proyecto se realiza a lo largo de cuatro fases denominadas: inicio, elaboración, construcción y transición. La fase de inicio consiste principalmente en la identificación de los requerimientos del software que se creará. En la fase de elaboración, el esfuerzo se concentra en la creación de la arquitectura del software. En la fase de construcción, se trabaja en el desarrollo de las funciones necesarias para el cumplimiento de los requerimientos, al mismo tiempo que se retroalimenta la arquitectura con la finalidad de representar el sistema de una manera cada vez más completa. La fase de transición culmina con una versión estable del software, la cual es entregada a los usuarios finales. En esta fase también se realizan las adaptaciones necesarias para que la funcionalidad del sistema sea la requerida por los usuarios.

### **El uso de metodologías de desarrollo de software en los cursos universitarios**

Las materias de programación que se imparten en las universidades con carreras profesionales de informática incluyen el desarrollo de proyectos de software que deben completarse durante un periodo escolar regular. En muchos de los casos, estos proyectos son realizados por los alumnos sin haber seguido un enfoque metodológico, ya que se busca el logro rápido de un software funcional sin tener las buenas prácticas en el proceso de desarrollo como una prioridad. Estos escenarios son comprensibles ya que el contenido de estos cursos está centrado en la creación de un sistema de software, para lo cual se deben estudiar lenguajes específicos de programación, así como estructuras, técnicas y algoritmos. Existen otras materias cuyo perfil, por el contrario, pone en mayor relieve los aspectos metodológicos del desarrollo de software. Estos cursos centran su estudio más en la ingeniería de software, es decir, en el proceso completo de creación de un programa de cómputo.

En ambos tipos de materias, el uso de un enfoque metodológico es altamente benéfico. Los alumnos aprenden sobre el producto y el proceso simultáneamente, se promueven las buenas prácticas, el trabajo organizado, así como la comunicación entre los participantes, quienes además, pueden verse motivados por la manera de desarrollar software que se les propone en clase.

### **La investigación de la ingeniería de software basada en la práctica**

La ingeniería de software es el área de conocimiento que estudia los aspectos propios del desarrollo de software y los orienta hacia la aplicación de un enfoque sistemático, cuantificable y centrado en la calidad (Pressman, 2014). Para comprobar teorías, validar supuestos, tomar decisiones informadas y evaluar distintas posibilidades en este campo de

estudio, se puede emplear un enfoque investigativo empírico (Juristo & M. Moreno, 2001). Con este objetivo, se diseña una modalidad experimental y se recaban datos provenientes directamente de la práctica. Si el estudio tiene elementos cuantitativos, se puede conducir un análisis de datos con técnicas estadísticas tradicionales y se llega a una conclusión, que posteriormente derivará en una toma de decisiones. Estas aportaciones son valiosas porque proporcionan nuevas perspectivas y aproximaciones a la verdad acerca de los escenarios específicos que son objeto de estudio.



## Método

Este trabajo se realizó con los siguientes objetivos: 1) caracterizar las evaluaciones que los estudiantes otorgan a distintos aspectos relacionados con una metodología de desarrollo de software que se utiliza durante un curso universitario de programación y 2) conocer cuáles de estos aspectos reciben las puntuaciones significativamente más altas y más bajas. Las características de interés de esta investigación se presentan en la Tabla 1.

## Tabla

1.

Aspectos estudiados en esta investigación.

<i>Categoría</i>	<i>Número</i>	<i>Descripción del aspecto sobre el enfoque metodológico</i>
<i>Aspectos</i>	1	Facilidad de entenderlo
<i>prácticos</i>	2	Facilidad de implementarlo
<i>de</i>	3	Facilidad de adaptarlo
<i>la</i>		
<i>metodología</i>		
<i>Aspectos</i>	4	Organización de los integrantes
<i>de</i>	5	Comunicación de los integrantes
<i>la</i>		
<i>dinámica</i>	6	Motivación del equipo
<i>del equipo</i>		
<i>de trabajo</i>		
<i>Eficiencia</i>	7	Eficiencia general
<i>general</i>		

Las siguientes son las definiciones de las características que se plantearon a los estudiantes en esta investigación:

- a) Facilidad de entender el enfoque metodológico.- Se refiere a que los participantes alcancen con facilidad un nivel suficiente de comprensión sobre las actividades que son parte de la metodología que utilizan.
- b) Facilidad de implementar el enfoque metodológico.- Se refiere al esfuerzo que realizan los participantes para poner en marcha las actividades de la metodología.
- c) Facilidad de adaptar el enfoque metodológico.- Se refiere al esfuerzo con que el participante percibe que puede adecuar las actividades de la metodología a un contexto específico de trabajo.
- d) Organización de los integrantes del equipo.- Es el grado con el cual la metodología permite a los participantes establecer prioridades en sus tareas y lograr un trabajo estructurado.
- e) Comunicación de los integrantes del equipo.- Es el grado con el cual la metodología fomenta entre los participantes el intercambio de información.
- f) Motivación del equipo.- Es el grado con el cual la metodología produce en los participantes una conducta activa, dispuesta y constante para la realización de las actividades de desarrollo de software.
- g) Eficiencia general.- Es el grado en el cual los participantes perciben que la metodología tiene una capacidad adecuada de la que se puede disponer para desarrollar software.

Para esta investigación, se trabajó con un grupo de 28 estudiantes universitarios de la carrera de Licenciatura en informática. Todos ellos cursaban la segunda mitad de su carrera profesional y tenían un nivel de experiencia homogéneo. A los participantes se les organizó en equipos de máximo tres integrantes y se les solicitó el desarrollo de una aplicación de cien requerimientos como máximo para satisfacer las necesidades reales expuestas por varios usuarios finales como parte de un curso curricular enfocado a la creación de programas de cómputo. El tiempo de trabajo promedio fue de dos meses, durante los cuales se realizaron dos iteraciones orientadas a una versión cada vez más completa de la aplicación que se estaba desarrollando. Cuando el software se concluyó, se solicitó a cada participante su evaluación hacia los aspectos investigados descritos en la Tabla 1. Para las posibles respuestas, a los estudiantes se les pidió utilizar una misma escala que consistió en un valor numérico entre 0 y 4 para cada una de las características. Este valor representaba en todos los casos la percepción del participante hacia el aspecto que se le solicitaba evaluar. La escala era ordinal: el 0 significaba una percepción muy mala mientras que el 4 representaba una calificación excelente.

Las respuestas de los participantes fueron introducidas en el software estadístico SPSS (Wagner, 2014), en donde se llevó a cabo el análisis cuantitativo de los datos. Primero, se obtuvo el valor de la mediana para cada uno de los aspectos estudiados y se observaron diferencias aritméticas. Posteriormente, se realizó la prueba no paramétrica de Friedman (Kuanli, Pavur, & Keeling, 2006) con el objetivo de determinar si las diferencias observadas en el conjunto de todos los aspectos podían considerarse significativas estadísticamente. Las hipótesis propuestas son las siguientes:

$H_0$ : No existen diferencias estadísticas significativas en las evaluaciones otorgadas a los aspectos estudiados.

$H_1$ : Existen diferencias estadísticas significativas en las evaluaciones otorgadas a los aspectos estudiados.

Como se encontraron diferencias, se realizaron comparaciones posteriores por pares entre los distintos aspectos estudiados para analizar su origen con más profundidad.

En todos los casos se utilizó como referencia un nivel de confianza del 95%. De esta manera, cualquier PValor menor que .05 en las pruebas de hipótesis indicó la existencia de diferencias estadísticas significativas.

## **Resultados**

Los valores de la Mediana para cada uno de los aspectos que se analizaron se encuentran en la Tabla 2.

**Tabla**

**2.**

Medianas de las evaluaciones obtenidas para los aspectos estudiados de la metodología.

<i>Categoría</i>	<i>Número</i>	<i>Descripción del aspecto sobre el enfoque metodológico</i>	<i>Mediana</i>
<i>Aspectos</i>	1	Facilidad de entenderlo	3
<i>prácticos</i>	2	Facilidad de implementarlo	3
<i>de</i>	3	Facilidad de adaptarlo	3
<i>la</i>			
<i>metodología</i>			
<i>Aspectos</i>	4	Organización de los integrantes	4
<i>de</i>	5	Comunicación de los integrantes	4
<i>la</i>			
<i>dinámica</i>	6	Motivación del equipo	4
<i>del</i>			
<i>equipo</i>			
<i>de</i>			
<i>trabajo</i>			
<i>Eficiencia</i>	7	Eficiencia general	3
<i>general</i>			

En la prueba de Friedman que se ejecutó para determinar si había diferencias entre las evaluaciones que los alumnos hicieron a los aspectos de la metodología de desarrollo de software con la que trabajaron, el conjunto de todas las evaluaciones recabadas para los aspectos estudiados resultó con diferencias estadísticamente significativas ( $\chi^2(6) = 52.456$ ,  $P_{\text{valor}} = 0.000$ ). De acuerdo a los rangos promedio (R.P.) que se muestran en la Tabla 3,

la mayor puntuación fue para la “organización de los integrantes”, mientras que la puntuación más baja fue para “facilidad de adaptar el enfoque metodológico”.

**Tabla**

**3.**

Rangos promedio obtenidos con la Prueba de Friedman.

<i>Categoría</i>	<i>Número</i>	<i>Descripción del aspecto sobre el</i>	<i>Rango</i>
	<i>de aspecto</i>	<i>enfoque metodológico</i>	<i>promedio</i>
<i>Aspectos</i>	1	Facilidad de entenderlo	3.20
<i>prácticos</i>	2	Facilidad de implementarlo	3.11
<i>de</i>	3	Facilidad de adaptarlo	2.95
<i>la</i>			
<i>metodología</i>			
<i>Aspectos</i>	4	Organización de los integrantes	5.02
<i>de la</i>	5	Comunicación de los integrantes	4.88
<i>dinámica</i>	6	Motivación del equipo	5.00
<i>del equipo</i>			
<i>de trabajo</i>			
<i>Eficiencia</i>	7	Eficiencia general	3.86
<i>general</i>			

También en la Tabla 2 y en la Tabla 3 puede observarse que las mayores puntuaciones fueron registradas en los aspectos relacionados con la dinámica del equipo de trabajo (organización (R.P. = 5.02), motivación (R.P. = 5.00), comunicación (R.P. = 4.88), todos

con mediana = 4) y las puntuaciones más bajas se obtuvieron en los aspectos prácticos propios de la metodología (facilidad de adaptarla (R.P. = 2.95), facilidad de implementarla (R.P. = 3.11), facilidad de entenderla (R.P. = 3.20), todos con mediana = 3).

Posteriormente, se realizaron comparaciones entre pares con la corrección de Bonferroni para comparaciones múltiples con el objetivo de determinar si las diferencias entre las puntuaciones de estas dos categorías eran significativas. Los análisis posteriores revelaron diferencias estadísticas significativas tal como se muestra en la Tabla 4.

**Tabla4.**

Comparaciones por pares entre “Aspectos prácticos de la metodología” y “Aspectos de la dinámica del equipo de trabajo” posteriores a la prueba de Friedman.

<i>Comparaciones por pares entre...</i>		<i>Resultados de las comparaciones por pares</i>		
<i>Aspectos prácticos de la metodología</i>	<i>Aspectos de la dinámica del equipo de trabajo</i>	<i>Xi<sup>2</sup></i>	<i>PValor ajustado (Bonferroni)</i>	<i>¿Es estadísticamente significativo? (95% conf.)</i>
	Comunicación entre los integrantes	-1.929	.018	Sí
<i>Facilidad de adaptarla</i>	Motivación del equipo	-2.054	.008	Sí
	Organización de los integrantes	-2.071	.007	Sí
	Comunicación entre los integrantes	-1.768	.046	Sí
<i>Facilidad de implementarla</i>	Motivación del equipo	-1.893	.022	Sí
	Organización de los integrantes	-1.911	.020	Sí
	Comunicación entre los integrantes	-1.679	.077	No
<i>Facilidad de entenderla</i>	Motivación del equipo	-1.804	.037	Sí
	Organización de los integrantes	-1.821	.034	Sí



## **Conclusiones**

En este artículo se presentaron los resultados de una investigación cuyo objetivo fue conocer más sobre las percepciones de los alumnos hacia diferentes aspectos relacionados con el enfoque metodológico de desarrollo de software denominado Proceso Unificado, el cual fue utilizado en un curso universitario de programación de aplicaciones.

Los resultados revelaron que de manera general, el enfoque metodológico fue bien evaluado por los estudiantes. El aspecto que obtuvo la mayor puntuación fue “Organización de los integrantes”. El que obtuvo la menor puntuación fue “Facilidad de adaptación”. Esto indica que los participantes sintieron que el Proceso Unificado les permitió trabajar de una manera sistematizada y con buena estructura. Sin embargo, su percepción respecto a la flexibilidad para realizar adecuaciones en la metodología fue significativamente menor.

Por otra parte, es importante resaltar que los aspectos relacionados con la dinámica del equipo de trabajo, tales como: comunicación, organización y motivación obtuvieron una puntuación mayor que los aspectos relacionados con el propio enfoque metodológico, tales como: facilidad de entenderlo, facilidad de implementarlo, facilidad de adaptarlo.

Los resultados permitieron determinar que la implementación de un enfoque metodológico como el Proceso Unificado durante un curso universitario es factible y que los alumnos evaluaron positivamente los distintos aspectos que se les plantearon acerca de él. Sin embargo, esta experiencia deja al descubierto una tendencia favorable de la dinámica de los participantes sobre la facilidad de trabajar con este enfoque metodológico.

Estos resultados proporcionan evidencias útiles para el proceso de toma de decisiones asociado a la implementación de un enfoque metodológico en el proceso de desarrollo de software en cursos universitarios.

## Referencias

- Jacobson, I., Booch, G., & Rumbaugh, J. (2005). *El Proceso Unificado de Desarrollo de Software*. Pearson Addison Wesley.
- Juristo, N., & M. Moreno, A. (2001). *Basics of Software Engineering Experimentation*. The Netherlands: Springer.
- Kuanli, Pavur, & Keeling. (2006). *Introduction to Business Statistics*. South-Western College Pub.
- Laudon, J., & Laudon, K. (2011). *Sistemas de información gerencial*. Ciudad de México: Pearson.
- Pressman, R. (2014). *Software Engineering: A Practitioner's approach* (6 ed.). United States: McGrawHill.
- Roque Hernández, R. V., Salinas Escandón, J. M., & López Mendoza, A. (2015). La complejidad de entender y enfrentar la formación de futuros desarrolladores de software. *Ciencia Ergo Sum*, 155-161.
- Rumbaugh, J., Jacobson, I., & Booch, G. (2000). *El Lenguaje Unificado de Modelado. Manual de Referencia*. Addison Wesley.
- Schach, S. (2005). *Análisis y diseño orientado a objetos con UML y el Proceso Unificado*. Ciudad de México: McGrawHill.
- Wagner, W. (2014). *Using IBM SPSS Statistics for Research Methods and Social Science Statistics*. SAGE Publications.