



ISSN: 2448-6574

EVALUACIÓN ESCOLAR A TRAVÉS DE PROYECTOS DE SOFTWARE EN UN CURSO DE PROGRAMACIÓN ORIENTADA A OBJETOS (POO)

Víctor Garduño Mendieta¹

Mario Alberto Sesma Martínez²

Mario Oviedo Galdeano³

RESUMEN

En los procesos de enseñanza y aprendizaje es muy importante la evaluación del estudiante para asentar su calificación en su kardex. Dicha evaluación debe ser justa y objetiva para que sea verdadera. En grupos numerosos, de 30 o más estudiantes, resulta más difícil realizarla. En cursos de programación se puede hacer orientada a proyectos de software que impliquen el análisis del problema a resolver y el diseño y desarrollo de la solución. El aprendizaje resulta muy provechoso si se emplean estrategias didácticas orientadas al aprendizaje colaborativo en pequeños grupos de trabajo realizando un proyecto de diseño y desarrollo de software.

En este trabajo se presenta el caso de evaluación aplicando dicha metodología a grupos del curso de Programación Orientada a Objetos de la Unidad Profesional Interdisciplinaria de Ingeniería y Ciencias Sociales y Administrativas del Instituto Politécnico Nacional. También se pretende contribuir a corregir una carencia importante que ocurre frecuentemente en los cursos de programación sobre un marco metodológico adecuado y coherente para la realización de los trabajos finales en dichas unidades de aprendizaje.

¹ vicgardm@yahoo.com.mx, IPN-UPIICSA

² masesma@hotmail.com, IPN-UPIICSA

³ mog974@yahoo.com.mx, IPN-UPIICSA



ISSN: 2448-6574

PALABRAS CLAVE

Aprendizaje colaborativo, evaluación por proyectos, proceso enseñanza/aprendizaje, grupos numerosos, proyectos de software, programación orientada a objetos, análisis, diseño, desarrollo de software, estrategia didáctica, profesor, estudiante.

EVALUACIÓN ESCOLAR A TRAVES DE PROYECTOS DE SOFTWARE EN POO

Planteamiento del problema

En grupos numerosos, de treinta alumnos o más, se dificulta la evaluación individual de su desempeño y por lo tanto pudiera perderse cierta objetividad y visión del desempeño real del estudiante. Además, el problema se agrava si consideramos que en general el profesor tiene que cumplir con los calendarios escolares y prefiere aplicar exámenes generales, tal vez con variantes en los datos de un mismo problema o aplicar exámenes diferentes y evitar hasta donde es posible la copia de respuestas.

Justificación

En el proceso enseñanza aprendizaje la evaluación permite saber si se lograron los objetivos del curso valorando el alcance de los conocimientos y habilidades adquiridos por los estudiantes. Por lo que es de suma importancia que sea real y objetiva, nada fácil de lograr sobre todo en grupos numerosos.

Objetivo

Mejorar las técnicas de evaluación de grupos numerosos mediante aprendizaje colaborativo y proyectos de diseño y desarrollo de software.

Presentación

En el caso específico de los cursos de programación de la UPIICSA gran parte de nuestros estudiantes aprenden a manejar las herramientas de desarrollo de software pero la mayoría tiene serias dificultades para adquirir la habilidad del diseño de la lógica de programación.

Algunas de las materias de programación de nuestras carreras son: Lógica de Programación, Fundamentos de Programación Orientada a Objetos y Programación Orientada a Objetos. En el primer curso se pretende que el estudiante aprenda las bases de la programación estructurada y



ISSN: 2448-6574

comprenda la importancia del diseño de la lógica para resolver un problema de baja a mediana complejidad. Para la implementación se utiliza el lenguaje C. En el siguiente curso se fundamenta el paradigma de la programación orientada a objetos y se reafirma en el tercer curso. Para la implementación se utiliza C++ en el segundo curso y Java en el tercero.

Para el proceso de evaluación se pueden aplicar varias técnicas en función de la estrategia didáctica que se diseñe e implemente. Una de ellas puede ser evaluación por proyectos de software. El curso de Programación Orientada a Objetos (POO) es un espacio adecuado para aplicar la estrategia didáctica del aprendizaje colaborativo, en grupos de trabajo con cinco integrantes como máximo, a través de proyectos de diseño y desarrollo de software. Para una propuesta de una mejor evaluación hemos seleccionado esta asignatura que se cursa en dos de nuestras carreras.

Estrategia didáctica

La evaluación con proyectos de software tiene varios propósitos. Uno de ellos es que el estudiante construya el conocimiento con base en las experiencias adquiridas al diseñar y desarrollar aplicaciones de software que ofrezcan las respuestas correctas y satisfactorias a los eventos generados por el usuario final. Otro es que el estudiante desarrolle sus habilidades para trabajar en equipo con otros compañeros en calidad de pares y también aprenda a dirigir y coordinar grupos de trabajo en calidad de líder de proyecto. Se pretende además que el estudiante aprenda a diferenciar entre un trabajo participativo de un trabajo colaborativo y fomentar este último y lograr un efecto multiplicador del profesor en su exposición de los conceptos y contenidos. Uno más es que el estudiante trabaje en proyectos medianamente complejos en plazos breves y medianos, como son los periodos bimestrales y semestrales. El proceso de evaluación con esta técnica implica un monitoreo y guía del profesor durante el diseño y desarrollo del proyecto.

Como primera actividad el profesor debe formar los grupos de trabajo. En el curso de POO se pidió a los estudiantes que ellos mismos formaran dichos equipos de trabajo. En seguida debe presentarse a los grupos el documento de requerimientos y la rúbrica que les indica los aspectos que se evaluarán y los niveles de calificación respectiva. Por otra parte se establece la mecánica de trabajo y el contexto del objeto de estudio.



ISSN: 2448-6574

Una vez formados los grupos de trabajo y definido el problema a resolver, se procederá a la asignación de tareas y la designación de funciones dentro del grupo, tales como el líder de proyecto, responsable de la documentación y elaboración del informe final, entre otros. El profesor pedirá a cada grupo que haga la planeación y el programa de actividades con base en la fecha de entrega establecida.

FASES DEL PROYECTO

Análisis. El contexto del objeto de estudio y el documento de requerimientos serán la base del análisis. Es necesario y conveniente realizar una investigación documental y, de ser posible, también de campo para complementar la información del análisis. En esta fase se pueden aplicar dos estrategias:

- a) Requerimientos cerrados especificados por el docente.
- b) Requerimientos abiertos establecidos por los estudiantes

En ambos casos, antes de establecerse dichos requerimientos debe tenerse un campo de aplicación contextualizado, un usuario real que los valide y una delimitación del proyecto.

Diseño. En esta fase del proyecto el profesor deberá trabajar muy estrechamente con el equipo para lograr un buen resultado al final del proyecto. El primer paso es diseñar la clase que resuelva el problema. Además de la clase, los diagramas de casos de uso facilitarán el diseño de los métodos que definen el comportamiento del objeto. Esta fase es de primordial importancia y el estudiante deberá comprenderlo cabalmente.

El diseño del comportamiento del objeto se hace a través de la lógica de programación de sus métodos. En el curso de POO se utilizó el pseudocódigo propuesto por Guillermo Levine en su libro⁴. El profesor deberá revisar que la lógica corresponda al comportamiento esperado y que se le haga la prueba de escritorio correspondiente.

Desarrollo

En el curso de POO se emplea el lenguaje Java para el desarrollo de la programación y como herramienta para crear las interfaces gráficas se utiliza Netbeans. El profesor deberá supervisar

⁴Levine, G.. (1989). *Introducción a la Computación y la Programación Estructurada*. México D.F.: Mc Graw Hill.



ISSN: 2448-6574

cuidadosamente y con la frecuencia conveniente el avance del proceso. En general se establece un modelo básico sugerido para la interfaz del usuario y se deja en libertad a los grupos de trabajo para su diseño final y solamente se les pide que se ajusten a la funcionalidad especificada en el documento de requerimientos. Además, la aplicación deberá manejar los errores a través de la estructura *try-catch* y las clases de excepción pertinentes. Se pretende lograr una aplicación robusta con una interfaz gráfica de usuario (GUI) amigable y efectiva que produzca software de alta calidad.

Organización de los equipos de trabajo

Organizar un grupo de 30 o más estudiantes en grupos de trabajo de máximo cinco miembros no es tarea fácil ni trivial. Debemos buscar cierto equilibrio entre sus componentes, propiciar la empatía y la tolerancia para que el debate de sus propuestas y finalmente toma de decisiones sea en un marco de respeto y armonía. Es muy importante la designación y reconocimiento de un líder de proyecto que, en general se convierte en el del grupo. También es muy importante la designación de roles y la asignación de tareas y funciones dentro del grupo para lograr el grado máximo de participación de sus integrantes.

“Los grupos base son grupos de largo alcance, o de largo plazo, pueden durar hasta un año, son grupos heterogéneos con una membresía estable cuyo propósito principal es el de brindar a cada uno de sus miembros apoyo, ayuda y asistencia en cada una de las necesidades que surjan mientras se lleva a cabo una tarea que llevará al progreso académico”⁵.

Monitoreo del trabajo colaborativo (implica el trabajo participativo)

En el proceso de diseño y desarrollo del proyecto, el profesor además del guía y asesor, debe tomar el rol del usuario final y actuar como el cliente que demanda la solución de sus requerimientos y necesidades. La interacción con el grupo es necesaria y le permite al profesor evaluar de manera continua el desarrollo individual de los alumnos.

⁵ Programa de Desarrollo de Habilidades Docentes. (2007). *Programa de Desarrollo de Habilidades Docentes. Versión 2007*. mayo 20, 2016, de Instituto Tecnológico y de Estudios Superiores de Monterrey Sitio web: <http://sitios.itesm.mx/va/dide2/documentos/pdhd2007.pdf>



ISSN: 2448-6574

El profesor deberá inducir al grupo a que realicen reuniones presenciales para analizar, debatir propuestas y que aprendan a elegir las mejores. Los grupos de trabajo deberán elaborar un resumen de la reunión a la que llamamos minuta de la reunión. Los estudiantes deben entender con claridad que cumplir con su responsabilidad individual no los releva de su responsabilidad grupal para que se logren los objetivos principales del ejercicio. Por lo tanto cada elemento del grupo deberá comprender las tareas asignadas a los demás integrantes, esto les da la posibilidad de hacer la autoevaluación del trabajo grupal y los avances conseguidos, así podrán supervisar el trabajo de los demás miembros y ofrecerles su apoyo en caso necesario. El estudiante debe desarrollar el sentido de pertenencia del grupo, de tal manera que perciba los éxitos y fracasos del equipo como propios.

Documento de requerimientos

Los requerimientos de un sistema o aplicación son el conjunto de requisitos que debe cumplir para que sea útil al usuario final para resolver sus problemas en el manejo de su información y toma de decisiones derivadas de ella.

¿Qué es un requisito?

- Condición o capacidad que un usuario necesita para poder resolver un problema o lograr un objetivo (IEEE).
- Condición o capacidad que debe exhibir o poseer un sistema para satisfacer un contrato, estándar, especificación, u otra documentación formalmente impuesta (IEEE).
- Una condición o capacidad que debe ser conformada por el sistema (RUP).
- Algo que el sistema debe hacer o una cualidad que el sistema debe poseer (Robertson - Robertson).

Requisito Funcional

Un requisito funcional define una función del sistema de software o sus componentes. Una función es descrita como un conjunto de entradas, comportamientos y salidas. Los requisitos funcionales pueden ser: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que se supone, un sistema debe cumplir. Los requerimientos de comportamiento para cada requerimiento funcional se muestran en los casos de uso. Son complementados por los requisitos no funcionales, que se enfocan en cambio en el diseño o la implementación.

Como se define en la ingeniería de requisitos, los requisitos funcionales establecen los comportamientos del sistema.

Típicamente, un analista de requisitos genera requisitos funcionales después de realizar los casos de uso. Sin embargo, esto puede tener excepciones, ya que el desarrollo de software es un proceso iterativo y algunos requisitos son previos al diseño de los casos de uso. Ambos elementos (casos de uso y requisitos) se complementan en un proceso bidireccional.

Un requisito funcional típico contiene un nombre y un número de serie único y un resumen. Esta información se utiliza para ayudar al lector a entender por qué el requisito es necesario, y para seguir al mismo durante el desarrollo del producto.

El núcleo del requisito es la descripción del comportamiento requerido, que debe ser clara y concisa. Este comportamiento puede provenir de reglas organizacionales o del negocio, o ser descubiertas por interacción con usuarios, inversores y otros expertos en la organización.

Un requisito no funcional o atributo de calidad es, en la ingeniería de sistemas y la ingeniería de software, un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que éstos corresponden a los requisitos funcionales. Por tanto, se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar, sino características de funcionamiento⁶.

Algunas características del software para establecer requisitos no funcionales son las siguientes:

- rendimiento
- disponibilidad
- accesibilidad
- usabilidad
- estabilidad
- portabilidad

⁶ https://es.wikipedia.org/wiki/Requisito_%28sistemas%29



ISSN: 2448-6574

- costo
- operatividad
- interoperabilidad
- escalabilidad
- concurrencia
- mantenibilidad
- interfaz
- seguridad

Rúbrica

La rúbrica es un instrumento que junto con el documento de requerimientos guían al alumno para realizar el proyecto según las expectativas que se tienen para su aprendizaje y práctica y hacer una evaluación objetiva y justa de su trabajo. Al equipo de trabajo le proporciona la posibilidad de hacer propuestas sobre la aplicación y debatir sobre las mismas para elegir e implementar las mejores de acuerdo con su propio criterio y coadyuva a su desarrollo.

“Las rúbricas son guías precisas que valoran los aprendizajes y productos realizados. Son tablas que desglosan los niveles de desempeño de los estudiantes en un aspecto determinado, con criterios específicos sobre rendimiento. Indican el logro de los objetivos curriculares y las expectativas de los docentes”⁷.

Calificación individual

Considerando que fuera necesaria una calificación individual diferenciada, los integrantes de un mismo equipo obtendrían una calificación como grupo de trabajo y se sumaría a cada miembro la parte que obtuviera por méritos individuales bajo ciertos criterios establecidos al inicio del periodo.

⁷ Gatica-Lara, F. & Uribarren-Berrueta, T. (2013). ¿Cómo elaborar una rúbrica?. mayo 5, 2016, de Elsevier Sitio web: <http://www.elsevier.es/es-revista-investigacion-educacion-medica-343-articulo-como-elaborar-una-rubrica-90219696>



ISSN: 2448-6574

Para el trabajo colaborativo son indispensables las reuniones presenciales, en ellas los miembros del grupo analizan los problemas que tienen que resolver, hacen propuestas y las analizan, debaten y toman acuerdos. Además en las reuniones los miembros del grupo se asignan las tareas y se organizan para desarrollar los trabajos cuyos resultados se presentarán en una reunión posterior. Es importante que se levante una minuta o resumen de la reunión, estas minutas se presentarán al profesor como evidencias de las reuniones y podrían tener un valor curricular y sumarse a la calificación individual.

Resultados y conclusiones

Los resultados que hemos obtenido al aplicar la estrategia didáctica con evaluación por proyectos de software, indican que debemos trabajar más e innovar en las técnicas de enseñanza y práctica para que el alumno comprenda mejor la importancia y ventajas del diseño. Sin un marco metodológico sencillo y consistente se pueden tener resultados de buen aspecto y visualmente convincentes pero comúnmente alejados de satisfacer las necesidades reales de un usuario y de la calidad que debe tener el software útil por modesto que éste sea. Es necesario demostrarle al estudiante que un buen diseño es el principio de un buen producto de software con calidad y robustez.