

Aprendizaje basado en problemas y en comparación de soluciones en un contexto semipresencial

Ramón A. Mollineda Cárdenas

Departamento de Lenguajes y Sistemas Informáticos

Universidad Jaume I

Av. de Vicent Sos Baynat, s/n. 12071 Castellón de la Plana

mollined@lsi.uji.es

Resumen

Este trabajo propone un *método de evaluación* continua, no presencial e implementable a través de entornos de aprendizaje en línea. En una primera fase, el profesor propone un problema y define una fecha de entrega, antes de la cual los alumnos deberán enviar sus soluciones (resultado *A*). Cumplido este plazo, el profesor publica una solución correcta. En una segunda fase, los alumnos comparan sus soluciones con la del profesor, y elaboran un informe de autoevaluación (resultado *B*). De la entrega *A* se evalúa corrección y optimalidad, mientras que de *B* se valora la capacidad del alumno para realizar un análisis crítico de los errores y singularidades de su solución. Como evaluación global del ejercicio se propone una media ponderada de las notas de ambos resultados, *A* y *B*.

Summary

This paper proposes a method of continuous assessment that can be implemented through online learning environments. In a first phase, the teacher poses a problem and sets a deadline for students to submit their individual solutions (outcome *A*), after which a correct solution is published. In a second phase, students will compare their solutions with that of the teacher, and write a self-assessment report (outcome *B*). The outcome *A* is evaluated based on its correctness and optimality, while *B* is assessed considering students' ability to detect their errors and to make a reasoned defense of their solutions. The overall evaluation is intended as a weighted average of the notes of both deliverables, *A* and *B*.

Palabras clave

Autoevaluación, semipresencialidad, aprendizaje basado en problemas.

1. Introducción

La convergencia del sistema universitario español al EEES implica retos notables, pero es también una oportunidad para reflexionar sobre cómo debe cambiar la universidad española para crear titulados capaces de integrarse en un mercado laboral cada vez más tecnológico y dinámico.

Una de las claves del cambio será la redefinición de los roles de los agentes del proceso de enseñanza-aprendizaje. El nuevo escenario requerirá más profesores facilitadores que transmisores de conocimientos, y alumnos conscientes de que el aprendizaje real se alcanza fuera del aula. Estos nuevos perfiles son particularmente necesarios en la formación no presencial pues, al no existir docencia tradicional, el alumno debe alcanzar regularmente objetivos de aprendizaje de forma autónoma. Otro de los pilares del sistema deberá ser el uso adecuado de las TICs, las cuales proporcionan disponibilidad, inmediatez, realimentación y canales de comunicación.

En [1] se propone un ejercicio docente, afín a estos principios, de trabajo grupal no presencial, en el que equipos de alumnos envían entregas dentro de ciertos plazos a través de un entorno de aprendizaje en línea (aulas virtuales), y a los que el profesor responde con sus valoraciones. Sin embargo, esta práctica no es concebida como ejercicio de evaluación docente.

En este artículo presenta un *método de evaluación continua*, de trabajo individual no presencial, e implementable a través de un aula virtual. Esta práctica fue concebida para una asignatura troncal de introducción a la programación en el contexto de una titulación semipresencial. En nuestra propuesta, a diferencia de [1], el alumno deberá autoevaluar su solución mediante comparación con la solución del profesor, antes de

que este lo haga. Además, el profesor evaluará tanto el primer ejercicio como la propia autoevaluación, otorgando finalmente una nota conjunta.

2. Contexto de aplicación

Esta sección presenta los niveles contextuales en los que se inserta el método propuesto: institucional, TIC, asignatura y sistema de evaluación.

2.1 Contexto institucional

La Universitat Jaume I (UJI) puso en marcha en 2007 el Programa UJI Virtual que, entre otras acciones, proponía implantar titulaciones semipresenciales como proyectos pilotos de armonización. Una de las titulaciones pioneras en adoptar este nuevo formato fue Ingeniería Técnica en Informática de Gestión (ITIG, plan de 2001). Los principios que guiaron el diseño de los estudios semipresenciales en la UJI fueron:

- reducción drástica de horas de clases (-70%)
- creación de grupos reducidos (≤ 40 alumnos)
- uso de entornos de aprendizaje en línea
- implantación de tutorías electrónicas
- atención compatible con horarios laborales

La menor cantidad de horas lectivas debía compensarse con metodologías docentes basadas en aprendizaje autónomo, sistemas de evaluación continua y una supervisión constante y efectiva.

2.2 Contexto TIC

En 2004, la Universitat Jaume I acordó adoptar Moodle como entorno virtual de enseñanza-aprendizaje [2]. Moodle es una plataforma web que permite crear espacios virtuales de trabajo en torno a recursos docentes, por ejemplo, materiales elaborados por un profesor, tareas enviadas por alumnos a través de la web, exámenes, encuestas, foros, entre otras posibilidades. Este tipo de herramientas se convierten, por tanto, en soportes necesarios para el uso de metodologías de aprendizaje activo, entre ellas, las que requieren trabajo autónomo y colaborativo.

2.3 Contexto de asignatura semipresencial

El método de evaluación que aquí se presenta se diseñó para la *versión semipresencial* de la asignatura troncal Metodología y Tecnología de la Programación (MTP), la cual se impartía en el primer curso de la titulación ITIG de la UJI. Esta asignatura consistía en la introducción de

conceptos básicos de programación, y significaba el primer contacto de muchos de los alumnos con la lógica algorítmica. Sus clases se impartían en una única sesión semanal de 2 h, en la que se alternaban sesiones teóricas y prácticas.

2.3.1 Sistema de evaluación

El sistema de evaluación de la asignatura MTP semipresencial constaba de las siguientes 4 actividades evaluables:

1. Controles periódicos de mínimos [10%]. Esta actividad tenía lugar al inicio de cada práctica durante 10 minutos. En ella se evaluaban contenidos teóricos y prácticos muy básicos, con el objetivo de comprobar si el alumno tenía la preparación mínima necesaria para la práctica. Esta actividad se suponía premiaba la asistencia a prácticas, aunque, por su reducido peso en la nota, no constituía una penalización significativa en caso de ausencia.
2. Defensa de las prácticas [10%]. Cada práctica consistía en la resolución de un boletín de ejercicios con grado creciente de dificultad, y su evaluación, en la defensa presencial en tutorías de las soluciones propuestas.
3. Pares de entregas periódicas [20%]. Esta actividad es la propuesta en este artículo, y sus detalles aparecen en la siguiente sección. El sistema de evaluación incluía una actividad de este tipo por cada tema, y su ejecución era totalmente *no* presencial a través del Aula Virtual. El ejercicio tenía dos partes. En una primera, el alumno enviaba su solución a un problema propuesto antes de una fecha límite, después de la cual el profesor publicaba su solución. En la segunda parte, el alumno debía elaborar (y enviar dentro de un segundo plazo) un informe crítico sobre su solución a partir de la comparación con la solución del profesor. La nota global del ejercicio se calculaba como una media ponderada de las notas de ambas partes. Esta actividad aportaba hasta 2 puntos de los 10 de la nota final.
4. Examen final escrito [60%]. La cuarta actividad era un examen final escrito. En él se planteaban problemas que los alumnos debían resolver sobre papel, lo cual exige una mayor seguridad a la hora de concebir soluciones. Durante su realización, los alumnos podían consultar todo tipo de materiales y apuntes.

3. Método

El método de evaluación que se propone consta de 2 fases, y en cada una el alumno deberá presentar un resultado evaluable a través del Aula Virtual.

En la primera fase, que podríamos denominar *solución de problemas*, el profesor propone un problema y define un plazo de entrega para que los alumnos envíen sus soluciones individuales (resultado *A*). Una vez cumplida la fecha de entrega, el profesor publica una solución correcta.

Para que esta actividad de evaluación mantenga su efectividad en un contexto *no* presencial, el problema debería ser lo suficientemente singular para que sea improbable encontrar soluciones exactas ya escritas y, por otro, suficientemente abierto para que no sea aceptable justificar 2 respuestas muy parecidas. Estos requisitos, *a priori* difíciles de lograr, son asequibles en asignaturas de programación, pues es frecuente encontrar problemas del mundo real con soluciones no estándares y abiertas.

En una segunda fase, a la que llamaremos *autoevaluación de la solución*, cada alumno debe comparar *su* solución con la publicada por el profesor, y enviar un informe de autoevaluación (resultado *B*) antes de una segunda fecha límite. Este informe deberá incluir una discusión de las diferencias entre las soluciones individual y la del profesor, el reconocimiento de errores propios, la defensa de las decisiones adoptadas, etc.

La evaluación global del ejercicio se propone como una media ponderada de las notas de ambos resultados, *A* y *B*. Del resultado *A* se evalúa corrección, optimalidad, orden, estética, y buenas prácticas en general. En el resultado *B* se valora la capacidad del alumno para detectar sus errores y para realizar una defensa fundamentada de las decisiones tomadas. Ambas partes son evaluadas independientemente a través del Aula Virtual, una vez se ha cumplido el segundo plazo. Además de la posibilidad de aprender de errores propios, esta segunda parte ofrece la oportunidad de mejorar la nota global del ejercicio en caso de realizar un análisis adecuado de la primera entrega, lo cual se supone un elemento motivador. El resultado *B* pierde sentido si el resultado *A* es insuficiente o no se entrega. En este caso, se propone evaluar el ejercicio como suspenso.

En nuestra implementación, los coeficientes de ponderación usados en el cálculo de la nota global fueron 0,6 y 0,4. Es decir, si *nA* y *nB* son

las notas de los resultados *A* y *B*, la nota *n* del ejercicio se calcula como sigue:

$$n = 0,6 nA + 0,4 nB$$

3.1 Ejemplos de problemas

A continuación se presentan 2 ejemplos de problemas reales propuestos a los alumnos, que se corresponden con niveles de aprendizaje y lenguajes de programación distintos. Ambos intentan cumplir los requisitos deseables en los problemas: singular y abierto. El primero de ellos se propuso al terminar el primer tema del curso, una vez habían sido introducidas estructuras *condicionales* básicas de programación en Python. Se trataba de un escenario de toma de decisiones al conducir vehículos en una ciudad:

Escribe un algoritmo en lenguaje natural (puedes usar instrucciones Python) que imprima la decisión que deberá tomar un conductor en una intersección de 2 vías (detenerse o continuar) en un trazado urbano. Deberás tener en cuenta la mayor cantidad de situaciones posibles de señalización en ambas vías, desde escenarios sin señales en tu sentido de conducción, hasta la existencia de señales explícitas como Stop, semáforos, Ceda el paso, u otras, en cualquiera de las vías.

El segundo ejemplo es un problema propuesto a mitad del segundo semestre, cuando los alumnos conocían estructuras de Python y C. A continuación se presenta un resumen:

Escribe un programa educativo para enseñar a niñ@s pequeñ@s a conjugar verbos regulares y a establecer concordancias de género y número. Dados un pronombre personal, un sustantivo (entre envase, botella y pila), y un número entre singular y plural, el programa pedirá al usuario una oración simple con el verbo reciclar en presente del indicativo según la siguiente estructura,

[Pronombre] [verbo reciclar conjugado] [artículo] [sustantivo].

El programa validará la corrección de la oración y, si el resultado es válido, imprimirá un mensaje indicándolo. En caso de error, deberá ofrecer otra oportunidad al usuario. Si el usuario

decide terminar sin acertar, el programa imprimirá la oración correcta.

4. Discusión

Según [3], “la *competencia* es la capacidad para enfrentarse con garantías de éxito a una tarea en un contexto determinado”. Tomando como referencia esta definición, en esta sección se discuten competencias que el método propuesto podría contribuir a fomentar.

Para simplificar el análisis, usaremos parte de la clasificación de competencias propuesta en [3]:

- **Epistémicas:** capacidades de comprensión, elaboración y utilización de la información .
- **Reflexión y autoaprendizaje:** capacidades de análisis y síntesis, resolución de problemas, trabajo autónomo, motivación de logro, etc.
- **Comunicativas:** capacidades de expresión oral, escrita, uso de nuevas tecnologías, etc.

En relación a las competencias *Epistémicas*, el método propuesto exige al alumno el uso de herramientas de programación conocidas para resolver situaciones nuevas. Por tanto, el alumno deberá demostrar estar en posesión de un mínimo aceptable de competencias cognitivas, lógicas e instrumentales.

De las tres categorías revisadas, probablemente la que tiene mayor presencia en nuestro método es la de *reflexión y autoaprendizaje*. En la segunda fase del método, el alumno debe comparar su solución con la del profesor, detectar errores, y finalmente elaborar un informe con sus conclusiones. La práctica de aprender a partir de ejemplos bien resueltos (solución del profesor) es, sin dudas, una de las más populares y efectivas en programación. Por lo tanto, con este ejercicio el alumno estaría incorporando a su forma de hacer un buen método de aprendizaje, es decir, estaría aprendiendo a aprender.

Por último, en relación a las competencias *comunicativas*, la redacción del informe de autoevaluación requiere explicar, de forma escrita, ideas lógicas de relativa complejidad para alumnos que se inician en la programación. Este ejercicio contribuye, por un lado, a desarrollar las habilidades de comunicación y el espíritu crítico, mientras que por otro, refuerza el aprendizaje.

Conclusiones

Este artículo presenta un método de evaluación continua en el contexto de una asignatura semipresencial. El método se compone de dos partes. La primera consiste en resolución de problemas, mientras que la segunda, en la comparación de la solución propia con una solución publicada por el profesor. Todo el proceso se realiza a través de un entorno de aprendizaje en línea, lo cual habilita el método para escenarios no presenciales. Ambas partes dan lugar a sendos informes que los alumnos deben enviar dentro de plazos habilitados, y que el profesor evalúa por separado antes integrar ambas calificaciones en un nota única.

Además, el artículo incluye reflexiones sobre las competencias que se fomentan con el ejercicio docente descrito. Entre las más importantes podríamos destacar el uso de conocimientos en la solución de problemas nuevos, el análisis crítico de trabajos propios, el aprendizaje a partir de errores y de ejemplos correctos, el reconocimiento de lo singular y la comunicación escrita.

Aunque no se realizó un estudio estadístico sobre la incidencia de este método en el progreso de los alumnos, si se observó una elevada tasa de éxito entre aquellos que hicieron un seguimiento regular de esta actividad.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Vicerrectorado de Ordenación Académica y Profesorado de la Universitat Jaume I de Castelló.

Referencias

- [1] Jiménez-González, D. y colaboradores. Trabajo no presencial en colaboración: triple realimentación a coste razonable. En Actas del XIV JENUI, p. 275-282, 2008.
- [2] Adell, J. y colaboradores. *Selección de un entorno virtual de enseñanza/aprendizaje de código fuente abierto para la UJI*. Centre d'Educació i Noves Tecnologies, UJI, http://cent.uji.es/doc/eveauji_es.pdf, 2004.
- [3] Goñi Zabala, Jesús. *El Espacio Europeo de Educación Superior, un reto para la universidad*. Ed. Octaedro, 2005.