

La autoevaluación con la plataforma web SIETTE como estrategia de aprendizaje

Lawrence Mandow Andaluz^a, José del Campo Ávila^a, Nathalie Moreno Vergara^a, and José A. Onieva^a

^aDepartamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga

ABSTRACT

La autoevaluación es una estrategia que ayuda al alumno a tomar conciencia de su progreso de aprendizaje y facilita al docente la comprensión sobre el proceso de enseñanza y aprendizaje realizado por el discente, en relación por ejemplo con las dificultades acontecidas o los objetivos perseguidos. En este proyecto hemos tratado de potenciar las ventajas de la autoevaluación incorporando la posibilidad de que éstas sean percibidas por los alumnos de forma inmediata. Con ello hemos conseguido una mayor implicación por parte del alumnado, al tiempo que se han podido detectar acciones concretas a tener en cuenta.

Keywords: Autoaprendizaje, Evaluación formativa, Automatización de tareas, SIETTE.

1. INTRODUCCIÓN

Existen metodologías para el proceso de enseñanza-aprendizaje que son utilizadas de forma minoritaria a pesar de las ventajas que ofrecen. Es el caso de la autoevaluación, que es el paradigma vertebrador de este trabajo. Entre sus ventajas más interesantes podemos citar la motivación que supone para los alumnos, puesto que les permite ser conscientes, de manera progresiva, de su evolución en el proceso de aprendizaje. Pero también es una herramienta que se puede adaptar a la diversidad del alumnado y propiciar diferentes ritmos y niveles de aprendizaje.

Los motivos para que su uso no sea más extenso pueden ser múltiples: desconocimiento de los beneficios que puede reportar o escaso interés en el alumnado si, configurada inadecuadamente, no se recibe información inmediata sobre su autoevaluación por citar algunas. Es importante incidir en que esa información que esperan los alumnos (a corto plazo), y que sólo puede generar el profesor, es necesaria porque los alumnos no habrán desarrollado aún las competencias necesarias previstas en su formación y tampoco tendrán las capacidades de evaluación propias de un docente; y por extensión, tampoco con sus pares.

En este trabajo pretendemos potenciar las ventajas de la autoevaluación incorporando la posibilidad de que el aprendizaje sea percibido por los alumnos de forma inmediata. Entre nuestros objetivos, por tanto, se encuentran conseguir una mayor implicación por parte del alumnado, al tiempo que se puedan detectar acciones concretas de refuerzo. Para ello, las actividades que hemos diseñado incluyen la autoevaluación como característica fundamental y presentan la particularidad de que se ofrecerá información inmediata (y reutilizable) preparada por los docentes. Se ha implementado mediante la utilización del sistema web SIETTE,¹ que está disponible de forma global y que además está incorporado de forma nativa en el campus virtual Moodle de la Universidad de Málaga.

Este artículo está organizado de la siguiente forma: en la sección 2 mostramos a la herramienta utilizada, así como sus capacidades técnicas. En la sección 3 establecemos el contexto de aplicación de la experiencia que hemos desarrollado para a continuación detallar los distintos tipos de items utilizados en las secciones 4 y 5. En las secciones 6 y 7 aparecen una evaluación de la experiencia llevada a cabo por los docentes así como la extracción de las conclusiones de dicha experiencia a partir de los datos recogidos del alumnado acerca de su participación en la misma.

Further author information: (Send correspondence to ...)

2. LA HERRAMIENTA SIETTE

El uso de sistemas educativos basados en plataformas web está cada vez más extendido, gracias a las ventajas que ofrecen.^{2,3} SIETTE es uno de esos sistemas web que contempla una amplia variedad de funcionalidades, pudiendo ser consideradas algunas de ellas como funcionalidades de vanguardia.

SIETTE* permite la creación y mantenimiento de bancos de preguntas y la evaluación automática mediante distintas estrategias. Proporciona control de acceso a las evaluaciones, así como sobre la selección de preguntas y los criterios de finalización. Incluye distintos tipos de preguntas, desde preguntas de múltiple opción a preguntas abiertas, preguntas de contenido matemático y otras definidas por extensiones del usuario. Es capaz de evaluar la corrección de objetos complejos, como programas de ordenador, o tareas complejas que requieren interacciones con el usuario, tales como dibujos o sonidos musicales. También incluye la generación automática de preguntas mediante plantillas extrayendo los contenidos de hojas de cálculo, bases de datos o de la web semántica. Implementa la Teoría Clásica de Test (TCT), la Teoría de Respuesta al Item (TRI), y permite realizar Tests Adaptativos Informatizados (TAI).

El sistema integra herramientas para análisis de resultados e indicadores psicométricos de la TCT y la TRI, así como enlaces a programas de calibración (Multilog), herramientas de visualización de datos (Ingrid), programas de control de plagio (MOSS) y mecanismos de gamificación (en desarrollo). Puede usarse como herramienta para el Aprendizaje colaborativo o como módulo de evaluación de un Sistema Tutor Inteligente (STI) y puede conectarse a una Plataforma de TeleEducación (LMS) como Moodle (de hecho, en el contexto de este artículo se encuentra conectada).

3. CONTEXTO

Este trabajo se enmarca en el contexto de un proyecto de innovación educativa realizado en la Universidad de Málaga, durante el bienio 2017-2019, en el que se implicaron nueve profesores y un total de once asignaturas de diferentes Grados. En algunas de esas once asignaturas se evaluó la idoneidad y beneficios de utilizar la plataforma SIETTE para el uso de diferentes pruebas de autoevaluación con corrección instantánea como elemento de aprendizaje. La tabla 1 resume las características de algunas de las asignaturas y titulaciones en las que se llevó a cabo la experiencia con SIETTE.

Titulación	Asignatura	Créditos	Curso	Semestre	Tipo	Alumnos
Grado en Ingeniería Informática y en Ingeniería de Computadores	Sistemas Inteligentes	6	1	2	Obligatoria	2 grupos y 100 alumnos aprox.
Grado en Ingeniería en Eléctrica, y dobles grados con Ingeniería Mecánica e Ingeniería Electrónica	Fundamentos de Informática	6	1	1	Obligatoria	1 grupo y 60 alumnos aprox.
Grado en Ingeniería del Software, en Ingeniería Informática e Ingeniería de Computadores	Análisis y Diseño de Algoritmos	6	2	1	Obligatoria	3 grupos y 150 alumnos aprox.
Grado en Gestión y Administración Pública	Informática Aplicada a la Gestión Pública	6	1	2	Obligatoria	2 grupos y 100 alumnos aprox.

Table 1: Titulaciones y asignaturas implicadas en la experiencia.

En las siguientes secciones se contextualiza el contenido de cada asignatura en relación con las herramientas utilizadas.

*<http://www.siette.org/siette>

4. TESTS DE FUNCIONALIDAD

Los tests de funcionalidad son una herramienta de SIETTE que permite la corrección de pruebas de forma programática. Un test en SIETTE puede estar formado por varias preguntas. Los test de funcionalidad se caracterizan por estar formados por preguntas que, en su totalidad o en su mayoría, tienen como objetivo la corrección de un programa informático entregado por un alumno en cualquier lenguaje de programación.

Estas preguntas siguen un principio de diseño básico: compilar, ejecutar y medir la calidad de los ficheros proporcionados por los alumnos de manera que, tras una batería de pruebas previamente diseñada por el profesor, se pueda emitir una calificación y unos consejos. Estos permitirán al alumno identificar qué prueba ha fallado, de forma que pueda acotar la funcionalidad que ha de revisar en su entrega antes de volver a intentarlo (si el test permite la repetición de intentos).

En particular, estos test de funcionalidad se han implementado y probado en dos asignaturas impartidas en diferentes titulaciones: Análisis y Diseño de Algoritmos y Fundamentos de Informática. En el caso de Análisis y Diseño de Algoritmos los test tienen como objetivo validar aplicaciones desarrolladas en el lenguaje Java para los distintos bloques con los que cuenta la asignatura: complejidad algorítmica y técnicas de diseño de algoritmos. Y, en el caso de Fundamentos de Informática, esos tests se desarrollan para el lenguaje C++ y cubren, igualmente, todos los bloques docentes de dicha asignatura. En ambos casos, los tests se han configurado permitiendo la posibilidad de repetir los intentos (el alumno obtendrá la nota más alta de todos ellos) y una cadencia de 5 minutos entre cada intento; lo que permite, como veremos más adelante, un progreso incremental en la adquisición de competencias.

Con objeto de posibilitar que el proceso de autoevaluación funcione correctamente siempre se proporciona al alumno:

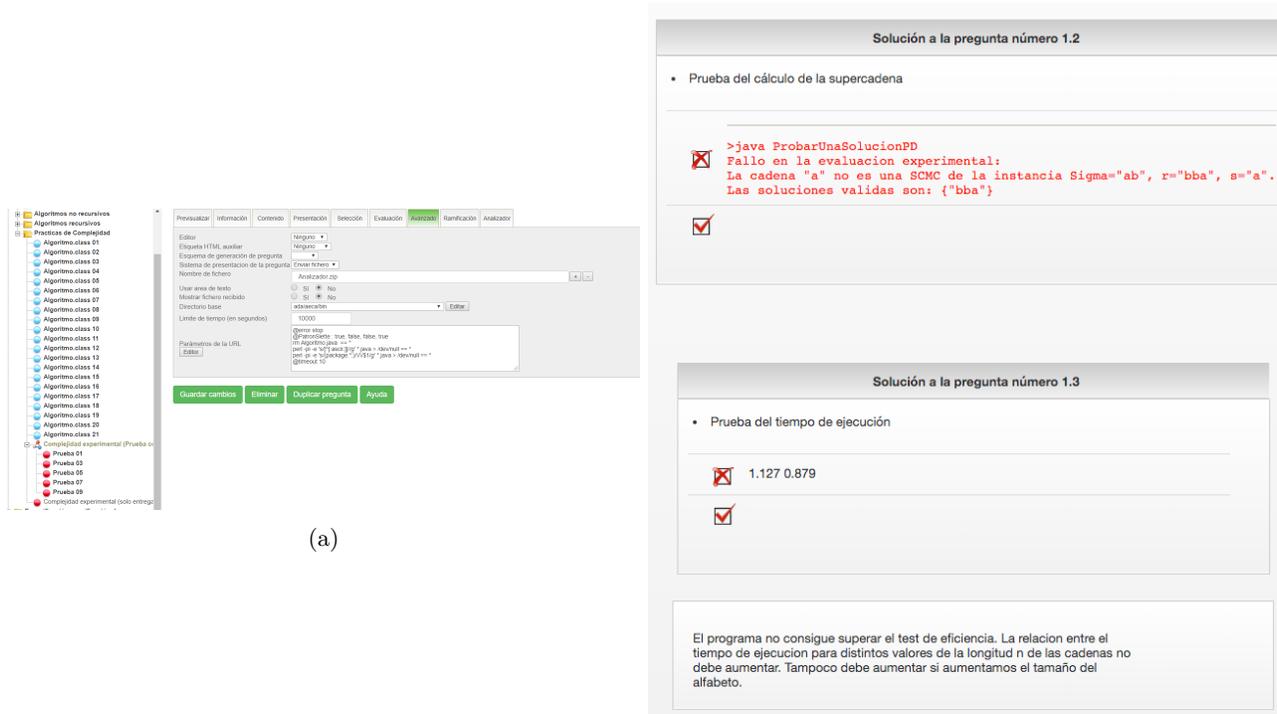
1. Un esqueleto en Java o C++ con todas las clases o subprogramas necesarios ya implementados y con la definición de algunos métodos o subprogramas que el alumno ha de implementar.
2. Una descripción detallada de la práctica, especificando claramente las entradas, las salidas esperadas y la funcionalidad esperada en los métodos o subprogramas propuestos.
3. Instrucciones detalladas de entrega: formato de ficheros, nombre de los mismos y cualquier otro requisito esperado por el servicio de corrección automática (SIETTE).
4. Los parámetros del entorno en los que se producirá la corrección, con lo que los alumnos puedan replicarlo para sus pruebas locales de manera previa a la entrega.
5. Un tiempo máximo para la ejecución del programa, tras el cual la corrección se detendrá.

En su evaluación, los tests diseñados se componen de una pregunta compuesta que a su vez contiene diversas preguntas que evalúan distintas partes del ejercicio: compilación, varias pruebas de corrección, tiempo de ejecución. y, opcionalmente para el caso de Análisis y Diseño de Algoritmos, un test automático de complejidad algorítmica experimental de forma que se pueda premiar a las soluciones propuestas más eficientes y mejor diseñadas. Las distintas preguntas que de forma transparente se evalúan con la entrega del alumno son secuenciadas en el orden esperado (por ejemplo, primero será necesaria evaluar la pregunta de compilación) y deseado de dificultad (con entradas cada vez más complejas y con casos frontera). De esta manera se puede ponderar el peso de cada pregunta a la calificación final obtenida en la entrega.

De manera previa al comienzo de la evaluación de las distintas preguntas, se realiza un preprocesamiento de los ficheros fuente entregados por los alumnos con objeto de evitar errores en el entorno de compilación (como acentos o códigos de caracteres distintos en los fuentes) mediante un script especialmente preparado para ello. También se establece el tiempo límite de ejecución del programa que está siendo corregido (ver figura 1a).

A continuación se lanza cada una de las pruebas de corrección en el orden preestablecido. Para cada una de ellas, que representan pruebas de la misma funcionalidad con distintas entradas, se realiza una prueba de compilación y de ejecución. En caso de que la primera falle se remite un error, se informa al usuario de la salida

del compilador y se detiene la corrección. En caso de que la ejecución no resulte en la salida esperada se informa al alumno de que la prueba que ha fallado, la salida esperada, la producida y se prosigue con la corrección. A continuación puede verse el tipo de información que recibe el usuario en caso de que una prueba falle (ver Figura 1b). Todas estas acciones son ejecutadas mediante el uso de un lenguaje de procesamiento creado para la corrección en SIETTE.⁴



(a)

(b)

Figure 1: a) Preprocesamiento de fuentes. b) Prueba del test con información sobre la corrección mostrada de forma inmediata

5. TESTS AUTOGENERATIVOS

Los tests autogenerativos son otra de las funcionalidades avanzadas que ofrece la herramienta SIETTE. Este tipo de tests se basa en un código capaz de generar, dinámicamente, tanto las preguntas como las respuestas que aparecerán en el test. El uso de preguntas autogeneradas tiene un gran interés para aquellos casos en los que se pueden definir múltiples instancias de un tipo de pregunta. Pueden servir como una fuente inagotable de pruebas que permitirán al estudiante probar una y otra vez su conocimiento sobre un tema. Evitan también la aparición de preguntas repetidas cuando un estudiante realiza muchas veces un test elaborado a partir de un conjunto de preguntas prefijado y limitado.

Desde el punto de vista técnico, las preguntas autogeneradas se basan en el uso de un generador de números aleatorios, que proporciona una semilla al código cada vez que debe generarse una pregunta. El código de generación de preguntas incorpora a su vez un elemento aleatorio que viene determinado por dicha semilla. Por tanto, debe ser capaz de reproducir un mismo enunciado de la pregunta y la correspondiente solución para cada semilla dada. De este modo se puede generar la pregunta, y una vez finalizado el test, comprobar la respuesta correspondiente a la semilla generada.

5.1 Tests autogenerativos para herramientas ofimáticas

La habilidad para usar herramientas ofimáticas es una competencia transversal presente en las asignaturas de informática que forman parte de múltiples grados, en su mayoría grados fuera del ámbito de la ingeniería

(ciencias, ciencias sociales, ciencias de la salud). Las preguntas diseñadas para esta experiencia pertenecen a la asignatura de Informática Aplicada a la Gestión Pública. En este contexto, uno de los aspectos más complejos es la resolución de problemas mediante el uso de las herramientas ofrecidas por los paquetes ofimáticos. Por ejemplo, un mismo cálculo se puede realizar de múltiples formas usando funciones en hojas de cálculo, o los resultados para una consulta sobre bases de datos se pueden conseguir con diferentes diseños.

Por ello, dada la cantidad de opciones disponibles que existen para evaluar el aprendizaje del correcto uso de una función (o consulta), es muy útil contar con herramientas que generen, de forma automática, diferentes preguntas. Aún más, ya que existen diversas formas de resolver una misma pregunta, es aún más necesario disponer de la posibilidad de construir, también automáticamente, diferentes soluciones que puedan encajar con las diversas estrategias de resolución que existan.

Para la definición de preguntas que permitan la generación de diversas alternativas de enunciado y múltiples respuestas correctas, SIETTE ofrece la posibilidad de utilizar el lenguaje de programación Javascript. Utilizando ese lenguaje y unos patrones, se definen todas las opciones posibles. En la Figura 2 se puede ver un ejemplo de este tipo de pregunta.

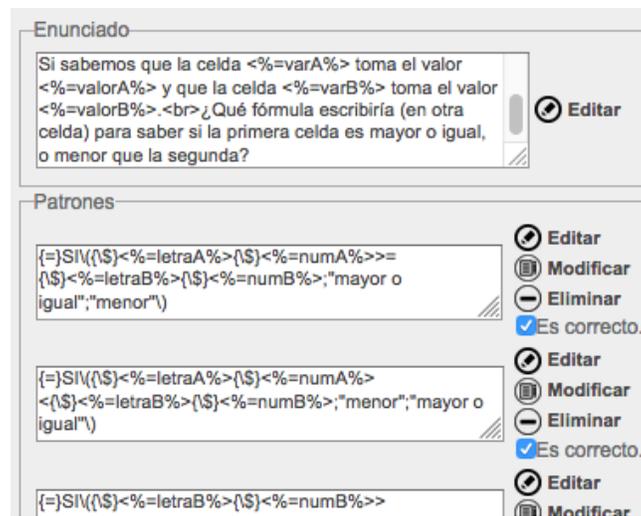


Figure 2: Definición de pregunta autogenerada con múltiples opciones de respuesta.

5.2 Test autogenerativo sobre el algoritmo A*

Como materia de evaluación mediante test también se ha seleccionado el algoritmo A*.⁵ Se trata de un algoritmo de búsqueda en grafos, enmarcado en un bloque dedicado a 'Búsqueda y resolución de problemas' (sistemas inteligentes), tradicionalmente incluido en las recomendaciones curriculares sobre estudios de informática.⁶ Los alumnos de esta asignatura suelen estar familiarizados con el problema de la búsqueda en grafos, fundamentalmente a través de materias relacionadas con Matemática Discreta o Análisis y Diseño de Algoritmos. El precedente más habitual suele ser el conocido como algoritmo de Dijkstra.⁷ El algoritmo A* incorpora a este el uso de una función de estimación heurística $h(n)$ que, bajo suposiciones razonables, mejora sustancialmente la eficiencia de la búsqueda.

En nuestro caso, todos los problemas generados en el test son problemas de búsqueda de caminos mínimos en redes de carreteras. El código genera una red de carreteras con un número de nodos en un rango prefijado, asegurando la conectividad entre todos ellos, así como un nodo de salida y otro objetivo. También se genera una función de estimación heurística optimista basada en la distancia Euclídea entre cada nodo de la red y el designado como objetivo. Para cada problema generado, el código comprueba igualmente que la solución cumple una serie de requisitos mínimos. Concretamente, comprueba que la solución requiere un número de iteraciones en un rango prefijado, lo que permite controlar la dificultad a la hora de resolver el problema. También comprueba que la solución es única, y que sólo haya una forma de resolver el problema (por construcción, la respuesta es única). Por último, se comprueba que la resolución del problema precisa de un conocimiento por parte del

alumno de aquellas características del algoritmo que se desean evaluar (por ejemplo, la redirección de punteros y las condición de terminación del algoritmo).

La Figura 3 muestra un ejemplo de enunciado sobre búsqueda con A* autogenerado por el test. El test está concebido para que el alumno deba resolver manualmente el problema antes de responderlo. Como respuesta se le pide que proporcione el valor de coste ($g(n)$) de todos los caminos generados por el algoritmo, así como la secuencia de nodos seleccionados por él mismo para la resolución. La herramienta resuelve el mismo problema y compara las respuestas.

Pregunta número 1:

Consideremos una red de carreteras entre varias ciudades.
La siguiente tabla muestra las distancias recorridas en kilómetros cuando hay una carretera que conecta dos ciudades directamente:

	c0	c1	c2	c3	c4	c5	c6	c7	c8
c0							11		8
c1			23			20			21
c2		23		13					25
c3			13				24		
c4						33	19	16	
c5		20			33			30	
c6	11			24	19				
c7					16	30			
c8	8	21	25						

Se pide aplicar el algoritmo A* para encontrar la ruta más corta desde la ciudad c7 hasta la ciudad c3, usando las siguientes estimaciones heurísticas de coste al objetivo (medidas en kilómetros):

Se pide aplicar el algoritmo A* para encontrar la ruta más corta desde la ciudad c7 hasta la ciudad c3, usando las siguientes estimaciones heurísticas de coste al objetivo (medidas en kilómetros):

n	c0	c1	c2	c3	c4	c5	c6	c7	c8
h(n)	11	5	11	0	17	12	17	13	16

Figure 3: Enunciado autogenerado del problema de búsqueda con A*

6. EVALUACIÓN

Tras realizar la experiencia en las distintas asignaturas participantes hemos constatado que:

1. La participación en las actividades decrecía conforme aumentaba el tiempo necesario para completar la prueba o aumentaba su complejidad. Esto se observó, principalmente, en las asignaturas de Informática Aplicada a la Gestión Pública, Fundamentos de Informática y Análisis y Diseño de Algoritmos.
2. Dicha participación, además, decrecía conforme avanzaba el curso. Es habitual que, a medida que avanza el semestre y el alumno ve próxima la fecha del examen, su interés por actividades en el caso de que no repercutan directamente en la calificación, aunque con ello mejore su aprendizaje, decaiga. En ese sentido, como ya ocurría en el punto anterior, ese decremento en el interés lo hemos observado en casi todas las asignaturas, aunque ha sido más explícito en las actividades propuestas para Informática Aplicada a la Gestión Pública y Fundamentos de Informática.
3. Habitualmente, quienes repiten las actividades de autoevaluación mejoran sus resultados en dichas actividades como quedó constancia en las asignaturas de Informática Aplicada a la Gestión Pública y Análisis y Diseño de Algoritmos. La autoevaluación mejora significativamente los procesos de aprendizaje del alumno además de ser una herramienta docente con muchas posibilidades a explorar por el profesor.

4. Los alumnos que llevaron a cabo la experiencia consiguieron realizar las posteriores repeticiones de las actividades de autoevaluación en menos tiempo. Ese hecho se constató tanto en Informática Aplicada a la Gestión Pública como en Análisis y Diseño de Algoritmos. La reducción del tiempo empleado puede deberse, entre otros, al conocimiento previo de la estructura del test, como a la mejora en el aprendizaje de los conceptos evaluados.

De forma adicional, se ha elaborado un conjunto de preguntas necesario para recabar información por parte de los alumnos y evaluar así su opinión acerca de la experiencia. Como resultado de esas reuniones, se formalizó en Moodle una encuesta compuesta por un total de siete preguntas (evaluadas siguiendo una escala Likert 1-4).

Al término de cada asignatura, se les presentó a los alumnos la encuesta diseñada. Se recopilaron un total de 48 encuestas y, tras ponderar adecuadamente las de los alumnos que habían participado en la actividad (37 alumnos), se obtuvieron los promedios para los diferentes ítems de la encuesta que están disponibles en el cuadro 2.

	TOTAL
Cuanto mayor es el peso de las pruebas de autoevaluación en la asignatura, mayor interés he puesto en hacerlas	3,1
El material de las pruebas de autoevaluación suministrado me ha parecido útil	3,4
Considera conveniente que la corrección se realizase en el mismo momento de hacer la prueba	3,6
Considera conveniente conocer la respuesta correcta, además de la nota alcanzada en la prueba (tuviese peso o no en la evaluación)	3,7
La cantidad de pruebas de este tipo es adecuada para la asignatura	3,3
Me gustaría que en otras asignaturas se incluyesen también este tipo de pruebas de autoevaluación.	3,5
Creo conveniente que en la mayoría de las clases hubiera una prueba rápida de autoevaluación sin nota para medir mi grado de asimilación de los contenidos	3,4

Table 2: Resultados de la encuesta de satisfacción siguiendo una escala Likert (1-4).

7. CONCLUSIONES

A raíz de los resultados obtenidos en cada una de las asignaturas y de cómo los alumnos han evaluado la experiencia, presentamos un conjunto de conclusiones organizadas por su temática.

7.1 Sobre el diseño de las pruebas de autoevaluación

- Las pruebas que se diseñen para la autoevaluación tienen que ser cortas. Es difícil mantener la concentración y/o interés en actividades de larga duración. Es preferible el diseño de distintas pruebas de la misma temática más cortas y organizadas por niveles de dificultad.
- En algunas pruebas, especialmente las prácticas de mayor complejidad, hemos observado que el alumno no efectúa todas las repeticiones que habrían sido aconsejables. Uno de los motivos para desistir en esta tarea es el desánimo consecuente de los primeros intentos. Con objeto de animar al alumnado a seguir progresando, creemos que sería conveniente informarles de los porcentajes de aciertos que son de esperar en sucesivas repeticiones según experiencias previas (bajos índices al principio y más altos conforme se repiten).
- Sería interesante ofertarles a los alumnos la posibilidad de repetir, en una segunda fase, sólo las preguntas para las cuales su solución propuesta no fue correcta inicialmente (primera fase). Esta segunda fase debería completarse inmediatamente después de finalizar el test de autoevaluación, para así poder reforzar los conceptos que hayan podido presentar mayor complejidad para el alumno.
- Las pruebas de autoevaluación diseñadas solo ofrecen al alumno la solución correcta para que este pueda contrastarla con la respuesta aportada por él. Pero puede ocurrir en algunos casos y con algunos alumnos que esta información resulte insuficiente. Para mejorar este diseño inicial, se puede hacer uso de la funcionalidad de SIETTE que permite incluir información de refuerzo que justifique la solución correcta, de

forma que el alumno pueda conocer el origen de su error y acelerar de esta forma su proceso de aprendizaje y asimilación.

7.2 Sobre el mantenimiento de la atención e implicación del alumno en la experiencia

- Una forma para conseguir la implicación del alumno en esta modalidad de autoaprendizaje mediante test de autoevaluación con corrección instantánea podría ser que las prácticas diseñadas contribuyan al cálculo de la calificación final. Las actividades que no tienen peso en la nota no suelen captar la atención del alumno en su aprendizaje autónomo. Definitivamente, la cuestión de la motivación del alumnado merecen un estudio futuro más profundo.
- Hemos constatado que la utilización de la herramienta decrece a medida que avanza el curso. Por ello, sería conveniente que, tras el uso de las primeras pruebas de autoevaluación, se muestren ejemplos reales anonimizados que pongan de manifiesto las ventajas de usar estas actividades. Y sería deseable recordarles, periódicamente, la importancia y utilidad de esta herramienta de autoaprendizaje, a fin de que los alumnos la exploren durante todo el curso académico y no solo en la fase inicial del curso.

7.3 Futuras ampliaciones

- En todas las pruebas analizadas, los resultados académicos obtenidos son positivos por lo que un mayor número de las mismas en las distintas asignaturas y/o bloques de las ya existentes debería contribuir al aumento de los niveles de aprendizaje.
- Del mismo modo nos planteamos, por un lado, introducir la autoevaluación en nuevas asignaturas para dar a conocer a los alumnos este recurso de aprendizaje. También sería conveniente ampliar el catálogo de actividades de autoevaluación en las asignaturas ya implicadas en este proyecto, estudiando y aplicando nuevas técnicas desarrolladas en la plataforma SIETTE.
- Actualmente contamos con algunas pruebas de autoevaluación implementadas sobre la herramienta Excel en lugar de sobre la plataforma SIETTE. Esta decisión es consecuencia de la imposibilidad de implementar en SIETTE algunos de los requisitos que exigían los ejercicios de autoevaluación que se precisaban. Actualmente esas limitaciones se están solventando y planeamos traducir de Excel a SIETTE las actividades que se hicieron en esa herramienta, para aprovechar el potencial que ofrece SIETTE a la hora de gestionar la información sobre su uso, así como para unificar la plataforma de autoaprendizaje.

AGRADECIMIENTOS

Este trabajo ha sido financiado parcialmente por los Proyectos de Innovación Educativa PIE17-168 y PIE19-149 de la Universidad de Málaga.

REFERENCIAS

- [1] Conejo, R., Guzmán, E., Millán, E., Trella, M., Pérez-de-la-Cruz, J., and Ríos, A., “SIETTE: A web-based tool for adaptive testing,” *I. J. Artificial Intelligence in Education* **14**(1), 29–61 (2004).
- [2] Romero, C. and Ventura, S., “Educational data mining: A survey from 1995 to 2005,” *Expert Systems with Applications* **33**, 135–146 (jul 2007).
- [3] Peña-Ayala, A., “Educational data mining: A survey and a data mining-based analysis of recent works,” *Expert Systems with Applications* **41**, 1432–1462 (mar 2014).
- [4] Conejo, R., Barros, B., and Bertoa, M. F., “Automated assessment of complex programming tasks using SIETTE,” *IEEE Transactions on Learning Technologies* **12**(4), 470–484 (2018).
- [5] Hart, P., Nilsson, N., and Raphael, B., “A formal basis for the heuristic determination of minimum cost paths,” *IEEE Trans. Systems Science and Cybernetics SSC-4* **2**, 100–107 (1968).
- [6] Shackelford, R., McGettrick, A., Sloan, R., Topi, H., Davies, G., Kamali, R., Cross, J., Impagliazzo, J., LeBlanc, R., and Lunt, B., “Computing curricula 2005: The overview report,” *SIGCSE Bull.* **38**, 456–457 (Mar. 2006).
- [7] Dijkstra, E. W., “A note on two problems in connexion with graphs,” *Numerische Mathematik* **1**, 269–271 (1959).