

POLITÉCNICA

LIBRO DE ACTAS

inece'09

**III Jornadas Internacionales UPM
sobre Innovación Educativa
y Convergencia Europea 2009**

e 

24, 25 y 26 de noviembre
Universidad Politécnica de Madrid
http://innovacioneducativa.upm.es/inece_09/

**Escuela Universitaria de Ingeniería
Técnica de Telecomunicación**



Actas de las Jornadas

 **Introducción**

 **Programa**

 **Comunicaciones**

- Por Sesiones
- Libro de Actas

 **CERRAR**

IMPLEMENTACIÓN DE PSEUDO-MÓDULO MOODLE PARA DETECCIÓN DE PLAGIO DE CÓDIGO FUENTE A TRAVÉS DE SERVIDOR MOSS

David Úbeda*, Arturo Gil, Óscar Reinoso, Luis Payá

Dpto. de Ingeniería de Sistemas Industriales
Escuela Politécnica Superior de Elche
Universidad Miguel Hernández

Avda. de la Universidad S/N. Ed. Quorum V. 03202 Elche (Alicante)
e-mail: {ubeda, arturo.gil, o.reinoso, lpaya}@umh.es web: <http://arvc.umh.es>

Resumen. *En este artículo se presenta una herramienta informática y su aplicación a la asignatura “Informática Aplicada”, obligatoria de Universidad. En la citada asignatura, el alumno es evaluado mediante un conjunto de prácticas de programación en C/C++ que éste debe presentar de forma individualizada. En cursos anteriores, el sistema de evaluación contaba con un gran problema: con frecuencia, los alumnos entregaban un código fuente plagiado, completamente o parcialmente, de otros compañeros. Para solventar el problema, hemos desarrollado un pseudo-módulo para Moodle capaz de detectar el plagio de código fuente de las prácticas de programación C/C++ enviadas a la plataforma. Dicho módulo devuelve una estadística completa al profesor acerca de los alumnos copiados, partes del código plagiadas, disminución o aumento del nivel de copia respecto a prácticas anteriores, etc. Además guarda un histórico de código fuente de las prácticas de años anteriores, con el cual se comparan las prácticas del curso actual, que puedan tener similitudes en el código con aquellas.*

1. Introducción

El presente artículo no pretende discutir ninguno de los algoritmos de detección de plagio existentes actualmente. Lo que se pretende es buscar una solución al alto índice de copia detectado en las asignaturas de primeros cursos en Ingeniería, concretamente y para el caso que nos ocupa, para la asignatura Informática Aplicada de primer curso de Ingeniería Industrial en la Universidad Miguel Hernández.

El índice de matriculados en esta asignatura suele rondar los 140 alumnos por curso tomando como media de las diez últimas promociones; número elevado por ser una asignatura de primer curso y por tanto obligatoria su matriculación para los alumnos de nuevo ingreso. A este número de alumnos cabe sumarle los alumnos repetidores, que suelen ser numerosos debido, principalmente, a la tasa de abandono de la asignatura, quizás porque no hayan recibido docencia en asignaturas similares en bachillerato, o ésta les resulte dificultosa de aprobar.

Debido a este gran número de alumnos matriculados, decidimos realizar un sondeo inicial el primer día de clase, relativo al motivo de elección de Ingeniería Industrial como titulación para la preparación a la dedicación profesional. El descubrimiento fue desalentador, puesto que entre un 70-80% de los alumnos, o bien habían escogido la titulación de Ingeniero Industrial como segunda opción en su elección de matriculación, o la habían escogido al azar, o incluso por el nombre de la titulación sin ni siquiera conocer los contenidos de las asignaturas. El resto de alumnos, entre un 20 y un 30%, decían haber escogido la titulación por tener algún

familiar trabajando en alguna competencia profesional derivada de la titulación, o haber leído el temario de las asignaturas y encontrar identificadas ciertas aptitudes que desean desarrollar o conocer para su dedicación profesional.

Distribución:

90% - 100%	12 #####
80% - 90%	2 #
70% - 80%	30 #####
60% - 70%	21 #####
50% - 60%	34 #####
40% - 50%	89 #####
30% - 40%	115 #####
20% - 30%	48 #####
10% - 20%	38 #####
0% - 10%	17 #####

Coincidencias clasificadas por la semejanza media (¿Qué es esto?):

alvaro_campo_g1.c	>	carlos_olivera_g1.c	carlos_olivera_g1.c	carlos_olivera_g1.c	(94.2%)
alvaro_campo_g1.c	>	carlos_olivera_g1.c	carlos_olivera_g1.c	carlos_olivera_g1.c	(75.5%)
carlos_olivera_g1.c	>	carlos_olivera_g1.c	carlos_olivera_g1.c	carlos_olivera_g1.c	(75.5%)
carlos_olivera_g1.c	>	carlos_olivera_g1.c	carlos_olivera_g1.c	carlos_olivera_g1.c	(90.3%)
carlos_olivera_g1.c	>	carlos_olivera_g1.c	carlos_olivera_g1.c	carlos_olivera_g1.c	(94.2%)
carlos_olivera_g1.c	>	carlos_olivera_g1.c	carlos_olivera_g1.c	carlos_olivera_g1.c	(99.5%)
carlos_olivera_g1.c	>	carlos_olivera_g1.c	carlos_olivera_g1.c	carlos_olivera_g1.c	(95.8%)
carlos_olivera_g1.c	>	carlos_olivera_g1.c	carlos_olivera_g1.c	carlos_olivera_g1.c	(88.8%)

Figura 1. Distribución de alumnos copiados de un grupo durante el curso 2007/08

Ante estas cifras abrumadoras, nos damos cuenta que, ya de raíz, nos vamos a encontrar con un problema de motivación o desinterés en algunas de las asignaturas por parte del alumnado, con lo que no resultará complejo pensar que pueda existir un nivel de plagio elevado.

Dada la situación, nos pusimos manos a la obra y decidimos idear un plan para tratar de reducir esta previsión de plagio. En un primer momento decidimos conceder el beneplácito de la duda y así procedimos a impartir las dos primeras prácticas. Una vez corregidas ambas prácticas y comprobada la similitud de código, a través de JPlag [6], decidimos buscar e integrar una herramienta que inquietara al alumnado debido a la eficacia en la detección, así como proporcionarnos al profesorado una herramienta de evaluación más objetiva y con datos más precisos acerca del plagio. Esta herramienta deseábamos que se pudiera integrar con el CMS de Moodle, puesto que era el portal a través del cual los alumnos accedían a depositar las prácticas realizadas.

2. Problemática actual con la corrección del código fuente de prácticas de programación: “El miedo a las consecuencias”

El plagio de prácticas en docencia es un tema ampliamente abarcado en múltiples congresos y jornadas, por profesores responsables de asignaturas de las más variadas: desde la titulación de Antropología hasta Ingeniería. Es un problema complicado de resolver puesto que por un lado hiere el amor propio del profesor, ya que a nadie le agrada que copien en su materia, y por otro lado demuestra la falta de responsabilidad del alumnado que llega por primera vez a la universidad, donde quizás no tiene todavía percepción de por qué tiene que estudiar o realizar una tarea de una asignatura, al margen de la consecución del aprobado. Hablamos de los primeros cursos de las titulaciones, donde radica el mayor de los problemas, puesto que está

comprobado que el número de copiados es mucho más alto que en cursos próximos a la finalización de los estudios.

Figura 2. Nuevo recurso de plagio en Moodle

Desde nuestro punto de vista, la solución, aunque compleja, no solo pasa por realizar un aviso al inicio del curso acerca de la detección de plagio y sus consecuencias, es conveniente demostrarlo al menos una vez cada dos o tres cursos, puesto que parece que lo que mejor funciona, coincidiendo con nuestro colega Clemente [1], es el miedo a las consecuencias, y por ende, la transmisión de esas consecuencias a través del boca a boca entre alumnos repetidores y de nuevo acceso. Es decir, el miedo a las consecuencias parece muy eficaz en todos los casos y con todas las herramientas empleadas.

Es cierto, que el alumno cree, en la mayoría de las ocasiones, que el profesor no se va a notar el plagio realizado, y quizás, en asignaturas donde el número de alumnos es elevado, tengan en cierto modo razón, puesto que sin un soporte informático como el que vamos a comentar, resulta casi imposible conocer si existe o no copia de las mismas, puesto que es imposible recordar las prácticas a medida que se corrigen para comprobar su autenticidad con el resto.

Como comentábamos en la introducción, realizamos un experimento, donde propusimos un par de prácticas iniciales, y no avisamos de que se realizaría un control de plagio. Este ejemplo se realizó durante el curso 2007/08 en la asignatura de Informática Aplicada, y encontramos que el número de alumnos con prácticas completamente o parcialmente plagiadas (Fig. 1) en un solo grupo era muy superior al del número de plagios detectados en los 5 grupos de dicha práctica durante el curso siguiente.

Además, hemos comprobado que la comprobación de plagio tiene otras consecuencias: como consecuencia de que los alumnos se cercioraran que controlábamos el plagio, el nivel de copiados durante ese curso se redujo drásticamente para el resto de prácticas, aunque desgraciadamente, el número de alumnos que presentaban prácticas también se redujo de igual forma. Resulta evidente pensar que esos alumnos coincidían con el número de alumnos que no querían realizar un esfuerzo por realizar las prácticas y que, muy probablemente, pretendían plagiarlas de sus compañeros.

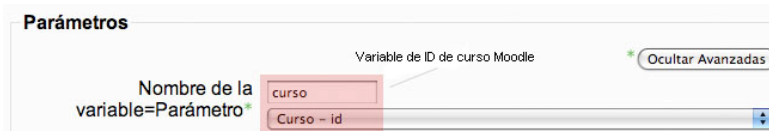


Figura 3. Lectura de variables propias de Moodle

3. Estado del arte de las aplicaciones de detección de plagio

Puesto que no se pretendía en ningún momento generar nuevo software a partir de los numerosos algoritmos que existen en la actualidad, acerca de la detección de plagio, decidimos acudir a la comunidad de software libre y buscar una determinada aplicación que se adaptara a nuestras necesidades. Concretamente, aquello que se buscaba se podía englobar en dos características fundamentales: por un lado, máxima capacidad de integración con nuestro sistema de recogida de prácticas, que no era otro que Moodle [7]; y por otro, que fuera un servidor externo a nuestra red el que implementara el algoritmo y devolviera los resultados.

Para ello, era necesario que no dependiera de una aplicación cliente que hubiera que instalar en nuestro equipo de recogida de prácticas, y que además, la comunicación en todo momento fuera en un lenguaje totalmente integrable con nuestra web, tipo shell scripting, C/C++ o Perl [9].

Si atendemos a la clasificación actual de Goel [4] en cuanto a las aplicaciones más relevantes de detección de plagio del código fuente, podemos descartar todas aquellas que utilicen el sistema a través de una aplicación distinta a una comunicación TCP/IP, es decir, que la aplicación se corriera en servidor. Además, requeríamos que la aplicación utilizara un interfaz web, puesto que de otro modo, la compatibilidad que se pretendía sería inviable o excesivamente complicada de realizar, en comparación con la rápida integración necesaria para el curso académico que se iba a impartir.

Es por ello, que descartamos el uso de JPlag, Sherlock [11], CodeMatch [2] y CPD [3], no sin antes probarlas todas en busca de la que mejores resultados obtuviese, respecto a las características exigidas.

JPlag fue la que más nos impresionó positivamente, por la rapidez en la respuesta del servidor, la métrica derivada del resultado, que el código sospechoso se encontrara resaltado y que el interfaz gráfico o la web a través de la cual se presentaban los resultados fuera ágil e intuitiva (Fig. 1).

Esta aplicación se basaba en el algoritmo *Greedy String Tiling* [5]. De hecho, las primeras pruebas decidimos realizarlas a través de esta plataforma. El problema se presentaba a la hora de que varios profesores utilizaran la aplicación o que ésta se usara desde distintos ordenadores. El cliente de la aplicación es un programa realizado en Java, y por tanto era necesario descargar la aplicación cada vez que se pretendía usar. Aunque era muy efectiva, no nos parecía la más optimizada para usarla con nuestro sistema CMS.

Decidimos, por tanto, continuar con la búsqueda de la aplicación que mejor se adaptara a nuestras necesidades, y observando las características fundamentales de

la aplicación que se había desarrollado en la Universidad de Stanford [12], MOSS [8], decidimos que, sin duda, la habíamos encontrado, puesto que nos facilitaba un método de envío de prácticas mediante sockets, a través de línea de comandos, programados en un sencillo script realizado en lenguaje Perl, muy rápido a la hora de

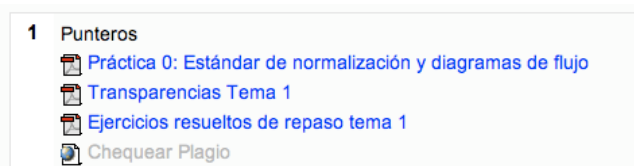


Figura 4. Acceso a la aplicación de chequeo de plagio

integrarlo en Moodle. La recogida de información devuelta por la petición HTTP, no era otra que una dirección web donde se alojaban los resultados de la comprobación de plagio: en la universidad de Stanford. El algoritmo que emplea MOSS para comparar el código, es el algoritmo Winnowing [13], que aunque menos potente que el anterior algoritmo citado, era suficientemente eficiente para nuestras prácticas.

Para que la aplicación se adaptara de forma perfecta a nuestras necesidades, requeríamos una última característica que MOSS no nos facilitaba: almacenar los resultados de la comprobación de plagio en nuestro servidor. No fue un gran problema implementar esta funcionalidad, a través de la descarga automatizada del directorio completo y de sus 2 niveles de subdirectorios, donde se almacenan imágenes, etc., e incluirla en nuestra aplicación, de tal forma que es posible almacenar en Moodle todos los resultados de la detección de plagio para su posterior consulta, sin temer que la universidad de Stanford nos borre dichos archivos temporales.

4. Interfaz de Uso de MOSS a través de Moodle



Figura 5. Muestra del envío de prácticas

Además, la decisión de utilizar MOSS para su integración con Moodle fue acertada puesto que ésta resultó relativamente sencilla, generando una especie de pseudo-modulo para este CMS. Lo denominamos pseudo-modulo porque realmente no es un módulo instalable en ninguna de las versiones de Moodle, sino que es una aplicación independiente que utiliza la estructura de ficheros y de base de datos MySQL que crea Moodle.

Por un lado, decidimos emplear la base de datos MySQL de Moodle para no tener que ir entregando a los alumnos un nuevo usuario y contraseña, que en la práctica resulta algo poco operativo que el alumno tenga varias claves distintas para entrar en las asignaturas.

De esta forma, y utilizando la estructura de directorios que genera Moodle para la entrega de tareas individualizadas, realizamos una serie de scripts en PHP [10] que nos

permitieran realizar un enlace con el servidor de MOSS de la universidad de Stanford, y nos permitiera almacenar el link, con los resultados del chequeo de plagio, en una variable de nuestro script PHP, la cual utilizaríamos a su vez para mostrarla en nuestra aplicación.

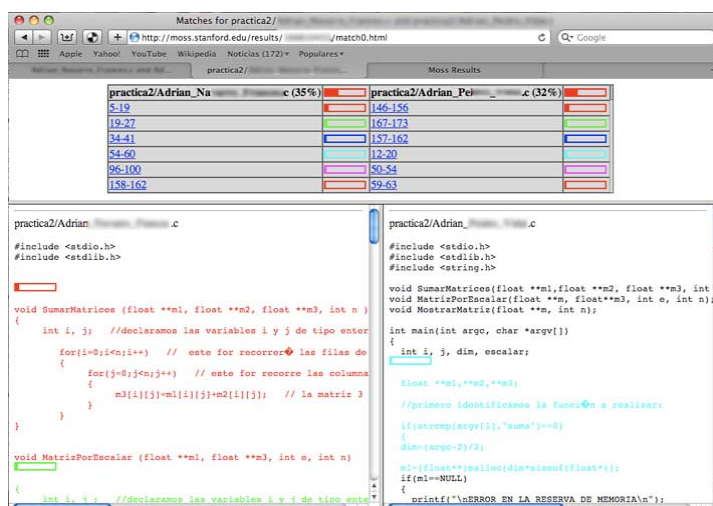


Figura 6. Resultado de la comprobación de plagio

5. Generación de un nuevo recurso de plagio en Moodle

El funcionamiento de nuestro pseudo-módulo es bien sencillo, tal y como se muestra en la Figura 2.

Inicialmente será necesario editar el tema o la práctica en Moodle, y añadir un nuevo recurso de enlace. Lo más importante es añadir la variable PHP correspondiente a la práctica en cuestión, a la dirección donde se encuentra nuestro script de solicitud de envío de prácticas a MOSS.

Otra cuestión importante para que nuestra aplicación funcione correctamente, será pasarle el identificador de curso Moodle o id de curso (Fig. 3). Mediante este identificador, la aplicación conocerá el directorio del cual debe extraer los ficheros, puesto que el número de tarea se la indicamos nosotros en el recurso que hemos creado anteriormente.

A partir de este momento, y de forma oculta para el alumnado, se ha construido nuestro acceso a la aplicación de control de plagio (Fig. 4).

Para realizar una consulta del número de alumnos plagiados, no sería necesario mas que entrar con la cuenta de cualquiera de los profesores de la asignatura Moodle, y a continuación pulsar sobre el acceso “Chequear Plagio”, una ventana desplegable se nos abriría mostrando los resultados de dicha comprobación de copia de la práctica concreta (Fig. 5 y 6)

6. Conclusiones

Como conclusiones al trabajo presentado, cabe decir que hemos obtenido una herramienta versátil y perfectamente adaptable a cualquier tipo de sistema docente CMS tipo Moodle. Este trabajo nos ha llevado a conocer cómo varía nuestra asignatura respecto al número de alumnos con plagio en nuestras prácticas, a través de la presentación de una aplicación que, si bien no va a acabar con las prácticas plagiadas, al menos servirá como lección a aquellos que lo intenten de forma muy evidente, ya no tanto por el miedo a las consecuencias del plagio, que en la mayoría de veces acaba en suspenso, sino por el miedo al ridículo a ser detectado dicho plagio. Como consecuencia, hemos reafirmado aquello que ya imaginábamos, y es que si algunos alumnos no se copiaran, abandonarían la asignatura. Ello nos ha hecho mejorar en nuestras encuestas de calidad docente respecto al número de presentados-aprobados, pero nos ha hecho reflexionar en la cuestión principal, y en la que nos encontramos actualmente trabajando: ¿cómo podemos motivar de una forma mayor a los alumnos de asignaturas de programación en primeros cursos?

REFERENCIAS

- [1] Pedro J. Clemente, Alberto Gómez, Julia González, *La copia de prácticas de programación: el problema y su detección*, Dpto. de Informática, Universidad de Extremadura, JENUI'04
- [2] Codematch, <http://www.zeidmanconsulting.com/CodeSuite.htm>
- [3] Copy/Paste Dectector (CPD), <http://pmd.sourceforge.net/cpd.html>
- [4] Sanjay Goel, Deepak Rao et. al., *Plagiarism and its Detection in Programming Languajes*, Department of Computer Science and Information Technology, JIITU□May, 2008
- [5] Greedy String Tiling Algorithm, Lutz Prechelt, Guido Malpohl and Michael Philippsen, "JPlag: Finding Plagiarisms among a Set of Programs", 2000. <http://page.mi.fuberlin.de/prechelt/Biblio/jplagTR.pdf>.
- [6] JPlag, <https://www.ipd.uni-karlsruhe.de/jplag/>
- [7] Moodle, <http://www.moodle.org>
- [8] Alex Aiken. MOSS (Measure Of Software Similarity) plagiarism detection system. <http://www.cs.berkeley.edu/~moss/> (as of April 2000) and personal communication, 1998. University of Berkeley
- [9] Perl scripting Language, <http://www.perl.org>
- [10] PHP Scripting Language, <http://www.php.net>
- [11] The Sherlock Plagiarism Tool, <http://www.cs.usyd.edu.au/~scilect/sherlock/>
- [12] Stanford University, <http://www.stanford.edu>
- [13] Saul Schleimer, Daniel S. Wilkerson, Alex Aiken, *Winnowing: Local Algorithms for Document Fingerprinting*, SIGMOD 2003, June 9-12, 2003, San Diego, CA.
- [14] J.C. Sánchez and R.L. Taylor, "La innovación educativa en la universidad española", *Revista de Educación*, Vol. 48, pp. 101-118, (2005).