

El Abogado del Diablo como Técnica de Trabajo Cooperativo

Pablo Sánchez

Dpto. Matemáticas, Estadística y Computación
Universidad de Cantabria
Facultad de Ciencias, Avda. Los Castros S/N
39071 Santander (Cantabria)
p.sanchez@unican.es

Resumen

El Espacio Europeo de Educación Superior introduce la necesidad de formar a los alumnos en nuevas competencias profesionales les permitan afrontar su inserción en el mercado laboral con mejores garantías. Una de las competencias profesionales más demandadas por las empresas es la de trabajo cooperativo. Surge por tanto la necesidad de desarrollar metodologías para que los alumnos aprendan a trabajar de forma cooperativa eficazmente. Este artículo presenta una metodología de trabajo cooperativo, denominada *El abogado del diablo*, aplicable a las materias relacionadas con la programación de computadores. Un valor añadido de esta metodología es que forma sin esfuerzo adicional alguno a los alumnos en competencias relacionadas con la Ingeniería del Software.

Summary

Training students in new, and not strictly academic, professional skills has become practically mandatory with the emergence of the new European Space of Higher Education. The goal is to increase opportunities for employment of recently graduated students. One of the professional skills most demanded by companies is teamworking. Therefore, it becomes necessary to create new teaching methodologies which allow students to learn how to work as a team. This paper presents a methodology for teamworking, called "the devil's advocate", which is suitable for teaching subjects related to computer programming in the first courses of a Computer Science degree. Moreover, a positive side effect of our methodology is that students are also trained on Software Engineering skills without any additional effort.

Palabras clave

Trabajo Cooperativo, Competencias Profesionales, Programación, Metodologías Docentes

1. Introducción

El Espacio Europeo de Educación Superior, entre muchos otros objetivos, establece la necesidad de formar a los alumnos universitarios en una serie de competencias profesionales [7] que suelen ir más allá de las meramente académicas. Entre estas competencias se encuentran habilidades tales como la capacidad de iniciativa y de liderazgo, el aprendizaje autónomo o la capacidad de trabajar en equipo. Estas competencias plantean nuevos retos a los docentes universitarios, los cuales tienen en muchas ocasiones que formar a los alumnos en competencias en las cuales el propio docente no ha sido formado.

Surge por tanto la necesidad de crear nuevas metodologías docentes que asistan a los profesionales de la enseñanza en los procesos de formación en las nuevas competencias profesionales. Idealmente, estas nuevas metodologías deberían integrarse con las metodologías destinadas a la formación en competencias propias de cada titulación universitaria. Es decir, se trata de que el alumno adquiera las competencias profesionales a la vez que se adquiere las competencias propias de su titulación. La formación en competencias profesionales no sólo debería no interferir con la formación en competencias académicas, sino que debería favorecer la adquisición de estas últimas.

Una de las competencias profesionales más demandadas por la mayoría de los empleadores actualmente es la capacidad de trabajo en equipo. Esta competencia es particularmente difícil de trabajar en materias relacionadas con la Programación de Computadoras, sobre todo en los primeros cursos. El principal problema de desarrollar software en equipo es que, aparte de una sólida y eficaz dirección, nece-

sita de una infraestructura tecnológica, tales como repositorios de código o sistemas de control de versiones, que en muchas ocasiones no están fácilmente disponibles en el entorno universitario. Además, en los primeros cursos, dado el tamaño de las aplicaciones desarrolladas por los alumnos, resulta especialmente difícil dividir en desarrollo de una aplicación en partes o módulos independientes que permitan a los alumnos trabajar de forma eficaz como un equipo.

Este trabajo presenta una metodología basada en una técnica conocida de trabajo en equipo [5, 8] denominada *el abogado del diablo*, que resulta de especial utilidad para realizar trabajos prácticos de Programación en parejas. Dicha metodología se adecua a las necesidades de los primeros cursos de una titulación universitaria y tiene como ventajas adicionales:

- Ayuda a formar al alumno en otras competencias profesionales, como el aprendizaje autónomo;
- Predispone al alumno para el aprendizaje de competencias específicas relacionadas con la Ingeniería del Software, tales como la realización metodológica de pruebas o la gestión de versiones de un productos software;
- Mejora la percepción que el alumno tiene del trabajo cooperativo.

Dicha metodología ha sido aplicada con éxito en la asignatura de Estructura de Datos y Algoritmos perteneciente a la titulación de Ingeniería en Informática de la Universidad de Cantabria. El presente artículo describe dicha metodología así como los resultados obtenidos.

Tras esta introducción, este artículo se estructura de la siguiente forma: La sección 2 explica la motivación existente tras este trabajo. La sección 3 describe la metodología de trabajo en equipo basada en el abogado del diablo particularizada para materias de primeros cursos relacionadas con la Programación de Computadoras. La Sección 4 comenta los beneficios esperados, mientras que la Sección 5 discute los resultados obtenidos tras aplicar esta metodología. Finalmente, la Sección 6 resume las conclusiones principales extraídas de este trabajo.

2. Motivación

El principal problema que presenta la realización de trabajos en equipo en materias relacionadas con la Programación de Computadoras en los primeros cursos de una titulación relacionada con la Ingeniería en Informática es que el tamaño de las aplicaciones software desarrolladas por los alumnos no alcanza el mínimo necesario para que la aplicación pueda ser fácilmente dividida en partes o módulos independientes.

El principal inconveniente es que la metodología de trabajo cooperativo normalmente utilizada por los alumnos se suele basar en varios estudiantes sentados alrededor de una computadora observando como sólo uno de ellos programa. El resto, en el mejor de los casos, se limita a: (1) verificar que el que está programando no comete errores; (2) hacer sugerencias; y (3) proponer mejoras. En esta escenario, el hecho de trabajar en equipo se considera estéril e improductivo, y así lo perciben muchos alumnos.

Aunque esta situación pueda tener cierto parecido con técnicas de desarrollo software modernas, tales como la *programación por pares* [1], entendemos que las habilidades y capacidades de los estudiantes de los primeros cursos no están lo suficiente maduras como para poder sacar beneficio de tales técnicas. Por ejemplo, en el caso de la programación por pares, mientras un programador está codificando, el otro supervisa el proceso y propone mejoras en aras de una mayor facilidad de adaptación y evolución del producto software. Para que esta técnica funcione de forma efectiva, se presupone que ambos programadores son programadores experimentados y tienen amplio conocimiento de técnicas avanzadas de diseño software, como un sólido conocimiento sobre Patrones de Diseño [3]. Obviamente, las competencias adquiridas por los alumnos al comienzo de un segundo curso de una titulación como Ingeniería en Informática no son suficientes para alcanzar este nivel.

Cabe destacar que si pretendemos enseñar a nuestros alumnos a trabajar en equipo, el primer paso es hacer que comprendan el valor del trabajo cooperativo. Según los datos recopilados entre el alumnado antes de aplicar nuestra metodología, la mayoría de los alumnos consideraban que el hecho de trabajar en equipo no les aportaba ningún beneficio. Argumentaban que podían concluir las tareas asignadas

con mayor rapidez si las realizaban de forma individual, en lugar de realizarlas en equipo. Además, en el caso de las tareas a realizar fuera del aula, el hecho de trabajar en equipo les obligaba a reunirse fuera del periodo de docencia presencial, con el consiguiente esfuerzo de coordinación. Todo esto aumenta el tiempo dedicado al trabajo en equipo y su descrédito. Por tanto, la realidad actual era que los estudiantes percibían el trabajo en equipo como algo que les perjudicaba más que beneficiarles. Si los alumnos no perciben las ventajas de trabajar de forma cooperativa, difícilmente podrán aprender a trabajar cooperativamente. Se hace por tanto patente la necesidad de crear metodologías que permitan a los alumnos valorar el trabajo cooperativo. Es decir, se trata de crear técnicas de trabajo en equipo que incrementen la productividad en lugar de decrementarla.

La siguiente sección describe la metodología del abogado del diablo aplicada al desarrollo de pequeñas aplicaciones software por parejas de estudiantes. Una de las ventajas de esta metodología es que permite a los alumnos trabajar de forma individualizada cooperativamente. Se benefician del hecho de pertenecer a un grupo sin que ello les suponga un gran esfuerzo adicional, viendo aumentada su productividad.

3. Descripción de la Metodología

La figura del *abogado del diablo* tiene un origen eclesiástico. Su función es la de verificar si un hecho considerado como milagroso es realmente un milagro o, por el contrario, es un hecho extraordinario pero basado en una explicación perfectamente racional. En este segundo caso, se descarta la intervención divina y no se le atribuye carácter milagroso. La persona que desempeña esta tarea, por tanto, parece empeñada en negar la existencia de Dios, de ahí que se le considere un mensajero o un abogado del diablo.

El papel de abogado del diablo se introdujo en las técnicas de trabajo en equipo hace ya varias décadas [8, 1]. Fue empleada incluso por el presidente americano John F. Kennedy. La idea de Kennedy era la de disponer de un asesor que criticara enérgicamente las decisiones de su gabinete. Si Kennedy lograba rebatir con contundencia las críticas de este asesor, las decisiones se consideraban acertadas. Además, el presidente adquiriría la destreza necesaria

para rebatir tales críticas si éstas eran planteadas por algún miembro de la oposición o de la prensa.

La metodología que proponemos en este artículo está basada en crear parejas de estudiantes en las cuales uno realiza una primera parte del trabajo asignado, y el otro ejerce el papel de abogado del diablo. El abogado del diablo evalúa a su propio compañero de forma exhaustiva. A continuación, los alumnos intercambian los papeles. De esta forma, el que ha ejercido de abogado del diablo continúa con el trabajo asignado; y el que había realizado el trabajo, pasa a ejercer el papel de abogado del diablo.

A continuación, exponemos como se ha aplicado esta metodología a la implementación de un Tipo Abstracto de Datos dentro de la asignatura de Estructuras de Datos y Algoritmos de la titulación de Ingeniería Informática. La asignatura pertenecía al segundo curso y tenía un total de 42 alumnos.

La tarea a realizar por los alumnos era implementar, en un lenguaje que nunca hubiesen utilizado anteriormente y de su elección, un Tipo Abstracto de Datos (por ejemplo, un conjunto implementado sobre listas enlazadas). El objetivo último del trabajo era fomentar el aprendizaje autónomo de nuevos lenguajes de programación, competencia fundamental para todo Ingeniero Informático. Dicha tarea se debía realizar por parejas, que podían ser libremente formadas por los alumnos de acuerdo a sus propias afinidades. Cada grupo debía elegir, según sus capacidades y expectativas, un Tipo Abstracto de Datos y una técnica de implementación (por ejemplo, conjuntos sobre tablas de dispersión).

Una vez elegido el Tipo Abstracto de Datos a implementar, la metodología de trabajo propone como siguiente etapa dividir la interfaz del mismo en grupos de operaciones relativamente independientes que se puedan implementar de forma secuencial. Lo mismo se aplica a la implementación del soporte para dicho Tipo Abstracto de Datos. Por ejemplo, suponemos que una pareja de alumnos escogen implementar un caso de conjunto implementado sobre tablas de dispersión. Habría que implementar primero una tabla de dispersión y a continuación, usando dicha tabla de dispersión, implementar un conjunto.

Las operaciones a implementar para este TAD podrían dividirse en los siguientes bloques:

1. Implementación de las operaciones de inserción, búsqueda y eliminación en la tabla de dis-

130 Sesión 3A: Arquitectura de computadores / Sistemas operativos / Sistemas distribuidos y paralelos

- persión.
2. Implementación de las operación de creación, inserción, eliminación y búsqueda en un conjunto (implementado sobre una tabla de dispersión).
 3. Implementación de las operaciones de unión, intersección y diferencia entre conjuntos.
 4. Implementación de las operaciones cardinal, subconjunto e igualdad entre conjuntos.

Una vez divididas estas operaciones en bloque, la idea principal de la metodología es que un estudiante implemente sin la presencia de su compañero. Por tanto, no se requieren esfuerzos de coordinación. Su compañero va a ejercer el papel de abogado del diablo sobre la implementación creada. El fundamento y motivación para el alumnado reside en que resulta mejor que los fallos cometidos por un compañero los descubra otro compañero antes de que el trabajo sea calificado por el docente, momento en el cual dichos fallos ya no serían reparables.

Es decir, la metodología de trabajo se basa en que un alumno implementa y el otro prueba y *certifica* el programa. Usamos el verbo certificar porque las tareas del abogado del diablo van más allá de la realización de pruebas exhaustivas.

Concretamente, las tareas del abogado del diablo son verificar que:

1. El programa funciona para todos los casos posibles. Ello implica que el abogado del diablo debe diseñar un conjunto exhaustivo de pruebas y ejecutarlas, con objeto de comprobar que el código creado por su compañero funciona para todos los casos posibles.
2. La documentación de los métodos es adecuada, es decir, es suficiente, precisa y concisa.
3. Las precondiciones para los métodos declarados son correctas y suficientes y que siempre que se satisfacen ningún método lanza una excepción (se sigue en la asignatura un proceso de aprendizaje basado en *Diseño por Contrato* [6]).
4. Siempre que se invoca un método se satisfacen las precondiciones, (de nuevo de acuerdo con los principios del *Diseño por Contrato*).
5. En los comentarios y documentación de los métodos no existen errores tipográficos u ortográficos.
6. No se incumplen las normas de estilo de pro-

gramación contenidas en la *Guía de Buenas Prácticas en Programación* proporcionada al comienzo de la asignatura.¹

7. El código creado se entiende bien, ya sea porque es trivial o porque su funcionalidad se deduce fácilmente de la documentación de los métodos o de los comentarios insertados en el código.
8. La información que se muestra al usuario por pantalla es clara, precisa, concisa y está libre de errores tipográficos u ortográficos.

Es decir, el abogado del diablo trabaja con una *rúbrica* idéntica a la que usaría el profesor para calificar dicho trabajo. De esta forma, el trabajo producido por un estudiante es precalificado por su compañero de trabajo. Si éste descubriese fallos, éstos se podrían corregir antes de que fuese demasiado tarde, es decir, antes de que el trabajo fuese definitivamente calificado por el profesor.

En caso de detección de fallos, el propio abogado del diablo podría optar por corregirlos él, o solicitar a su compañero que lleve a cabo las acciones correctivas pertinentes.

Una vez que el trabajo realizado recibe la aprobación del abogado del diablo, se pasaría a la siguiente iteración en la realización del trabajo. En la siguiente iteración, los estudiantes se intercambian los papeles, siendo el abogado del diablo el que implementa el siguiente bloque de código; y el que implementó anteriormente pasa a ejercer ahora el papel de abogado del diablo.

Destacar que si el número de bloques en los que se ha dividido el trabajo fuese impar no habría problema alguno dado que la carga de trabajo por bloque asignada al que implementa y al abogado del diablo es aproximadamente la misma. Es decir, aunque al principio a los estudiantes le pueda parecer lo contrario, el abogado del diablo trabaja igual, o algo más incluso, que el que implementa, siendo por tanto la distribución del trabajo equitativa.

Para que esta metodología tenga éxito, el abogado del diablo debe criticar, de forma racional y constructiva, pero con firmeza y exhaustividad, el trabajo de su compañero. El compañero debe tomarse estas críticas no como un ataque personal, sino como un trabajo cuyo único objetivo es mejorar el resul-

¹ Disponible en <http://personales.unican.es/sanchezbp/teaching/faqs/programming.html>

tado global del equipo. Si el compañero que es inspeccionado por el abogado del diablo detectase que después del trabajo del abogado quedasen aún existiesen defectos que deban ser reparados, deberá instar al abogado a repetir su trabajo, pero esta vez con mayor diligencia y profesionalidad.

La siguiente sección describe los beneficios obtenidos al aplicar esta metodología.

4. Beneficios Esperados

Los beneficios derivados de la aplicación de la metodología del abogado del diablo se resumen en los siguientes puntos, los cuales desarrollaremos a lo largo de esta sección:

1. Los alumnos pueden trabajar de forma individual a la vez que realizan un trabajo cooperativo.
2. Permite la colaboración no presencial entre alumnos.
3. Los estudiantes aprenden de los errores de sus compañeros, es decir, se produce en parte un *aprendizaje por observación y experimentación*.
4. Fomenta el espíritu crítico del alumno, que aprende a autoevaluarse, competencia necesaria para alcanzar el aprendizaje autónomo.
5. Los alumnos, mediante la experimentación, descubren la existencia de ciertos problemas relacionados con la Ingeniería del Software, tales como el control de versiones o el diseño de pruebas.
6. La percepción y valoración del estudiante del trabajo cooperativo mejora.

4.1. Trabajo individual y cooperativo

De acuerdo con los principios de nuestra metodología, los alumnos trabajan de forma individual pero coordinada. Por su propia construcción, la metodología evita el que haya dos alumnos sentados frente a un mismo computador, uno trabajando y el otro mirando. En la metodología del abogado del diablo, ambos estudiantes se convierten en elementos productivos del grupo.

Mencionar que esta forma de trabajar en equipo es la que se corresponde con la práctica diaria de las empresas de desarrollo software. En tales empresas,

un director de proyecto coordina el trabajo de varios desarrolladores. Estos desarrolladores trabajan de forma individual sin que exista excesiva comunicación entre ellos. Es decir, se trata de un trabajo cooperativo pero autónomo.

Podría también plantearse como solución para convertir a los dos estudiantes en elementos productivos que, tras dividir el código a desarrollar en bloques, cada alumno implementase un bloque de forma individual y luego los integrasen. Debido a las dependencias normalmente existentes entre diferentes bloques, este desarrollo independiente se hace francamente complejo. Incluso en los casos donde es posible alcanzar cierto grado de independencia, los esfuerzos requeridos para integrar el código desarrollado de forma aislada por los estudiantes suelen ser bastante grandes y tediosos. Por tanto, el estudiante vuelve a tener la sensación de que podría haber completado el trabajo en menor tiempo si hubiese realizado la tarea de manera individual en lugar de colectiva.

Por tanto, consideramos que la metodología del abogado del diablo ayuda a mejorar percepción que poseen los alumnos sobre los beneficios del trabajo en equipo.

4.2. Coordinación no necesariamente presencial

Como consecuencia directa de que la metodología obligue a que los alumnos trabajen de forma individual pero coordinada, se elimina la necesidad de que los alumnos tengan que reunirse en un momento concreto en un espacio físico determinado para realizar el trabajo asignado.

Los alumnos sólo necesitan reunirse para decidir el tema del trabajo (si se les da la opción de elegir), dividir el trabajo en diferentes bloques y decidir quién implementa y quién es abogado del diablo en la primera iteración. Es conveniente también que los alumnos acuerden un calendario de trabajo, así como que se comprometan a cumplirlo. Tras esta reunión inicial, y dado los avances tecnológicos existentes hoy en día, los alumnos no tienen necesidad alguna de volver a reunirse.

En cada iteración, el alumno que implementa sólo necesita enviar el código creado al abogado del diablo una vez que ha terminado su trabajo. El abogado del diablo elabora entonces su informe de fallos,

corrige aquellos que considere triviales, y le vuelve a reenviar el código, si lo hubiese modificado, junto con el informe de errores a subsanar. El alumno que creó el código subsana los errores notificados y vuelve a enviar el código al abogado del diablo. Este último repite el proceso hasta que ambos estudiantes den el visto bueno al trabajo.

Por tanto, tener acceso al correo electrónico es suficiente para poder trabajar de forma cooperativa pero individualizada. Ello evita el que los alumnos se tengan que reunir para trabajar. Este hecho, según las opiniones, suele ser el primer obstáculo que los alumnos encuentran para trabajar en equipo.

4.3. Aprendizaje mediante la evaluación por pares

Se dice que el hombre es el único animal capaz de tropezar dos veces con la misma piedra. A pesar de ello, es conveniente permitir que los alumnos tropiecen. De esta forma, tras varios tropiezos, acaban por darse cuenta de dónde están las piedras y aprenden a evitarlas.

La metodología del abogado del diablo fomenta que los alumnos descubran dichas piedras y aprendan a evitarlas. Usando esta metodología, los alumnos descubren dónde fallan las aplicaciones de sus compañeros y cómo se solucionan esos fallos. A partir de ese momento, toman conciencia de la existencia de tales fallos y comienzan a adoptar las actitudes necesarias para evitarlos.

Un ejemplo muy claro es la operación de eliminación en listas enlazadas con punteros al principio y al final de dicha lista. La mayoría de los alumnos prueban el código desarrollado para listas vacías y listas con varios elementos, pero casi nunca para listas con un único elemento, donde dicho elemento sea además el elemento que hay que eliminar.

La metodología del abogado del diablo ayuda a que los alumnos descubran estos casos particulares. Esto es debido en gran parte al hecho de que el abogado del diablo sólo tiene que diseñar los casos de prueba, lo cual ayuda a que el alumno que desempeña el papel del abogado del diablo sea más exhaustivo diseñando casos de prueba, dado que es su única función y no se encuentra agotado o cansado por el esfuerzo de la implementación. Este cansancio o hastío es la razón principal que esgrimen los alumnos para no diseñar casos de prueba exhaustivos.

4.4. Aprender a autoevaluarse

El hecho de ejercer de abogado del diablo convierte temporalmente al alumno que desempeña dicho papel en evaluador de su compañero. Ello hace que para la siguiente iteración, el abogado del diablo asuma que va a ser evaluado de la misma forma que él ha sido evaluado. Por tanto, desarrolla el código evitando cometer los mismos errores que él ha detectado en el trabajo de su compañero. Es decir, el alumno aprende a autoevaluarse, a detectar sus propios errores y a corregirlos. Por tanto mejora sus habilidades y competencias de manera autónoma. Es decir, se fomenta el *aprendizaje autónomo* y el alumno aprende a aprender.

4.5. Competencias relacionadas con la Ingeniería del Software

La aplicación de la metodología del abogado del diablo prepara al alumno para una mejor adquisición de competencias tradicionalmente relacionadas con la Ingeniería del Software, materia que suele impartirse en un tercer o cuarto curso. Esto se debe a que el alumno experimenta y sufre mediante la aplicación de la metodología presentada algunos de los problemas que la Ingeniería del Software trata de solucionar. Una mejor comprensión del problema, y sobre todo el haberlo sufrido, permite al alumno entender mejor la necesidad de aprender y comprender las soluciones existentes a esos problemas. Es decir, el haber experimentado ciertos problemas y dificultades motiva al alumno a aprender y comprender cómo solucionarlos.

Dos claros ejemplos son la *gestión de versiones* [2] y la necesidad de disponer de *metodologías para la realización de pruebas sistemáticas* [9].

Uno de los primeros problemas con los que se enfrenta el abogado del diablo es el diseño de casos de prueba sistemáticos. Sin ayuda de una metodología adecuada, el abogado del diablo se encuentra la mayoría de las veces desamparado. Por tanto, el alumno toma conciencia de la existencia una rama de conocimiento dedicada al desarrollo sistemático de pruebas software. Consecuentemente, el alumno no sólo entiende la necesidad de su existencia, sino que además se encuentra motivado hacia su aprendizaje, al ser plenamente consciente de su utilidad práctica.

El mismo fenómeno acontece con la gestión de versiones. Al tener que compartir y modificar el mismo código fuente la pareja de estudiantes, aparecerán los inevitables problemas de versiones. Es por ello que el alumno, al sufrir y experimentar el problema, entenderá y valorará mejor la necesidad de disponer de metodologías para la gestión del cambio y la configuración, así como la necesidad de disponer de herramientas como *Subversion* que sirvan para automatizar parte de estas tareas.

4.6. Valoración del trabajo en equipo

Al ser los dos miembros del grupo elementos productivos con escasa necesidad de coordinación, los alumnos perciben que el hecho de trabajar en equipo aumenta la productividad. Como consecuencia, los alumnos valoran los resultados obtenidos como de mayor calidad que los que podrían haber obtenido trabajando a nivel individual, y por tanto descubren las ventajas de trabajar en equipo.

La siguiente sección analiza las argumentaciones de esta sección a partir de los resultados obtenidos, las opiniones recogidas entre el alumnado y una comparación con los cinco elementos clave que debe poseer todo aprendizaje cooperativo.

5. Discusión de los Resultados Obtenidos

Esta sección discute los resultados obtenidos a partir de las opiniones y sensaciones recopiladas entre el alumnado acerca de su percepción de trabajo cooperativo antes y después de aplicar la metodología propuesta en este artículo. Se analiza también cuándo la metodología propuesta satisface los 5 elementos clave del aprendizaje cooperativo [4].

En general, la mayoría de los alumnos comentaban al principio del cuatrimestre que el trabajo en equipo era a veces positivo, pues en ocasiones podían aprender algo de sus compañeros. Pero, por otra parte, les suponía normalmente un esfuerzo extra considerable debido a la dificultad de encontrar un horario y espacio determinados donde poder reunirse para trabajar juntos. Además, algunos alumnos, normalmente los que mejores calificaciones tenían, eran de la opinión de que el trabajo en equipo les perjudicaba más que les beneficiaba. Estos alumnos consideraban que podían llegar a los mismos resultados y de forma más rápida trabajando de forma in-

dividual que cooperativamente.

Al final del cuatrimestre las opiniones del alumnado acerca de la metodología del abogado del diablo eran mayoritariamente positivas. En especial, porque les permitía trabajar de forma coordinada, pero sin estar físicamente juntos ni trabajando al mismo tiempo. La percepción de los alumnos acerca del trabajo en equipo parecía haber mejorado notoriamente al final del cuatrimestre. Casi la totalidad de ellos, en mayor o menor grado, eran de la opinión de que el hecho de trabajar en equipo les había ayudado a obtener resultados de mejor calidad que los que hubiesen obtenido trabajando individualmente, incluyendo a los alumnos con mejores calificaciones. Este tipo de alumno comentaba que la metodología les ayudaba a mejorar su trabajo, dado que sus compañeros detectaban errores que a ellos se les habían inicialmente escapado. Por otra parte, cuando ejercían de abogados del diablo se sentían mucho más cómodos a la hora de criticar, siempre de forma constructiva, el trabajo de sus compañeros. Comentaban que, estas mismas críticas, cuando se hacían usando otra metodología, no eran siempre recibidas con agrado por el resto del equipo, mientras que ahora, las mismas críticas, si eran bien recibidas.

Como aspectos negativos de la metodología, la mayoría de los alumnos comentaban que les resultaba difícil diseñar casos de prueba exhaustivos, principalmente porque no sabían que pasos debían seguir, qué casos considerar, o cuántos casos eran suficientes. Por tanto, nos proponemos como mejora a esta metodología formar a los alumnos de los próximos cursos en el diseño sistemático de casos de prueba. De este modo el alumno que ejerza el papel de abogado del diablo, podrá hacerlo de manera más efectiva, sistemática y científica.

Por último, analizamos como la metodología del abogado del diablo cumple con los 5 aspectos clave del trabajo cooperativo: (1) interdependencia positiva; (2) interacción positiva cara a cara; (3) exigibilidad individual; (4) habilidades cooperativas; y (5) autoanálisis del grupo.

Interdependencia positiva A tenor de las opiniones recogidas entre el alumnado, el alumno entiende que el hecho de pertenecer a un equipo le beneficia, por lo que considera el hecho de trabajar de forma cooperativa como algo positivo. El fomentar la interdependencia positiva era uno de los objetivos prin-

cipales que perseguíamos con la introducción de esta metodología.

Interacción positiva personal En este aspecto la metodología del abogado del diablo ayuda a mejorar la percepción que los alumnos tienen de sus compañeros. Obsérvese por ejemplo que los alumnos con mejores calificaciones mejoraron la percepción que tenían de sus compañeros menos capaces. De la misma forma, los menos capaces aprendieron a aceptar las críticas de los alumnos más aventajados.

Exigibilidad individual Tal como planteamos la metodología, cada alumno tiene una tarea bien definida y ha de responder con su trabajo ante su compañero. Si alguno de los dos miembros de un grupo no trabaja, queda patente y manifiesto quién es el que no trabaja. No obstante, la metodología no incluye ningún punto específico acerca de cómo resolver los conflictos, en caso de que se produzcan, o de cómo evitar que se produzca ningún tipo de *parasitismo*.

Habilidades cooperativas La metodología del abogado del diablo es en sí una técnica de trabajo cooperativo que los alumnos aprenden mediante su aplicación. Por tanto, mejoran su competencia de trabajo cooperativo mediante la incorporación a su saber y habilidades de una nueva técnica de trabajo cooperativo, en mi humilde opinión, de gran utilidad en el ámbito del desarrollo software. La metodología además no es incompatible con la formación en otras habilidades cooperativas, como la toma de decisiones o la gestión de conflictos.

Autoanálisis del grupo En nuestro caso particular, no ha existido ningún tipo de autoanálisis de los grupos, a excepción de la propia de final del cuatrimestre, con objeto de conocer las opiniones del alumnado acerca de la técnica aplicada. No obstante, la técnica no es incompatible con el autoanálisis. Es más, al estar las funciones de cada miembro del grupo claramente identificadas, es más fácil comprobar si cada uno está cumpliendo su función y si dicha función es eficaz para alcanzar un objetivo determinado.

6. Conclusiones

El presente artículo ha presentado una metodología de trabajo en equipo que resulta especialmente adecuada para las materias relacionadas con la Programación de Computadoras que se imparten en los primeros cursos de las titulaciones universitarias. Las principales ventajas de esta metodología es que mejora la visión que los alumnos tienen del trabajo cooperativo, al servir para incrementar su productividad y desarrollar productos de mayor calidad.

Referencias

- [1] Kent Beck. *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional, October 1999.
- [2] Stephen P. Berczuk. *Software Configuration Management Patterns: Effective Teamwork, Practical Integration*. Addison-Wesley Professional, November 2003.
- [3] Erich Gamma, Ralph Johnson Richard Helm and, and John M. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, November 1994.
- [4] David W. Johnson, Roger T. Johnson, and Karl A. Smith. *Active Learning: Cooperation in the College Classroom*. Interaction Book, December 1991.
- [5] Thomas Kelley and Jonathan Littman. *The Ten Faces of Innovation: IDEO's Strategies for Defeating the Devil's Advocate and Driving Creativity Throughout Your Organization*. Doubleday, Diciembre 2005.
- [6] Bertrand Meyer. *Touch of Class: Learning to Program Well with Objects and Contracts*. Springer, 2009.
- [7] Mercedes Marzo Navarro, Marta Pedraja Iglesias, and Pilar Rivera Torres. Las competencias profesionales demandadas por las empresas: el caso de los ingenieros. *Revista de Educación*, 341:643–661, Septiembre-Diciembre 2006.
- [8] Michael A. West. *Effective Teamwork: Practical Lessons from Organizational Research*. Wiley-Blackwell, December 2003.
- [9] Colin Willcock, Thomas Deiss, Stephan Tobias, Stefan Keil, Federico Engler, and Stephan Schulz. *An Introduction to TTCN-3*. Wiley, July 2005.