

# Métodos formales en programación: ¿desmitificar para motivar?

Inés Jacob

Facultad de Ingeniería  
Universidad de Deusto  
Apartado 1, 48080 Bilbao  
e-mail: [ines@eside.deusto.es](mailto:ines@eside.deusto.es)

## Resumen

Los métodos formales gozan de una imagen entre alumnos y profesores que plantea dificultades en el aprendizaje y la docencia de su aplicación a la programación. Son reconocidas las dificultades de aplicar en la industria mediante métodos formales los razonamientos matemáticos que son abstractos por naturaleza. Se presentan en este trabajo algunas reflexiones que tienen como resultado la identificación de la necesidad de encontrar fuentes de motivación para los alumnos. Se propone la incorporación de actividades en la preparación y docencia de asignaturas que tratan métodos formales que contribuyan a su innovación pedagógica.

## 1. La experiencia docente

Las reflexiones que en este artículo presentamos vienen provocadas por la experiencia docente de la asignatura Metodología de la Programación impartida en la Universidad de Deusto en segundo de Ingeniería Técnica de Informática de Gestión.

Se basa en los conocimientos de lógica formal y de programación que el alumno adquiere en primero. Es la primera asignatura de la titulación en la que se trata la aplicación de métodos formales al desarrollo de software.

Con el fin de:

- cubrir el contenido fijado en el plan de estudios para esta asignatura troncal,
- completar el itinerario del alumno por las asignaturas del área de Programación y Lenguajes,
- fijar y afianzar las bases para el aprendizaje de contenidos de disciplinas relacionadas y abordadas en parte en cursos posteriores.

se fijan los siguientes objetivos generales para la asignatura:

- familiarizarse con la verificación y derivación formal de programas
- razonar formalmente el diseño recursivo de funciones
- entender la recursividad y su relación con los programas iterativos.

Los objetivos específicos han sido elegidos en función de estos objetivos generales de forma que el alumno identifique en ellos los logros que ha de conseguir a lo largo de la asignatura. Coinciden con los aspectos que serán evaluados.

Este diseño de objetivos [14], da lugar al programa de la asignatura, dividido en seis capítulos que tratan:

1. La motivación para el estudio de métodos formales en programación y los conceptos básicos empleados durante el curso.
2. El lenguaje utilizado, con la descripción de los métodos de verificación y derivación de cada una de las instrucciones.
3. El principio de inducción aplicado a la demostración de teoremas inductivos y a la validación de bucles y funciones recursivas.
4. La derivación de programas recursivos.
5. La transformación de programas por inmersión.
6. La derivación de funciones iterativas.

El equipo docente de la asignatura considera que este diseño puede permanecer estable, en líneas generales, durante un tiempo. Tras varios años dedicados a su ajuste estima por lo tanto que es el momento de hacer una reflexión crítica y constructiva sobre la asignatura.

## 2. Sobre la motivación

Las calificaciones obtenidas por los alumnos de Metodología de la Programación no son muy

buenas, siendo además bastante elevado el porcentaje de alumnos matriculados que ni siquiera se presentan a examen.

Se plantea como prioritario en este momento mejorar los resultados académicos obtenidos en esta asignatura, incidiendo en la motivación del alumno como motor de su aprendizaje.

Las dos siguientes consideraciones justifican la necesidad de proponerse que el alumno esté motivado como medio para mejorar su rendimiento:

- La motivación es la responsable de iniciar, mantener y dirigir la conducta de un individuo hacia la consecución de una meta u objetivo [18].
- La calificación constituye un indicador del grado de consecución de los objetivos planteados en una asignatura, siempre y cuando las herramientas y métodos de evaluación sean adecuados.

Centrar nuestro interés en la motivación no supone ignorar la importancia que los métodos y herramientas de evaluación tienen. Somos plenamente conscientes de que no es trivial lograr una evaluación que se ajuste realmente a los objetivos de la asignatura y valore el proceso desarrollado por el estudiante reconociéndolo en la calificación final [15].

En cualquier caso la motivación y la evaluación, no deberían ser consideradas independientemente dadas las influencias mutuas existentes entre ambas.

Los estudiantes no motivados no aprenden [16] y por lo tanto los profesores deben encontrar la forma de motivar a sus alumnos. Pensar en los conceptos castigo, recompensa e incentivo relacionados con la motivación extrínseca no es suficiente y se hace necesario estudiar las vías para aumentar la motivación intrínseca. Se trata de lograr que los alumnos tengan interés por la asignatura, pasando si es necesario por un estado anterior de curiosidad.

### 3. Posibles orígenes de las dificultades

Nos parece fundamental reflexionar sobre las particularidades de la asignatura, que se derivan de que los métodos explicados y aplicados son formales.

#### 3.1. Las matemáticas y la informática

La definición "Método formal es cualquier técnica que trate la construcción y/o el análisis de modelos matemáticos que contribuyen a la automatización del desarrollo de sistemas informáticos" [10] podría ser útil para explicar a los estudiantes el sentido de estudiar métodos formales, para contestar al tan frecuente "para qué sirve esto".

Quedaría además patente la relación de la asignatura con las matemáticas, con lo que el anclaje de esta asignatura en otras de cursos precedentes quedaría explícito.

El estudiante de informática debería conocer los motivos por los que las matemáticas son importantes en su formación [12]:

1. El software es abstracto.
2. El uso preciso de notaciones y símbolos es propio tanto de las matemáticas como de la ingeniería del software.
3. Las matemáticas son necesarias para modelar sistemas de software.
4. Muchos dominios de aplicación (ingeniería ciencia, economía, etc.) tienen bases matemáticas.
5. Es esencial aplicar razonamiento matemático a los sistemas informáticos.

La informática es una ciencia matemática, abstracta. De hecho, los estudiantes, aunque a veces no son muy conscientes de ello, comienzan sus estudios utilizando lenguajes formales (los lenguajes de programación tienen una sintaxis y una semántica formalmente definida) y utilizando fórmulas matemáticas (programas) para resolver problemas con un ordenador [1].

#### 3.2. Los métodos formales y la programación

En el conjunto de asignaturas que los alumnos cursan en el área de programación, Metodología de la Programación se distingue de las demás porque

- los programas obtenidos no pueden probarse por ejecución, pues se utiliza un pseudocódigo,
- contradice en cierta forma el modo en el que los estudiantes han aprendido a programar en su primer año de carrera,
- los métodos que se explican no se aplican habitualmente en otras asignaturas.

Estas características no sirven precisamente de refuerzo para el estudiante que, atendiendo a los resultados obtenidos de las encuestas realizadas para evaluar la docencia, considera que esta asignatura es una de las más difíciles.

Esta asignatura parece quedar así aislada de sus compañeras del área de programación cuando en realidad tiene un puente entre las matemáticas y la programación tradicional, apoyándose en los conocimientos que de esas materias los alumnos han adquirido en el curso precedente (Figura 1).

Creemos necesario que los estudiantes puedan apreciar realmente los vínculos entre la programación y las matemáticas tradicionales a través de la aplicación de los métodos formales a la programación.

Aunque en los planes de estudio de informática se da importancia a las matemáticas, normalmente aparecen aisladas, no aplicadas al desarrollo de software [2].

Algunos de los contenidos de las asignaturas de primer curso: lógica (implicación, negación,

cuantificación, principio de inducción), teoría de conjuntos (unión, intersección, pertenencia), álgebra de Boole (and, not, or, leyes de Morgan), se aplican directamente en Metodología de la Programación.

Sería conveniente [17] que estos conceptos se explicaran aplicados a la programación o a otros campos de la ingeniería del software.

Si esto no fuera posible habría que ver la manera de establecer el vínculo desde 2º curso hacia esas asignaturas del curso precedente.

Los estudiantes tienen presente desde el principio la relación con las asignaturas de programación, pero se hace necesario ahondar en los vínculos que se pueden establecer con asignaturas posteriores. Sería así mismo deseable transcribir algunos de los programas obtenidos en pseudocódigo a alguno de los lenguajes de programación conocidos por los estudiantes.

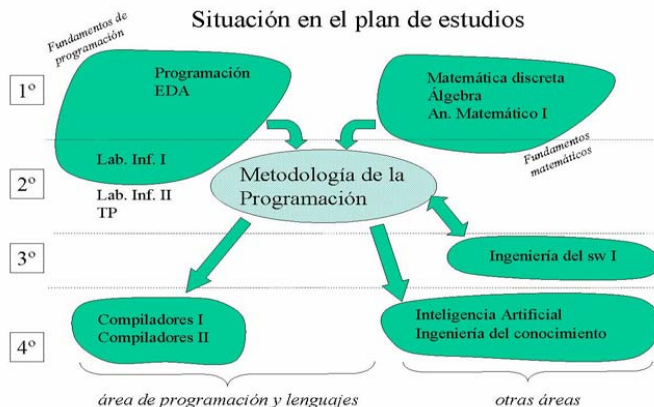


Figura 1. Situación de la asignatura Metodología de la Programación en los planes de estudios de Ingeniero Técnico en Informática de Gestión y de Ingeniero en Informática en la Universidad de Deusto

### 3.3. Aplicaciones en la industria

El uso de los métodos formales en la industria no está tan extendido como sería deseable desde el punto de vista de sus defensores. Se han estudiado las causas [7][8][9][13] de esta situación, entre las que señalamos:

- El desarrollo de herramientas que apoyen la aplicación de métodos formales es complicado y los programas resultantes son incómodos para los usuarios.
- Los investigadores por lo general no conocen adecuadamente la realidad industrial.
- Es escasa la colaboración entre la industria y el mundo académico, que en ocasiones se muestra demasiado dogmático.
- Se considera que la aplicación de métodos formales encarece los productos y ralentiza su desarrollo.

Parece claro que los métodos formales se implantarán en la industria a través de nuevos profesionales con sólidos conocimientos de las técnicas matemáticas [10].

Luego el cambio deberá venir desde el mundo académico que tendrá que hacerse eco tan pronto como sea posible de los avances de las técnicas formales en la industria. Ésta ya hace algún tiempo que con esperanza se interesa por los métodos formales como herramienta para asegurar la calidad de productos cuyo desarrollo se complica cada día.

Además ya existen productos y estándares ampliamente aceptados con un componente matemático y formal importante en su diseño [5].

## 4. Mitos y mandamientos

Los mitos y mandamientos sobre los métodos formales constituyen dos aproximaciones clásicas a las particularidades de tales métodos.

### 4.1. Los catorce mitos sobre los métodos formales.

Podríamos decir que a los métodos formales "la fama les precede". Los conocidos 14 mitos sobre los métodos formales (7 mitos [11] y 7 más [3]) recogen ideas preconcebidas sobre ellos:

1. Los métodos formales garantizan la perfección del software y hacen innecesaria su verificación.
2. Los métodos formales sólo sirven para demostrar que los programas son correctos.
3. Los métodos formales sólo es necesario aplicarlos en sistemas donde la seguridad es crítica.
4. Los métodos formales sólo pueden ser aplicados por expertos en matemáticas.
5. La aplicación de métodos formales aumenta los costes de desarrollo.
6. Los métodos formales no son bien vistos por los usuarios.
7. Los métodos formales no se aplican en sistemas reales de gran tamaño.
8. Los métodos formales retrasan el desarrollo.
9. No existen herramientas informáticas de apoyo a la aplicación de los métodos formales.
10. Los métodos formales implican la renuncia a los métodos tradicionales de diseño.
11. Los métodos formales sólo sirven para diseño de software, no para diseño de hardware.
12. Los métodos formales no son necesarios.
13. Los métodos formales no son sustentados por la industria y los investigadores.
14. Los defensores de los métodos formales sólo utilizan métodos formales.

Estos mitos ofrecen una imagen verdaderamente negativa de los métodos formales pero es previsible que sean aceptados como ciertos por los alumnos que se acercan por primera vez a la aplicación de métodos formales a tareas propias de informáticos.

En la Tabla 1 recogemos los resultados de una pequeña encuesta realizada para conocer el calado que estos mitos tienen entre los alumnos.

Participaron en el sondeo 45 alumnos de Metodología de la Programación que calificaron como cierto o falso cada uno de los enunciados que aparecen listados más arriba.

Para algunos de los mitos el porcentaje de alumnos que lo consideran cierto no es muy elevado. Hay que tener en cuenta que en el momento de realizar la encuesta los métodos formales ya habían sido explicados y aplicados en clase por lo que es muy posible que ya se hubiera producido una cierta desmitificación.

Mito nº	% de alumnos que lo consideran <i>cierto</i>	% de alumnos que lo consideran <i>falso</i>
1	56%	44%
2	58%	42%
3	4%	96%
4	4%	96%
5	44%	56%
6	60%	40%
7	40%	60%
8	60%	40%
9	58%	42%
10	29%	71%
11	87%	13%
12	33%	67%
13	31%	69%
14	7%	93%

Tabla 1. Mitos de los métodos formales entre los estudiantes

#### 4.2. Los diez mandamientos

Los diez mandamientos [4] sobre los métodos formales sugieren

1. Elegir una notación apropiada.
2. No formalizar en exceso.
3. Hacer una buena estimación de costes.
4. Tener disponible un conocedor de los métodos.
5. No abandonar los métodos tradicionales.
6. Documentar suficientemente.
7. Alcanzar los estándares de calidad.
8. No ser dogmático.
9. Repasar una y otra vez.
10. Reutilizar.

Pueden ser considerados como una serie de recomendaciones a tener en cuenta durante el diseño y la docencia de asignaturas que incluyen la aplicación de métodos formales como algo central.

Tener presentes estos consejos es también conveniente para intentar dar solución a los problemas de implantación de los métodos formales en la industria.

#### 5. Propuesta de mejora

Hacemos nuestra propuesta con el fin de procurar la mayor motivación de los estudiantes de la asignatura Metodología de la Programación.

Pensamos que los mitos sobre los métodos formales [3][11] dan de ellos una imagen negativa que no se justifica y sospechamos que estos mitos son considerados realidades por nuestros estudiantes. Proponemos una serie de actividades orientadas a desmitificar los métodos formales bajo la hipótesis de que el lograrlo puede hacer que el alumno se sienta más motivado.

Tras haber estudiado algunos de los condicionantes del proceso de enseñanza-aprendizaje de los métodos formales y a la luz de los diez mandamientos relativos a su aplicación [4], proponemos para el equipo docente de la asignatura las siguientes acciones:

1. Organizar una actividad en grupo a principio de curso en la que los alumnos debatan los mitos de los métodos formales.
2. Analizar los temarios de las asignaturas que en el futuro los estudiantes cursarán para tener ocasión de establecer vínculos concretos con ellas.
3. Localizar, en cursos anteriores, los momentos concretos en los que los alumnos aprenden conceptos básicos empleados en la aplicación de métodos formales, considerando la posibilidad de proporcionar a los profesores de esas asignaturas ejemplos o comentarios que hagan referencia a la nuestra.
4. Documentar para los estudiantes la afirmación de que la informática, como ciencia, tiene sus bases en las matemáticas.
5. Informar a los alumnos sobre aplicaciones de los métodos formales en la industria.
6. Recurrir durante las explicaciones a programas ya desarrollados por los alumnos aplicando los nuevos métodos sólo en algunos fragmentos del código, de forma que el resto sea considerado de utilidad.
7. Presentar algoritmos de cierta complejidad desarrollados formalmente, aunque no se espere de los alumnos que sean capaces de obtener soluciones parecidas. Permitirá observar que los métodos no sólo se aplican a problemas triviales.

8. Transcribir algunos programas obtenidos a un lenguaje de programación conocido por los alumnos, para bajar el nivel de abstracción.
9. Repetir la encuesta sobre el calado de los mitos entre los estudiantes, a principio y a final de curso. Podremos así observar la evolución de la visión que los alumnos tienen de los métodos formales.

En esta asignatura ya se ha probado con éxito la realización de actividades en grupo, que son recibidas con notable entusiasmo por los estudiantes.

Se han hecho también referencias a programas desarrollados en C o Pascal y se ha podido observar que tratar con los lenguajes que conocen da confianza a los estudiantes.

En el diseño de actividades y presentación de documentos complementarios en clase hay que tener en cuenta que normalmente los alumnos consideran que saben programar y que no dominan las matemáticas. Es positivo aprovechar su confianza en sus habilidades como programadores evitando presentar los métodos formales como algo que las hace prescindibles y ayudarles a superar su complejidad en lo que se refiere a las matemáticas.

## 6. Conclusión

Somos conscientes de la dificultad de la tarea de incidir en la motivación de una persona, sobre todo si prescindimos de los conceptos de recompensa, castigo e incentivo propios de la motivación extrínseca más fácilmente modulable desde el exterior del individuo. Sin embargo la motivación es el desencadenante del esfuerzo necesario para el aprendizaje y debe ser por lo tanto tenida en cuenta.

## Referencias

- [1] Almstrum, V.L., Dean, C.N., Goelman, D., Hilburn, T.B., Smith, J. *Support for teaching formal methods*, ACM SIGCSE Bulletin, Volume 33, Issue 2, June 2001.
- [2] Baldwin, D., Henderson P.B., *The importance of mathematics to the software practitioner*, IEEE Software, 19(2): 110-112, March-April, 2002.
- [3] Bowen, J.P., Hinchey, M.G., *Seven more myths of formal methods*, IEEE Software, 12(4):34-41, July 1995.
- [4] Bowen, J.P., Hinchey, M.G., *Ten Commandments of Formal Methods*", IEEE Computer, 28(4):56-63, April 1995.
- [5] Bruce, K.B., Drysdale R.L.S., Kelemen C., *Why Math?* En preparación para Communications of the ACM, 2002. <http://brastias.cs.geneseo.edu/~baldwin/math-thinking/CACM4.pdf>
- [6] *Computing Curricula 2001, Computer Science*. The Joint Task Force on Computing Curricula, IEEE Computer Society, ACM.
- [7] Craigen, D., Gerhart, S., Ralston, T., *An International Survey of Industrial Applications of Formal Methods*, Technical Report, National Institute of Standards and Technology, Nistcr 93/626. 1993.
- [8] Craigen, D., Gerhart, S., Ralston, T., *Formal Methods Reality Check: Industrial Usage* IEEE Transactions on Software Engineering, 21 (2): 90-98, 1995.
- [9] Crocker, D., *Making formal methods popular through automated verification*. International Joint Conference on Automated Reasoning, Siena, Italy, June 2001.
- [10] Gibson, P., Méry, D., *Teaching Formal Methods: Lessons to Learn*" BCS electronic workshops in computing, ISSN: 1477-9358, Butterfield and Flynn, 1998.
- [11] Hall, A., *Seven myths of formal methods*. IEEE Software, 7(5):11-19, September 1990.
- [12] Henderson, P., *Mathematical Reasoning in Software Engineering Education*, en preparación para Communications of the ACM. <http://brastias.cs.geneseo.edu/~baldwin/math-thinking/MathInSE.PDF>
- [13] Holloway, C.M., Butler, R.W., *Impediments to industrial use of formal methods*, IEEE Computer: 25-26, April 1996.
- [14] Jacob, I., *Metodología y herramientas para el aprendizaje de la verificación y derivación formal de programas*, Jenui 2000 (Libro de comunicaciones de las VI Jornadas sobre la Enseñanza Universitaria de la Informática) ISBN: 84-138-409-7.
- [15] Martínez, J.R., Galán, F. *Estrategias de aprendizaje, motivación y rendimiento académico en alumnos universitarios*. Revista

- Española de Orientación y Psicopedagogía. 11, (19), 35 – 50. 2000.
- [16] Mateo, M. *La motivación, pilar básico de todo tipo de esfuerzo*. Proyecto social: revista de relaciones laborales. - N.9 (nov. 2001), p. 163-184.
- [17] Page, R., *Formal Methods Education and Programming Effectiveness*, Beseme Project, <http://www.cs.ou.edu/~beseme/> (2000-2003)
- [18] Wittrock, M.C., *Procesos del pensamiento de los alumnos*. En autor (ed). La investigación de la enseñanza III: Profesores y alumnos. 544 - 585. Barcelona: Paidós. 1986.