

# Un Nuevo Paradigma de Enseñanza para la Disciplina de Computación

Juan Alvarez Rubio

Departamento de Ciencias de la Computación

Universidad de Chile

Blanco Encalada 2120, Santiago, Chile

uucp: jalvarez@dcc.uchile.cl - BITNET: jalvarez@uchcecvm

## Resumen

La reciente definición de la computación como disciplina científica, matemática y de ingeniería ha inducido un nuevo paradigma de enseñanza. El anterior énfasis en la programación de computadores está siendo reemplazado por un tratamiento general de todas las subáreas temáticas de la disciplina. Este nuevo enfoque se está aplicando en la formación de especialistas, de ingenieros de otras áreas, y en la educación secundaria.

**Keywords:** Educación en Computación, Informática Educativa, Curriculum ACM-IEEE/CS.

## 1 Introducción

A fines de 1988 se produjo un hito histórico en el área de Computación. Un grupo de tarea conjunto de las sociedades científicas americanas más importantes del área, la ACM (Association for Computing Machinery) y la Sociedad de Computación de la IEEE (Institute of Electric and Electronic Engineers), publicó el informe *Computing as a Discipline* [10] que, después de más de 40 años de investigación y desarrollo informático, propone un marco que intenta definir y delimitar el quehacer computacional.

En primer lugar, la computación se considera ahora una disciplina, por cuanto sus dimensiones teóricas, de abstracción o modelamiento, y de diseño provienen de las matemáticas, de las ciencias experimentales, y de la ingeniería, respectivamente. De esta manera, se supera la polémica respecto de intepretar y reducir la computación a uno sólo de los aspectos anteriores.

En segundo lugar, se propone una breve definición de la disciplina al considerarla “el estudio sistemático de los procesos algorítmicos que describen y transforman información: su teoría, análisis, diseño, eficiencia, implementación, y aplicación“. La pregunta fundamental que subyace en la computación es ¿Qué podemos automatizar (eficientemente)?.

Finalmente, se plantea una subdivisión de la disciplina en nueve subáreas temáticas, cada una con sus propias bases teóricas, sus propios procesos de diseño y abstracción, y una comunidad de investigadores y publicaciones: Algoritmos y Estructuras de Datos, Lenguajes de Programación, Arquitectura, Computación Numérica y Simbólica, Sistemas Operativos, Ingeniería y Metodología de Software, Sistemas de Bases de Datos y de Recuperación de Información, Inteligencia Artificial y Robótica, y, Comunicación Humano-Computador (que incluye Computación Gráfica e Interfaces con el Usuario).

## 2 Enseñanza de la Disciplina de Computación

El marco anterior sugiere un cambio significativo del paradigma de enseñanza. El antiguo enfoque que se iniciaba y centraba en la programación, debe ser reemplazado por un tratamiento horizontal que, introduciendo todas las subáreas temáticas, permita obtener una visión panorámica de la disciplina. La verticalidad o profundización se reserva a cursos especializados más avanzados.

El esquema anterior, si bien estuvo motivado por la necesaria renovación de los programas de estudios de los especialistas, sin embargo puede extenderse a distintas audiencias, haciendo las adaptaciones que resulten pertinentes. A continuación revisaremos la situación en diferentes contextos educacionales.

### 2.1 Especialistas en computación

#### 2.1.1 Curriculum 1991

En el informe *Computing Curricula 1991* [5] se propone un nuevo marco para la definición de programas de pregrado en computación. A la subdivisión de nueve sub-áreas, se agregó el contexto social y profesional de la disciplina. Adicionalmente se distinguen doce conceptos “recurrentes”, que están presentes tanto en las sub-áreas como en las dimensiones teórica, de abstracción y de diseño: asociación (binding), complejidad de problemas grandes, modelos conceptuales y formales, consistencia y completitud, eficiencia, evolución, niveles de abstracción, ordenamiento espacial, ordenamiento temporal, reusabilidad, seguridad, balances y consecuencias.

Para efectos de la estructuración curricular se define un conjunto de 55 unidades de conocimiento que subdividen los contenidos de las nueve (más un) subáreas temáticas, conformando un núcleo mínimo común a todos los programas de pregrado. La idea es que estas unidades sirvan como base para estructurar cursos específicos, combinándolas y orientándolas de acuerdo a los objetivos, prioridades y restricciones de cada institución educacional. Por supuesto, este núcleo debe ser complementado con requerimientos de matemáticas, ciencias, y material adicional y suplementario de computación.

Los lineamientos generales recomiendan que los programas de pregrado aseguren tanto amplitud como profundidad. Amplitud, en el sentido de cubrir todas las áreas temáticas. La profundidad se debe reflejar enfatizando aquellas áreas y procesos acordes con la orientación y especialización del programa. Por su parte, el rol de la programación, no se reduce a la codificación en un determinado lenguaje o arquitectura, sino que se amplía a todas las actividades que involucra la solución algorítmica de problemas, presente en todas las sub-áreas. Adicionalmente, y en consistencia con la dimensión científica de la disciplina, se recomienda un novedoso enfoque para el trabajo de laboratorios. Al respecto, se sugieren actividades de laboratorio tanto abiertas como cerradas. Estas últimas de naturaleza semejante al trabajo práctico y experimental de otras especialidades, con sesiones guiadas de realización de trabajos específicos.

### 2.1.2 Cursos introductorios

Resulta particularmente importante el diseño del primer curso para los especialistas en computación. Considerando que la propuesta curricular anterior permite diversas alternativas, es conveniente revisar algunas implementaciones y proposiciones específicas.

El curso *Foundations of Computer Science* de la State University of New York at Stony Brook [12] está diseñado para superar la debilidad de los alumnos en matemáticas discretas y en resolución de problemas, y para ilustrar los fundamentos matemáticos de la computación. Por ejemplo, los principios de la lógica se usan para entender la programación lógica. Por su parte, las funciones son la teoría que subyace en los lenguajes aplicativos o funcionales. El aprendizaje es reforzado a través de módulos de laboratorio computacionales.

En la misma línea anterior, David Gries propone mezclar en un curso de dos semestres el primer curso de programación y el segundo de matemáticas discretas [11]. De esta manera en cada etapa se enseña la teoría y la aplicación correspondiente. Concretamente se enseñarían los estilos funcional y algorítmico de programación. Primero el funcional, pasando de la inducción matemática

a la manipulación de funciones recursivas. El enfoque imperativo entraría en escena al tratar la economía de tiempo y espacio.

## **2.2 Ingenieros de otras especialidades**

El nuevo marco, representado tanto por la definición de la disciplina como la proposición curricular, está influenciando y modificando la enseñanza de la computación en los distintos niveles y para diferentes audiencias. En primer lugar, se están rediseñando los primeros cursos para los estudiantes de ingeniería (especialistas y no-especialistas), de modo de cambiar el énfasis en programación hacia la introducción de los conceptos fundamentales de cada una de las áreas temáticas [6].

Recomendaciones específicas sugieren un primer curso anual de *Introducción a la Programación y al Uso de Herramientas de Software* como una manera de presentar la disciplina y nivelar los eventuales conocimientos previos [8]. Cabe señalar que el contexto para tratar las herramientas es introducir las subáreas de Comunicación Humano-Computador, Sistemas de Bases de Datos, e Ingeniería de Software. Un segundo curso semestral [7] permite concluir el tratamiento de las subáreas temáticas.

En resumen, se apunta a entregar a los estudiantes de ingeniería una visión general y horizontal de la computación que permita comprender y asimilar los progresos tecnológicos presentes y futuros. La ventaja de este esquema es que puede ser completamente compartido con estudiantes de la especialidad de computación, los que profundizarán sus conocimientos en cursos más avanzados.

## **2.3 Enseñanza secundaria**

### **2.3.1 Proposición de un Nuevo Curso**

Recientemente, en noviembre de 1991, un grupo de tarea del consejo de educación de la ACM publicó un borrador de una proposición de un curso anual de computación para la educación secundaria [3]. Su alcance, profundidad, amplitud y metodología es similar a los cursos secundarios de otras asignaturas científicas, y consistentemente está destinado a todos los estudiantes, y no sólo a los interesados en carreras especializadas. Este esfuerzo complementa, actualiza, y proporciona otro enfoque, a la proposición previa de 1985 centrada en programación y las aplicaciones e implicaciones del uso de los computadores [1].

El curso propuesto sigue los lineamientos más recientes de la disciplina, incluyendo todas las subáreas temáticas organizadas de manera de adecuarse a estudiantes secundarios. La metodología

sigue el modelo de la enseñanza general de otras disciplinas científicas, basándose principalmente en laboratorios cerrados usando computadores y ejercicios a ser desarrollados sin computadores. Para el trabajo supervisado en laboratorio se deben diseñar guías que, teniendo objetivos y contenidos específicos, permitan a los estudiantes formular modelos (abstracciones) y entender la teoría subyacente. Los alumnos deben informar sus observaciones entregando informes escritos. Adicionalmente, se deben plantear ejercicios y proyectos que puedan ser desarrollados por los propios alumnos fuera de las horas de clases.

### 2.3.2 Preparación de Profesores

Las proposiciones curriculares para la preparación de profesores de computación en el nivel medio han evolucionado de acuerdo a las proposiciones de cursos para los alumnos. Así, la proposición de ACM para la certificación de profesores que pudieran enseñar los cursos propuestos en el año 1985 [2] se centraba consistentemente en la programación. Una posterior actualización de la propuesta anterior [4] se planteó explícitamente con una orientación hacia los fundamentos de la ciencia de la computación.

Una proposición que asimila tanto la nueva definición de la disciplina de computación, como la nueva proposición de curso para los estudiantes, recomienda al menos cuatro cursos en los programas de postgrado en Informática Educativa [9]: un curso de Introducción a la Resolución de Problemas y a la Programación; dos cursos de Introducción a las Subáreas de la Disciplina de Computación; un curso para tratar el Contexto Social, Etico y Profesional.

## 3 Conclusiones

Un nuevo paradigma de enseñanza de computación se ha consolidado en los últimos años. Derivado de la reciente definición de la computación como disciplina, su propósito es proporcionar una visión general de todas las subáreas temáticas, reemplazando el énfasis anterior en programación de computadores. Este nuevo enfoque se está aplicando en diversos contextos educacionales: formación de especialistas, enseñanza de computación para ingenieros, alumnos y profesores de computación en la enseñanza secundaria.

Respecto de la formación de especialistas, existe un documento con recomendaciones para diseñar currículums que se ajusten a estas nuevas ideas. Por su parte, los primeros cursos están siendo diseñados de manera que integren contenidos de matemáticas discretas y de computación.

La enseñanza de computación para estudiantes de ingeniería, si bien puede coincidir completamente con los cursos introductorios para los especialistas, pretenden entregar una visión general y horizontal de la disciplina.

A nivel de la educación secundaria, se pretende enseñar computación siguiendo el modelo de otras asignaturas científicas. El tratamiento panorámico global de la disciplina se basa en la realización de laboratorios computacionales cerrados y ejercicios y proyectos abiertos de mayor envergadura. Consecuentemente, es necesario renovar los planes de formación y actualización de los profesores.

En síntesis, el nuevo paradigma de enseñanza de la disciplina de computación comprende una perspectiva globalizadora que apunta a enseñar y aprender los conceptos fundamentales de todas las subáreas temáticas. De esta manera se pretende satisfacer el desafío de asimilar los desarrollos tecnológicos presentes y futuros del área.

## Referencias

- [1] Task Force on Curriculum for Secondary School Computer Science; *Computer Science for Secondary School: Course Content*; Communications of the ACM, Volume 28 Number 3, 270-274; March 1985
- [2] Task Force on Teacher Certification in Computer Science; *Proposed Curriculum for Programs Leading to Teacher Certification in Computer Science*; Communications of the ACM, Volume 28 Number 3, 275-279; March 1985
- [3] Task Force of the Pre-College Subcommittee of the Education Board of the ACM; *ACM Model High School Computer Science Curriculum*; Draft Report; November 1991
- [4] ACM/IEEE Task Force on Teacher Retraining; *Retraining Teachers to Teach High School Computer Science*; Communications of the ACM, Volume 31 Number 7, 912-917; July 1988
- [5] ACM / IEEE-CS Joint Curriculum Task Force; *Computing Curricula 1991*; ACM Press; 1991
- [6] Alvarez Rubio, Juan; *Computación en Licenciaturas en Ingenierías*; I Encuentro Interamericano de Educación en Ingeniería; Puebla, México; Agosto 1990

- [7] Alvarez Rubio, Juan; *Un Curso de Introducción a las Sub-Areas de la Disciplina de Computación*; I Congreso Iberoamericano de Educación Superior en Computación; Santiago, Chile; Octubre 1991
- [8] Alvarez Rubio, Juan; *A First Computing Course Based on Curricula 1991*; Proceedings 23rd SIGCSE Technical Symposium; Kansas City, USA; March 1992
- [9] Alvarez Rubio, Juan; *Cursos de la Disciplina de Computación en Postgrados de Informática Educativa*; Congreso Iberoamericano de Informática Educativa; Santo Domingo, República Dominicana; Junio 1992
- [10] Denning, P.J., Comer, D.E.; Gries, D.; Mulder, M.C.; Tucker, A.; Turner, A.J.; Young, P.R.; *Report of the ACM Task Force on the Core of Computer Science*; ACM Press; August 1988
- [11] Gries, David; *Teaching Calculation and Discrimination: A More Effective Curriculum*; Communications of the ACM, Volume 34 Number 3, 44-55; March 1991
- [12] Henderson, Peter B.; *Discrete Mathematics as a Precursor to Programming*; Proceedings 21st SIGCSE Technical Symposium; Feb 1990