“Introduction to Programming Using Ada” de Dennis Volper y Martin D. Katz, Prentice-Hall, Inc., 1990, 653 pp.

SpAda, Número 19, 1993.

En el número correspondiente al pasado mes de Noviembre de *Communications of the ACM* (Vol. 35, No. 11) aparece una sección especial dedicada al lenguaje Ada que contiene, entre otros, un artículo de Mike Feldman titulado “Ada Experience in the Undergraduate Curriculum”. Imprescindible para quienes ejerzan la docencia de la programación: para los detractores de Ada como *primer lenguaje* porque, cuando menos, se sentirán contentos al observar en cuántas universidades de todo el mundo están equivocados (“¡y, para más risa, argumentan su error, ja, ja!”); para sus partidarios, porque a todo el mundo le gusta que le den la razón, especialmente si la tiene (caso contrario —si no—, uno se puede plantear si no será eso de que “a los tontos se les debe dar siempre la razón” —lo que es difícil, y puede ser terrible, claro, es averiguar de qué caso se trata: si sí porque sí —tienes la razón—, o si sí porque sí —eres tonto y, ¡cuidado!, no te hagan enfadar—). Allí, el autor recomienda seis textos para un primer curso de programación a través de Ada: cuatro que ya hemos comentado en *Adateca* (recordemos las obras de F. Culwin, del propio M.B. Feldman y E.B. Koffman, de W.J. Savitch y C.G. Petersen y de J. Skansholm), otro de reciente aparición, “Ada: A First Course in Programming and Software Engineering” de D.C.C. Bover *et al.*, que no hemos tenido todavía la, siempre emocionante, oportunidad de estudiar, y, el sexto (¡nada de chistes, eh!), la obra de Volper y Katz que hoy traemos aquí.

Primer tema: *Building Procedures and Using Packages*: ¡oh sorpresa! Esa capacidad de sorprendernos es la característica primordial de este libro, y esto, no tanto por lo que se cuenta, ni por cómo se cuenta, sino por el orden en que se cuenta. Insistiendo fundamentalmente en tres ideas, la modularidad, la estructuración jerárquica y el tipado fuerte, los conceptos aparecen distribuidos en forma de *espiral* (recuérdese que esta denominación aparecía originalmente en la obra de J. Skansholm): partiendo de la idea básica de *herramientas* (procedimientos), suministradas en *kits* (paquetes), y presentando desde el principio programas “listos para compilar”, muestra incrementalmente nuevos elementos para la programación, a la vez que abunda en los aparecidos previamente. Así por ejemplo, la construcción de procedimientos que aparece por primera vez en el primer capítulo, se describe progresivamente hasta el capítulo seis. Recordemos que el texto de Feldman y Kauffman sigue una estrategia similar, aunque, como diferencias fundamentales, 1) hace más énfasis en conceptos pertenecientes a la Ingeniería de Software (y, en especial, al *Programming in the Large*); 2) las “herramientas pedagógicas” que allí se emplean son originales y variadas (aquí no se pasa de una mera descripción de los conceptos, seguida de ejercicios prácticos, que son, siguiendo el planteamiento global del libro, sucesivas modificaciones y mejoras de ejemplos anteriores); 3) presenta un *método* definido para el desarrollo de software. Esta última circunstancia, el que no se defina claramente una *metodología* para, a partir de la descripción de un problema, diseñar, construir y validar un “programa” que lo resuelva, es quizás la carencia principal de la obra de Volper y Katz, aunque, eso sí, es el único de los textos citados que es exhaustivo en su descripción de Ada: ¡hasta las claúsulas de representación! ¡Y nada menos que tres temas sobre programación concurrente!

Si me dan a elegir, *casi* me quedo contigo.