“Ada: An Introduction to the Art and Science of Programming” de Walter J. Savitch y Charles G. Petersen, The Benjamin/Cummings Publishing Company, Inc., 1992, 764 pp.

SpAda, Número 18, 1992.

Nos encontramos con un nuevo texto para la enseñanza de la programación a través de Ada. Quizá sea oportuno recordar ahora los otros libros con esta misma finalidad, aunque con planteamientos muy diferentes, que ya han aparecido comentados en Adateca:

- *Ada from the Beginning* de J. Skansholm
- *Ada: Problem Solving and Program Design* de M.B. Feldman y E.B. Koffman
- *ADA: A Developmental Approach* de F. Culwin

Walter J. Savitch, profesor de la Universidad de California en San Diego, es autor de tres libros de texto, entre los que destaca “Pascal: An Introduction to the Art and Science of Programming”. Charles G. Petersen enseña en la Iowa State University y destaca su labor como autor de varias propuestas de *curriculum* para los estudios universitarios de *Computer Science*.

“Ada: An Introduction to the Art and Science of Programming” está concebido para servir como libro de texto en cursos de introducción a la programación. Está estructurado de manera que permita diferentes lecturas: apropiado tanto para un curso cuatrimestral como para un curso de año completo. Para ello, se suministran varios “itinerarios” posibles a través de los temas, al tiempo que, en cada uno de ellos, aparecen apartados señaladas como de carácter opcional. Sorprende tal vez alguna de estas posibles lecturas: por ejemplo, aquélla que permite estructurar un curso completo sin tratar un tema tan significativo como son los *arrays*.

Sobre el qué se cuenta, podemos considerar este texto como un libro clásico. Frente al planteamiento revolucionario que aparece en el texto de Feldman y Koffman, aquí nos encontramos con una obra con la tradicional estructura ascendente (*bottom-up*) en la que, a partir de ejemplos muy sencillos, se van describiendo, recurrentemente, conceptos cada vez más avanzados. Al margen de estas consideraciones “metodológicas” hay algunas decisiones en cuanto al orden en que aparecen las materias que son cuando menos debatibles. Así, y al igual que ocurría en el libro de F. Culwin, no es hasta la página 315 cuando se describen los tipos discretos, en especial los enumerados, habiendo sido descritas ya sentencias como *for* ó *case*, o la declaración de procedimientos con parámetros. Esto obliga a los autores a una reconsideración de esas cuestiones, teniendo que realizar una nueva definición de las mismas.

El temario comprende desde un tema de introducción a los computadores hasta temas avanzados como paquetes genéricos y recursión. No aparecen, al igual que ocurre en los otros textos mencionados, temas como la concurrencia y los mecanismos para la interacción con el hardware. Merece ser destacado el tema dedicado a la resolución de problemas empleando recursión, así como los dos temas sobre ficheros, bastante más exhaustivos de lo que nos solemos encontrar. También señalar la elegancia del tratamiento de las estructuras dinámicas, donde se insiste en el empleo de la recursión para procesar estructuras de datos cuya naturaleza es de por sí recursiva.

Desde una perspectiva pedagógica se puede decir que es un libro “guiado por ejemplos”. Aparecen abundantes “casos de estudio”, que sirven tanto para abundar en cuestiones recién explicadas, como para introducir nuevas materias. Se suministran tres tipos de ejercicios: *Self-Test Exercises* —preguntas para resolver “a lápiz”—, *Interactive Exercises* —pequeños programas para realizar delante del terminal—, y *Programming Exercises* —programas completos que requieren la aplicación de algún método de diseño—. Resaltar que tanto los “casos de estudio” como los ejercicios propuestos constituyen quizá lo más interesante del libro.

A título anecdótico, cada capítulo está encabezado con una breve nota biográfica sobre algún informático ilustre: absolutamente injusta la semblanza que, ya casi al final de libro, como si ése fuera su puesto, se hace de ese ser genial e irrepetible que es Edsger W. Dijkstra: una genuflexión de desagravio, ¡ar!