“ADA: A Developmental Approach” de Fintan Culwin, Prentice Hall, 1992, 648 pp.

SpAda, Número 17, 1992.

Esta obra es una nueva aportación a la literatura de “Ada como primer lenguaje de programación”. El autor, *senior lecturer* en el South Bank Polytechnic de Londres, presenta su libro como un texto pensado para un curso de introducción a la programación, que aporta los conocimientos necesarios para una posterior formación en Ingeniería de Software. La obra pretende ser un compendio de tres materias que no aparecen habitualmente en un mismo tratado: la descripción de un lenguaje de programación (*language reference manual*), el método de diseño de programas y la metodología para el desarrollo y verificación de software. Este carácter compilatorio del texto (“tres en uno”) obliga a que, necesariamente, contempladas por separado, no aparezcan estas materias con la misma profundidad que en obras dedicadas en exclusiva a cada una de ellas. Analizaremos el libro desde esa triple perspectiva.

Aunque sea sólo un juego de palabras (o de siglas), la aparición de “ADA” (así, todo con mayúsculas) en el título de libro parece anunciar su dedicación prioritaria a la descripción de este lenguaje: un libro para aprender Ada. Bajo este punto de vista, la obra presenta graves defectos. Y no se trata de que no sea exhaustiva en su relación de los elementos del lenguaje, circunstancia más que justificada en un texto que no pretende ser en exclusiva un “manual de Ada”, sino de que lo que se describe se hace en un orden harto caprichoso (y, consecuentemente, poco pedagógico) y, en ocasiones, no siempre (quede claro), con falta de rigor. Como tampoco pretendemos ser exhaustivos, destaquemos sólo, por su gravedad, la siguiente circunstancia: en un texto con unas quinientas cuarenta páginas “útiles”, aparece por primera vez el concepto de *tipo* en la página 185, donde se describen los tipos númericos, mientras que los tipos enumerados no se encuentran hasta la página 236. Si tenemos en cuenta que a esas alturas ya se han descrito sentencias como *for* o *case*, en las que son fundamentales estos tipos enumerados, o que ha aparecido previamente cómo construir procedimientos con parámetros, y si recordamos, además, esa máxima de que, para horror de los “Hombres C”, “Ada es un lenguaje fuertemente tipado”, habrá común acuerdo en señalar lo poco afortunado de este hecho.

La metodología de diseño de programas empleada a lo largo de toda la obra es la que se conoce habitualmente como de refinamiento sucesivo (*top-down stepwise refinement*). El proceso global de desarrollo de programas propuesto por el autor comprende las siguientes fases: *user interface design, the program design, the data design design, the program listing.* Este manera de proceder parece adecuada para la elaboración de programas de “andar por casa”, pero, evidentemente, es insuficiente para la construcción de sistemas de grandes dimensiones (*programming in the large*). Aunque el autor prometa tratar técnicas más avanzadas como el Diseño Orientado a Objetos en los últimos capítulos de la obra, allí no pasa de describir cómo implementar un *abstract data type.* Por último, el autor dedica sendos capítulos a la evaluación de algoritmos (criterios subjetivos —eficiencia, adecuación, elegancia, etc.—) y a la complejidad algorítmica (criterio objetivo).

Como tratado para el desarrollo de software, en el libro aparecen cuestiones como *Testing, Tracing and Debugging*, y *Production and Documentation.* Basándose en ejemplos sencillos (excesivamente, quizás), se presentan ciertos “consejos” (no me atrevería a llamarlo “técnicas”) sobre cómo validar y depurar un programa. El hecho de que se utilicen criterios demasiado “ad hoc” para los ejemplos propuestos, hace que aquéllos sean difícilmente trasladables a casos diferentes. Por otro lado, el capítulo de Producción y Documentación es un pequeño manual de cómo distribuir el tiempo de desarrollo de un programa y de cómo documentar el producto final.

Observada desde un punto de vista didáctico, la obra, por culpa de una verbosidad sin dudas excesiva, resulta un tanto difícil de digerir. Los programillas propuestos como ejercicios no son tampoco lo suficientemente atractivos como para despertar en el alumno su devoción por Ada. Y de eso se trata.