



aprenderaprogramar.com

Modificación directa del flujo de programas: algoritmia. Estructuras secuenciales, de decisión y de repetición. (CU00177A)

Sección: Cursos

Categoría: Curso Bases de la programación Nivel I

Fecha revisión: 2024

Autor: Mario R. Rancel

Resumen: Entrega nº 76 del Curso Bases de la programación Nivel I

24

MODIFICACIÓN DIRECTA DEL FLUJO DE PROGRAMAS. INTRODUCCIÓN

Hasta ahora hemos abordado la construcción de algoritmos a partir de tres tipos de estructura:

- a) Estructura secuencial.
- b) Estructura de decisión.
- c) Estructura de repetición.

Existe un amplio consenso, y la experiencia de cualquier programador lo confirma, en relación a que éstas son las tres estructuras absolutamente necesarias para programar. Los programas, por muy largos y complejos que sean, se pueden construir combinando adecuadamente estas tres estructuras básicas. Si esto es así, podemos afirmar que con los conocimientos que tenemos estamos preparados para:

- Diseñar algoritmos eficientes.
- Desarrollar algoritmos largos.
- Desarrollar algoritmos complejos.

Y en buena medida esto es posible porque desarrollamos algoritmos con flujos muy ordenados, donde una instrucción se ejecuta a continuación de la anterior, donde los caminos dispersos siempre van confluyendo, donde las entradas, salidas y número de iteraciones en los bucles están muy bien controladas y donde existe un inicio, desarrollo y fin del programa. Todo ello nos lleva a:

- Programas claros para leerlos.
- Programas fáciles de entender.
- Programas fáciles de corregir.
- Programas eficientes en su funcionamiento.

Ahora nos preguntamos: ¿Existen otras posibilidades a la hora de desarrollar programas? (Posibilidades como que una instrucción no se ejecute a continuación de la otra, como que los caminos dispersos no confluyan, como poder salir de un bucle sin hacerlo por la vía normal, como que no exista un único fin sino varios, etc.)

La respuesta es que sí, esas posibilidades existen. Para bien o para mal, los ordenadores nos ofrecen mucha potencia y muchas posibilidades. Somos casi dioses. Cuando se utilizan estas herramientas que no obedecen a ninguna de las tres estructuras básicas, se habla de programación no estructurada, por contraposición a la programación estructurada¹ basada en la secuencialidad, decisión y repetición. El propio término “no estructurado” nos avisa de algo. Posiblemente nos recuerda a “desorden” o “falta de organización”. La aparición de “desorden” normalmente está vinculada a instrucciones que dan lugar a una modificación directa del flujo del programa (“saltar” de un punto a otro).

Hace ya varias décadas surgió un movimiento abanderado por estudiosos de la programación que pretendía establecer unas técnicas y disciplina que permitiera mejorar la calidad de los programas. Fue

¹ No vamos a entrar en el desarrollo o estudio de estos conceptos mas que en ideas básicas.

Seguir considerándolas como herramientas igual de válidas para construir programas apoyándose en ellas hubiera significado aceptar que eran igual de buenas que otras técnicas de programación, y esto no es así, como ha demostrado la programación estructurada. A modo de conclusión, propondremos:

1. Crear programas basados en estructuras secuenciales, de decisión y de repetición, con flujos naturales.
2. Evitar el uso de control directo de flujos ("saltos").
3. En circunstancias concretas y no de forma habitual, apoyarnos en el control directo del flujo del programa.

¿A qué llamamos circunstancias concretas? Pues aquellas situaciones en las que el uso de instrucciones de control directo del flujo:

- Permitan salida a una situación difícilmente programable de otra manera.
- Introduzcan una gran claridad y facilidad de comprensión frente a otra opción.
- Permitan condensar una gran cantidad de código teniendo fácil lectura e interpretación.

Hemos hecho una pequeña introducción para las instrucciones que vamos a estudiar a continuación. Los motivos para ello han sido dos: por un lado, continuar con la línea que venimos manteniendo de reflexionar en torno a estrategias de aproximación a los problemas, aspecto más "filosófico" que técnico. Y por otro, evitar presentar estas instrucciones como si de unas más se trataran. Con todo lo expuesto, creemos que se parte de unas pautas útiles para hacer un buen uso de ellas.

La cantidad de instrucciones disponibles para el control directo de flujos es variable en función del lenguaje que se utilice. Vamos a estudiar, para la creación de pseudocódigo, las siguientes:

- Finalizar.
- SalirDesde.
- SalirMientras.
- SalirHacer.
- IrA.

Próxima entrega: CU00178A

Acceso al curso completo en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) --> Cursos, o en la dirección siguiente:

http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=28&Itemid=59