aprenderaprogramar.com

Economía, eficiencia y lenguaje del algoritmo ejemplo (CU00124A)

Sección: Cursos

Categoría: Curso Bases de la programación Nivel I

Fecha revisión: 2024

Autor: Mario R. Rancel

Resumen: Entrega nº 23 del Curso Bases de la programación Nivel I

24

ECONOMÍA, EFICIENCIA Y LENGUAJE EMPLEADO

En la entrega anterior del curso, vimos distintas versiones de un algoritmo. Vamos a utilizar el ejemplo visto para analizar cuestiones como economía del algoritmo, eficiencia del algoritmo y el lenguaje utilizado.

En cuanto a economía del algoritmo, es la *versión 1* la que tiene mayor número de instrucciones, apreciándose que existe cierto grado de repetición entre ellas. Si el resultado es el mismo, buscaremos que un algoritmo sea lo más corto y menos repetitivo posible. Por ejemplo, *la versión 2*, hace prácticamente lo mismo eliminando repeticiones.

En cuanto a eficiencia del algoritmo, vamos a suponer que se trata de una mesa grande en la que se emplean diez segundos en desplazarse desde el puesto de un comensal al de otro. Nos enfrentamos a la "concepción" del algoritmo. En las *versiones 1 y 2* se coloca elemento a elemento, sin tratar de agrupar todos los elementos colocables tras un desplazamiento. En las *versiones 3, 4 y 5* se agrupan los elementos a colocar por silla. En la *versión 6* no queda suficientemente definido cómo se colocará la mesa, con lo cual no sabemos el tiempo que llevará.

Los tiempos de desplazamiento serían los siguientes:

Versión 1 y 2: 4 vueltas a la mesa x 30 seg/vuelta = 120 seg.

Versión 3, 4 y 5: 1 vuelta a la mesa x 30 seg/vuelta = 30 seg.

Versión 6: no definido.

De momento, pues a continuación matizaremos, sería el algoritmo versión 5 el que resultaría óptimo pues aúna economía y eficiencia.

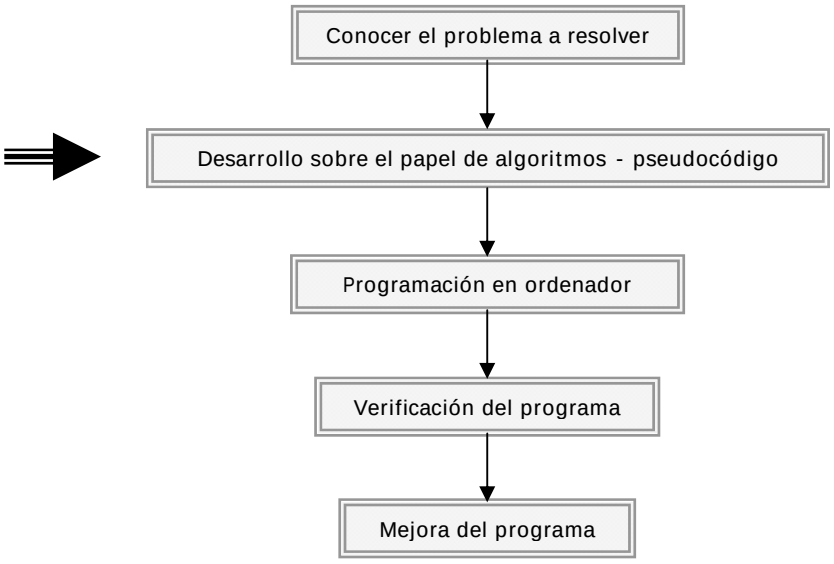
Analizaremos ahora el lenguaje empleado. Éste evoluciona con las versiones del algoritmo requiriendo cada vez mayor capacidad de comprensión. Supongamos que el algoritmo lo ejecuta un robot con capacidad para entender órdenes básicas. La capacidad de comprensión necesaria y la probabilidad de error variaría según esta tabla:

Versión	Capacidad de comprensión necesaria	Probabilidad de error
1	Mínima Necesita interpretar “1º, 2º, 3º”	Mínima
2	Mínima Necesita interpretar “cada”	Mínima
3	Media Necesita interpretar “cada”, “siguiente”	Baja
4	Media Necesita interpretar “cada”, “siguiente”, frase más larga	Baja
5	Alta Necesita interpretar “cada”, “siguiente” y “colocar cubiertos”	Alta P. ej. 3 cubiertos al mismo lado
6	Muy Alta Necesita interpretar “colocar platos y cubiertos”	Muy Alta P. ej. todos los platos y cubiertos en una pila

Se trata pues, de distintos niveles de lenguaje, algo de lo que ya hemos hablado. Podemos desarrollar algoritmos con un lenguaje “libre”, incluso usando elementos gráficos, esquemas, dibujos, etc., sin que su utilidad se vea menoscabada. Sin embargo, dado que nuestro interés se centra en programar ordenadores, dejaremos de lado los estilos “libres” y empezaremos a trabajar enfocados única y exclusivamente hacia el lenguaje de los ordenadores.

No usaremos directamente un lenguaje, puesto que éstos cambian y a nosotros mismos nos puede interesar usar uno u otro, sino un lenguaje algorítmico aplicable a la mayor parte de los lenguajes de programación.

Sobre el esquema ya planteado recordemos dónde estamos.



Un programa o serie de instrucciones en un lenguaje informático se llama *código*. El lenguaje que utilizaremos para crear algoritmos tiene la misma concepción de fondo, las mismas herramientas y similar sintaxis. De ahí que lo llamemos pseudocódigo. Las normas que indicaremos para escribir pseudocódigo no son de obligado cumplimiento: cada cual puede utilizar sus criterios, si bien es verdad que la mayoría de los programadores tienen formas similares de escribir pseudocódigo.

Los procesos o sucesión de instrucciones también se pueden representar gráficamente usando esquemas o dibujos. Tampoco es obligado ajustarse a unas normas para crear estos esquemas. Sin embargo, gozan de cierta estandarización entre los programadores, incluso a través de normas internacionales como la *ISO*. Dado los objetivos que perseguimos, usaremos únicamente un modelo simplificado de los denominados *diagramas de flujo.* Este tipo de esquemas mezcla texto con símbolos y flechas para describir los procesos. Tienen la ventaja de ser didácticos e intuitivos, por los que los consideramos de gran interés para comenzar a programar.

Próxima entrega: CU00125A

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:
http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=28&Itemid=59