



aprenderaprogramar.com

# Verificación de algoritmos o programas. Enfoque al resultado, al proceso o zonal. (CU00235A)

Sección: Cursos

Categoría: Curso Bases de la programación Nivel II

Fecha revisión: 2024

Autor: Mario R. Rancel

Resumen: Entrega nº 34 del Curso Bases de la programación Nivel II

24

## VERIFICACIÓN DE ALGORITMOS CON ENFOQUE AL RESULTADO, AL PROCESO O ZONAL.

Otra de las ventajas de los ordenadores o dispositivos programables de cara a la verificación de algoritmos es la posibilidad de ignorar algunas partes del mismo que no nos resultan de interés para centrarnos en una o varias partes que son conflictivas o delicadas. Es lo que llamamos “enfoque de la verificación del algoritmo”.

Si nos centramos en los resultados sin prestar atención a los procesos y variables intermedias hablamos de *enfoque al resultado*. Es lo que hacemos cuando confiamos en que el funcionamiento del algoritmo va a ser correcto. Comprobamos los resultados y si son correctos, entendemos que los procesos intermedios también han de serlo.

Si tenemos sospechas de que un proceso intermedio está fallando de alguna manera, damos instrucciones al ordenador para que nos informe de todos los cambios de valores en variables que puedan estar existiendo en ese proceso, sin prestar atención a los resultados. Se trataría de un enfoque al proceso.

Por último, cuando un algoritmo es relativamente largo, damos instrucciones al ordenador para que nos muestre todo lo que ocurre en torno a una zona donde sabemos o sospechamos que puede haber problemas. Se trataría de un enfoque zonal.

La clasificación de enfoques es muy teórica, ya que en la práctica existen todas las posibilidades (p. ej. enfoque al resultado + enfoque al proceso ó enfoque a la zona A + enfoque a resultados, etc.). Lo importante es tener la idea de, ante el fallo de un algoritmo cuando lo probemos en ordenador, recurrir a enfocar aquellas partes o aspectos que nos suponen problemas.

Vamos a verlo con un ejemplo. Partimos del anidamiento ya conocido y suponemos que al ejecutarlo en ordenador obtenemos unos resultados erróneos y queremos determinar dónde se encuentra el problema y corregirlo. Vamos a ver distintos enfoques; en este caso los escribiremos con pseudocódigo por no utilizar un lenguaje concreto, pero obviamente esto no es necesario. Partiremos del pseudocódigo para desarrollar el código, pero la realización de los distintos enfoques la haríamos ya directamente en el lenguaje correspondiente.

- **Enfoque al resultado:**

```
Desde i = a hasta b Hacer
    Desde j = m hasta n Hacer
        A = i * j
        Mostrar A
    Siguiendo j
Siguiendo i
```

El ordenador nos muestra el valor de A y confiamos que todo está correcto si A es correcto. Es habitual, sobre todo cuando las cosas van “sobre ruedas”.

- **Enfoque al proceso:**

```

Mostrar a : Mostrar b
Desde i = a hasta b Hacer
    Mostrar m : Mostrar n : Mostrar i
    Desde j = m hasta n Hacer
        Mostrar j
    Siguiente j
Siguiente i
    
```

Analizamos cambios en variables que dan lugar al resultado, incluso “olvidándonos” del resultado (en este caso hemos eliminado el cálculo y la muestra del resultado). Introduciendo unos datos de partida obtendremos un rastreo de qué ha pasado. Supongamos  $a = 1$   $b = 3$   $m = 0$   $n = 2$ . El ordenador mostraría algo así:

```

a = 1 b = 3
m = 0 n = 2 i = 1
j = 0
j = 1
j = 2
m = 0 n = 2 i = 2
j = 0
j = 1
j = 2
m = 0 n = 2 i = 3
j = 0
j = 1
j = 2
    
```

El rastreo parte de bases razonadas pero también de hipótesis más o menos aleatorias. Por ejemplo, *Mostrar a* y *Mostrar b* nos puede informar de un problema con los valores de gestión del bucle exterior. No estamos seguros de que  $a$  o  $b$  puedan estar dando lugar a un fallo, pero los chequeamos por si acaso.

*Mostrar m* y *Mostrar n* en el interior del bucle en teoría no es necesario porque  $m$  y  $n$  no varían en el algoritmo: en teoría. Pero quizás al observar resultados encontremos que  $m$  y  $n$  se trastocan a valores extraños por procesos que no habíamos tenido en cuenta... y obtengamos una pista. Somos “detectives” en busca de un error y “cualquier pista es buena”. Esta es la razón que justifica el que, en ocasiones, procedamos a estudiar cada una de las variables, procesos o pasos de un programa, incluyendo aquellos que pensamos no influyen en el problema detectado. Comprobaremos que en determinadas ocasiones los errores se localizan en sitios totalmente inesperados.

- **Enfoque zonal a la entrada del bucle:**

```

Mostrar a
Mostrar b
Mostrar m
Mostrar n
Mostrar i
Mostrar j
Mostrar A
Desde i = a hasta b Hacer
    Desde j = m hasta n Hacer
        A = i * j
        Mostrar A
    Siguiente j
Siguiente i
    
```

En este caso sospechamos que hay un problema en una zona que identificamos como “entrada del bucle” y chequeamos lo que ocurre en ese entorno. En principio es lógico verificar  $a$ ,  $b$ ,  $m$  y  $n$  pero no  $i$ ,  $j$  y  $A$ . Qué se verifica lo decide el programador. Cuando no se encuentra dónde está el posible fallo, se chequea todo con el fin de “buscar pistas”, que pueden aparecer en sitios “poco lógicos”.

- **Enfoque zonal a la salida del bucle:**

Consistirá en trasladar las 7 instrucciones *Mostrar* a la salida del bucle.

Hemos visto diferentes posibilidades de rastreo existiendo un número de combinaciones posibles elevadísimo. El programador decide cómo proceder atendiendo a su experiencia y a una estrategia razonada para ir descartando posibles puntos de fallo hasta dar con el problema.

**Próxima entrega: CU00236A**

**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=36&Itemid=60](http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=36&Itemid=60)