



aprenderaprogramar.com

# Problemas en el uso de bucles Desde-Siguiente (for-next) y cómo solucionarlos. (CU00153A)

Sección: Cursos

Categoría: Curso Bases de la programación Nivel I

Fecha revisión: 2024

Autor: Mario R. Rancel

Resumen: Entrega nº 52 del Curso Bases de la programación Nivel I

24

## USO Y MAL USO DE LA INSTRUCCIÓN DESDE ... SIGUIENTE

La instrucción *Desde ... Siguiendo* tiene una sintaxis y funcionamiento que aconsejan estar muy despiertos a la hora de su uso. El problema puede venir por varios motivos: valores inicial y final poco coherentes, uso de un paso inapropiado, manipulaciones del contador indebidas, etc. El resultado puede ser un error, funcionamiento incorrecto o inesperado, etc.



La solución a estos problemas pasa por:

- Hacer un buen uso de la instrucción *Desde*, evitando las prácticas inadecuadas que describiremos a continuación.
- Elegir otro tipo de estructura de repetición cuando resulte ventajoso.
- Programas limpios, ordenados, etc. desde el principio contribuyen a instrucciones limpias y ordenadas hasta el final.

En cuanto a lo que describiremos como “malos usos de la instrucción *Desde*” hay que tener en cuenta que algunas de las situaciones planteadas son aceptables cuando son situaciones controladas. Vendría a ser equivalente a circular con un coche a 200 Km/h. Si la situación es controlada (circuito cerrado, piloto profesional, etc.) es asumible. En cambio, en una situación no controlada o imprevista (carretera convencional, piloto no profesional) la situación es indeseable.

- Caso valor final menor que valor inicial y variantes

<p>[Ejemplo aprenderaprogramar.com] Desde VAR = Vi hasta Vf Hacer Mostrar “Mensaje” Siguiendo</p>
---------------------------------------------------------------------------------------------------------------

Si resulta que  $V_i$  es mayor que  $V_f$  el bucle se ejecutará cero veces. Si en toda circunstancia se produce  $V_i > V_f$  resultaría que nos encontramos con un fragmento de código muerto, y que resulta totalmente indeseable. Un programa debe contener cero código muerto. En cambio, si  $V_i > V_f$  sólo circunstancialmente y de forma controlada, es asumible.

Con el uso de pasos negativos, estaríamos en situación equivalente cuando  $V_i$  es menor que  $V_f$ .

- **Caso valor inicial igual al valor final**

**Desde VAR = Vi hasta Vf Hacer**  
 Mostrar "Mensaje"  
**Siguiente**

Si  $V_i$  es igual a  $V_f$  el bucle se ejecutará una única vez. Si en toda circunstancia se produce  $V_i = V_f$  resultaría que nos encontramos con una estructura repetitiva funcionando a modo de estructura secuencial. Algo indeseable por cuanto pueden aparecer errores, mal funcionamiento, etc. y, aún en caso de que no hubiera ningún problema indeseable, por ir en contra de la claridad, economía y eficiencia del programa. Las instrucciones deben ser las justas y necesarias y lo más claras posibles.

- **Caso de paso más grande que el intervalo**

**Desde VAR = Vi hasta Vf Paso = Inc Hacer**  
 Mostrar "Mensaje"  
**Siguiente**

Llamamos intervalo al valor del paso que permite al menos una repetición del bucle. Por ejemplo: Si  $V_i = 1$  y  $V_f = 10$ , el valor del paso para que se repita dos veces el proceso es  $10 - 1 = 9$ . En este caso:

1ª Iteración: VAR = 1

2ª Iteración: VAR = 10

3ª Iteración: al ser VAR = 19 se sale del bucle.

Si se diera  $V_i = 1$ ,  $V_f = 10$ ,  $paso = 10$  tendríamos:

1ª Iteración: VAR = 1

2ª Iteración: VAR = 11 y se sale del bucle sin que se haya dado siquiera una repetición del mismo.

Esta situación de "ejecución una única vez" es análoga a la comentada en el caso anterior. Por lo tanto, sólo es aceptable cuando se produce de forma circunstancial y controlada.

- **Caso manipulación del contador externa al Desde**

Cuando el contador de una instrucción *Desde* va tomando sucesivos valores decimos que sufre variaciones internas. Si durante la ejecución de un *Desde* otras instrucciones afectan al valor del contador decimos que se ha producido una manipulación externa. Esta actuación supone que la instrucción deja de ejecutarse como sería previsible y precisamente los resultados pasan a ser eso: imprevisibles. Por ello pondremos un “prohibido” a esta práctica, y a partir de aquí cada cual que obre en consecuencia.

Como sabemos que hay mucho amante de lo prohibido supongamos un:

<p><b>Desde <math>i = 1</math> hasta 20 Hacer</b>          Mostrar <math>i</math>  <math>i = i + 2</math>  <b>Siguiente</b></p>
---------------------------------------------------------------------------------------------------------------------------------------------

La instrucción  $i = i + 2$  supone una manipulación del contador. En teoría entramos con  $i = 1$ , muestra  $i$ ,  $i$  pasa a valer 3, itera y suma uno pasando  $i$  a valer 4, muestra  $i = 4$ ,  $i$  pasa a valer 6 y suma uno pasando  $i$  a valer 7... y así sucesivamente. Nos mostraría 1, 4, 7, 10, 13, 16, 19... en teoría. En la práctica, veremos que hay formas más claras y elegantes de hacer esto así que... no lo hagas por favor.

- **Caso Paso con valor impropio**

<p><b>Desde VAR = Vi hasta Vf Paso = Inc Hacer</b>          Mostrar “Mensaje”  <b>Siguiente</b></p>
-------------------------------------------------------------------------------------------------------------

Si  $Vi \leq Vf$  y  $Inc = 0$  nos encontramos con que cada repetición del bucle  $Vi = Vi + 0$  con lo cual nunca se llegará a producir la condición para salida del bucle. Esta situación se llama de “bucle infinito” y resulta altamente indeseable. Por supuesto, que se produzca una situación como la descrita es un fallo del programador por un mal diseño, una mala previsión de procesos internos o entradas inesperadas del usuario. Cuando se produce un bucle infinito es habitual que el ordenador se quede “colgado”, pues lo que está haciendo es repetir el mismo proceso una y otra vez. Para evitarlo será el diseño del algoritmo y el seguir las pautas de buena programación lo más efectivo. También podremos disponer, como veremos más adelante, mecanismos de seguridad.

Una variante de este caso es tener un valor inicial inferior al final y un paso negativo. Si esto ocurre se evalúa si el valor inicial es menor que el final, y al ser así, el bucle se ejecuta cero veces. Los comentarios que venimos haciendo son válidos al respecto.

Los valores más habituales del paso serán números enteros. Usar números fraccionarios será, en general, menos deseable. De cualquier forma, en caso de usar fracciones se buscará que el número sea decimal exacto. Por ejemplo:  $3 / 5 = 0,6$ . El uso de números decimales periódicos se considerará

indeseable (p. ej.  $2 / 3 = 0,666666\dots$ ) pues supone una indefinición de fondo. Remitimos a lo expuesto en el apartado de “Herramientas lógicas y matemáticas” (paradoja de los decimales finitos).

- **Caso de usos impropios de variable contador o variable de paso**

Siempre que sea posible, las variables que se usen para definir el contador y el paso no deben usarse para otras funciones ni dentro de la instrucción *Desde* ni en el resto del programa.

**Ejemplo:**

a)

```
Desde i = 1 hasta 5 Hacer
    Mostrar "Por favor introduzca su edad"
    Pedir i
    Mostrar "Usted nació en" 2005 - i
Siguiente
```



b)

```
Desde i = 1 hasta 5 Hacer
    Mostrar "Contando"
Siguiente
    Mostrar "Por favor introduzca su edad"
    Pedir i
```

Ambas situaciones son indeseables. Las variables deben estar ordenadas, y no usarse “al boleo”. El caso a) es más grave si cabe, pues está modificando el contador de una forma que produce escalofríos. Hay formas mucho más razonables de pedir 5 veces una edad.

Con el paso puede suceder otro tanto:

a)

```
Desde i = 1 hasta 10 Paso = P Hacer
    Mostrar "Introduzca el número de pisos del edificio"
    Pedir P
    Si P > 10 Entonces
        Mostrar "El edificio es alto"
    SiNo
        Mostrar "El edificio es bajo"
    FinSi
Siguiente
```



b)

```
[Ejemplo aprenderaprogramar.com]
Desde i = 1 hasta 10 Paso = P Hacer
    Mostrar "Contando"
Siguiente
Mostrar "Introduzca el número de pisos del edificio"
Pedir P
Si P > 10 Entonces
    Mostrar "El edificio es alto"
SiNo
    Mostrar "El edificio es bajo"
FinSi
```

La situación a) es relativamente distinta cuando se afecta al paso respecto a cuando se afecta al contador. La variación radica en que cuando se entra en el bucle, el valor de paso queda fijado y es inamovible, de modo que aunque la variable sufra modificaciones el bucle en teoría sigue su funcionamiento previsto. Sin embargo, esto no modifica para nada nuestra indicación de que este tipo de uso de variables debe ser evitado a toda costa.

**Próxima entrega: CU00154A**

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