



APRENDERAPROGRAMAR.COM

PROGRAMACIÓN PARA
NIÑOS Y ADOLESCENTES.
CÓMO ESCRIBIR CÓDIGO
EFICIENTE CON EJEMPLO
"DE MÉXICO A
MONTERREY" (PN00540G)

Sección: Para niños

Categoría: Curso de programación para niños con Didac-Prog Cartesia

Fecha revisión: 2039

Resumen: Entrega n°40 del Curso de programación para niños.

Autor: Mario Rodríguez Rancel

ORGANIZAR EL CÓDIGO DE FORMA EFICIENTE

¡Hola a todos! Avanzamos con nuestros ejercicios resueltos de programación para niños y adolescentes. En una entrega anterior hemos visto un código que dibuja tres circunferencias. Cuando creamos código una cosa interesante es hacerlo de forma eficiente. Eficiencia significa no malgastar. Por ejemplo un coche es más eficiente cuanto menos gasolina gasta.



Esta entrega la catalogamos como de programación avanzada. Por tanto no está recomendada para primaria sino sólo para secundaria.

Esta entrega está dirigida a mayores de 12 años. Si tienes menos edad, sáltala.

EL EJEMPLO DEL VIAJE POR MÉXICO

Supongamos que estamos en México y tenemos que ir desde la Ciudad de México hasta Monterrey. Podemos hacerlo de distintas maneras. Podríamos ir directamente, o podríamos ir pasando por la ciudad de Guadalajara.



Supongamos que tanto desde Guadalajara como desde Ciudad de México tuviéramos que recorrer 200 kilómetros para llegar a Monterrey, y que para ir de Ciudad de México a Guadalajara tuviéramos que recorrer 100 kilómetros. Supongamos que gastamos 1 litro de gasolina por cada kilómetro que recorremos. Si vamos directamente de Ciudad de México a Monterrey habremos gastado 200 litros de combustible, mientras que si vamos por Guadalajara habremos gastado 300 litros de combustible. En este

caso decimos que es más eficiente ir de Ciudad de México a Monterrey directamente que pasando por Guadalajara, porque gastamos menos combustible.

Supongamos que dos coches, un Ford y un Toyota, quieren ir de Ciudad de México a Monterrey, y el Ford va directamente y el Toyota va pasando por Guadalajara.

Ambos coches llegan a Monterrey. ¿Cuál de los dos ha cumplido su objetivo? Ambos coches han cumplido su objetivo, porque ambos han llegado a Monterrey. ¿Cuál de los dos coches ha sido más eficiente? Ha sido más eficiente el Ford porque ha tardado menos tiempo y ha gastado menos combustible.

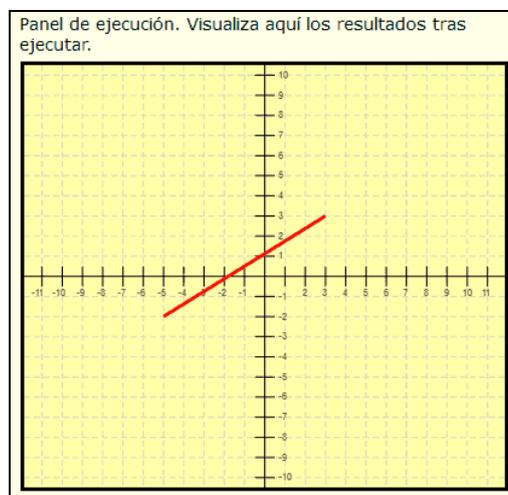
Con la programación de ordenadores ocurre lo mismo: dos programas pueden cumplir el objetivo, pero uno de ellos ser más eficiente porque uno tarda menos tiempo y menos energía del computador que el otro.

Hay varias formas de medir la eficiencia de un programa de ordenador. Una de ellas se basa en contar el número de instrucciones que se ejecutan. El programa que cumpla el objetivo ejecutando un menor número de instrucciones es más eficiente. Otra forma de medir la eficiencia es por el número de líneas de código que requieren los programas. Nosotros vamos a fijarnos en el número de líneas de código que tienen nuestros programas. Vamos a verlo con ejemplos.

DOS CÓDIGOS QUE HACEN LO MISMO

A continuación tenemos dos códigos y ambos hacen lo mismo: dibujar una línea roja entre dos puntos del plano cartesiano.

<p>-- EJEMPLO PN00540G-0</p> <p>Iniciar programa Nuevo color lápiz (1) Nuevo grosor lápiz (2) Dibujar línea desde (-5,-2) hasta (3,3) Finalizar programa</p>	<p>-- EJEMPLO PN00540G-1</p> <p>Iniciar programa Nuevo color lápiz (5) Nuevo color lápiz (1) Nuevo grosor lápiz (2) Dibujar línea desde (-5,-2) hasta (3,3) Finalizar programa</p>
--	---



¿Cuál de los dos programas cumple el objetivo de dibujar la línea? Ambos programas cumplen el objetivo.

¿Cuál de los dos programas es más eficiente? Para responder esta pregunta vamos a contar las líneas que tiene cada programa. Las líneas de comentario no las contaremos porque los comentarios no son instrucciones que se ejecuten, por lo tanto nos da igual si un programa tiene más o menos comentarios, al menos en cuanto a su eficiencia.

El programa PN00540G-0 tiene 5 líneas, mientras que el programa PN00540G-1 tiene 6 líneas. El resultado es el mismo, pero el programa PN00540G-0 es más eficiente "porque gasta 5 unidades de energía del computador" mientras que el programa PN00540G-1 "gasta" 6 unidades. ¿Por qué ocurre esto? Si nos fijamos el programa PN00540G-1 primero establece el color del lápiz de dibujo a azul (color 5) e inmediatamente después a rojo (color 1). Pero aquí poner el color 5 primero no nos ha servido para nada porque no hemos dibujado nada. Por tanto lo que hemos hecho es perder energía en algo que no nos ha sido útil. Ese código que no sirve para nada lo llamamos "código inútil" o "código muerto". Para mejorar el código y hacerlo más eficiente sería mejor borrar todas las líneas que no resulten útiles.

REGLAS PARA CONTAR CUANDO HAY CONDICIONALES Y BUCLES

Las reglas que utilizaremos para contar líneas en Cartesia son:

- Cada instrucción del tipo "Iniciar programa", "Dibujar punto", "Dibujar línea", "Nuevo color lápiz", "Nuevo grosor lápiz", "Asignar valor a variable" o "Finalizar programa" cuenta como 1 línea.
- En un condicional, el "Si verdadero..." y el comienzo de bloque y fin de bloque cuentan como una línea. Además habrá que contar todas las líneas que haya dentro del bloque.
- En una repetición, el "Repetir (...) veces" y el comienzo de bloque y fin de bloque cuentan como una línea. Además habrá que contar el número de líneas que haya dentro del bloque multiplicado por el número de repeticiones que tengan lugar.

A continuación tenemos dos códigos y ambos hacen lo mismo: dibujar diez puntos en unas coordenadas determinadas sobre el plano cartesiano.

<p>-- EJEMPLO PN00540G-2</p> <p>Iniciar programa Nuevo color lápiz (1) Dibujar punto en (-5,-5) Dibujar punto en (-4,-4) Dibujar punto en (-3,-3) Dibujar punto en (-2,-2) Dibujar punto en (-1,-1) Dibujar punto en (0,0) Dibujar punto en (1,1) Dibujar punto en (2,2) Dibujar punto en (3,3) Dibujar punto en (4,4) Finalizar programa</p>	<p>-- EJEMPLO PN00540G-3</p> <p>Iniciar programa x nuevo valor es (-5) Nuevo color lápiz (1) Repetir (10) veces ejecutar bloque Comienzo bloque Dibujar punto en (x,x) x nuevo valor es (x+1) Fin bloque Finalizar programa</p>
---	---

¿Cuál de estos programas cumple el objetivo? Ambos lo cumplen porque ambos dibujan los diez puntos en las coordenadas deseadas.

¿Cuál de los dos programas es más eficiente? Para responder esto contamos el número de líneas de que consta cada programa. Los comentarios no los contamos.

En el programa PN00540G-2 hay 13 líneas, que corresponden a las instrucciones de comienzo y final de programa, a la instrucción de nuevo color del lápiz, y a las diez instrucciones para dibujar los diez puntos.

En el programa PN00540G-3 tenemos las líneas de comienzo y final de programa (2 instrucciones), más la instrucción para asignar valor a x (1 línea, llevamos 3 acumuladas), más la instrucción de nuevo color del lápiz (1 línea, llevamos 4 acumuladas), más la instrucción repetir (1 línea, llevamos 5 acumuladas). Dentro del bloque repetir tenemos las líneas de dibujar punto y de asignar valor a x (2 líneas, llevamos 7 acumuladas).

Por tanto el programa PN00540G-2 tiene 13 líneas y el PN00540G-3 tiene 7 líneas y es el que resulta más eficiente, al menos en cuanto a número de líneas que es en lo que nos estamos fijando nosotros ahora.

¿Qué ocurriría si quisiéramos dibujar 20 puntos de forma similar? El primer programa requeriría 23 líneas, mientras que el segundo seguiría requiriendo 7 líneas.

¿Qué ocurriría si quisiéramos dibujar 200 puntos de forma similar? El primer programa requeriría 203 líneas, mientras que el segundo seguiría requiriendo 7 líneas.

Por tanto en general es mucho mejor escribir programas con bucles y variables que programas donde repetamos muchas líneas parecidas.

EL PROBLEMA DE LAS INSTRUCCIONES QUE SE REPITEN INÚTILMENTE DENTRO DE BUCLES

Antes hemos visto cómo el número de líneas nos puede servir para saber entre dos programas cuál es más eficiente. Pero a veces hay programas con igual número de líneas y uno resulta más eficiente que otro.

A continuación tenemos dos códigos y ambos hacen lo mismo: dibujar diez puntos en unas coordenadas determinadas sobre el plano cartesiano. Ejecuta ambos códigos y compruébalo.

<p>-- EJEMPLO PN00540G-3</p> <p>Iniciar programa x nuevo valor es (-5) Nuevo color lápiz (1) Repetir (10) veces ejecutar bloque Comienzo bloque Dibujar punto en (x,x) x nuevo valor es (x+1) Fin bloque Finalizar programa</p>	<p>-- EJEMPLO PN00540G-4</p> <p>Iniciar programa x nuevo valor es (-5) Repetir (10) veces ejecutar bloque Comienzo bloque Nuevo color lápiz (1) Dibujar punto en (x,x) x nuevo valor es (x+1) Fin bloque Finalizar programa</p>
---	---

Si contamos ambos programas tienen el mismo número de líneas por lo que podríamos pensar que son igual de eficientes. No obstante hay otro criterio que resulta interesante tener en cuenta: el número de instrucciones que ejecuta el computador.

En el programa PN00540G-3 tendríamos las instrucciones "Iniciar programa", "Asignar valor a variable", "Nuevo color lápiz", "Repetir" y "Finalizar programa" que son 5 instrucciones. Además tenemos que contar que hay 10 repeticiones y dentro de cada repetición 2 instrucciones, que suponen $10 \times 2 = 20$ instrucciones más, con lo cual llegamos a un total de 25 instrucciones ejecutadas.

En el programa PN00540G-4 tendríamos las instrucciones "Iniciar programa", "Asignar valor a variable", "Repetir" y "Finalizar programa" que son 4 instrucciones. Además tenemos que contar que hay 10 repeticiones y dentro de cada repetición 3 instrucciones, que suponen $10 \times 3 = 30$ instrucciones más, con lo cual llegamos a un total de 34 instrucciones ejecutadas.

Como cada instrucción sería como recorrer un kilómetro en coche, cuantas menos instrucciones gastemos para llegar a un resultado mejor. Por tanto el programa PN00540G-3 es más eficiente que el programa PN00540G-4 porque gasta menos instrucciones, a pesar de tener el mismo número de líneas.

¿Por qué ocurre esto? Porque hemos puesto dentro de un bloque de repetición una instrucción que en realidad no necesitamos repetir (la instrucción para establecer el color del lápiz). Esto hace que esta instrucción se repita una y otra vez cuando no era necesario. Cuando esto ocurre decimos que se están produciendo repeticiones innecesarias de una instrucción. En este caso, si ponemos la instrucción antes del bucle, conseguimos el mismo resultado y evitamos repeticiones innecesarias.

Como conclusión: ten cuidado con las instrucciones que colocas dentro de un bucle de repetición. Coloca dentro del bloque de repetición solo aquellas instrucciones realmente necesarias.

EJERCICIO

Dos programadores han creado estos códigos. Ejecútalos en la aplicación Didac-Prog Cartesia, comprueba que ambos llegan al mismo resultado y responde a las preguntas:

<pre>-- Programa creado por Andrés Pérez -- Profesor Arnaldo Cabrales -- Instituto Asunción de México A.C. -- Año 2049 Iniciar programa x nuevo valor es (-5) Repetir (700) veces ejecutar bloque Comienzo bloque Dibujar punto en (x, x^2) x nuevo valor es (x+0.25) Fin bloque Finalizar programa</pre>	<pre>-- Programa creado por Juan Solís Iniciar programa x nuevo valor es (-5) y nuevo valor es (25) Repetir (700) veces ejecutar bloque Comienzo bloque Dibujar punto en (x, x^2) x nuevo valor es (x+0.25) y nuevo valor es (x^2) Fin bloque Finalizar programa</pre>
---	--

a) Siguiendo las reglas para contar líneas de programas, ¿de cuántas líneas consta cada programa?

b) Por el número de líneas, ¿cuál sería el programa más eficiente?

- c) Cuenta el número de instrucciones que ejecuta cada programa de la misma forma que hemos hecho en el ejemplo visto anteriormente. ¿Cuántas instrucciones ejecutaría cada programa?
- d) ¿Cuál sería el programa más eficiente por número de líneas?
- e) ¿Cuál sería el programa más eficiente por instrucciones ejecutadas?
- f) Si tuvieras que proponer mejoras para hacer algún programa más eficiente ¿Cuál sería tu propuesta? ¿Por qué?

Puedes comprobar si tus respuestas son correctas consultando en los foros [aprenderaprogramar.com](https://www.aprenderaprogramar.com).

Próxima entrega: PN00541G

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

https://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=115&Itemid=311