



aprenderaprogramar.com

# Mejoras finales en programas: código, comentarios, diagramas de flujo y organización. (CU00249A)

Sección: Cursos

Categoría: Curso Bases de la programación Nivel II

Fecha revisión: 2024

Autor: Mario R. Rancel

Resumen: Entrega nº 48 del Curso Bases de la programación Nivel II

24

## MEJORAS FINALES. DOCUMENTACIÓN DEL PROGRAMA.

Nuestro automóvil está listo. Nos falta atar con una cinta unos cables sueltos, separar dos mangueras que están entrelazadas y elaborar dos documentos a partir de toda la información disponible: el manual del usuario y el manual del mecánico.

Nuestro programa está listo. Nos falta dar los últimos retoques a la presentación del código, ese módulo que ha quedado con las sangrías desordenadas, introducir algunas líneas en blanco para facilitar la lectura, esos archivos sueltos que vamos a dejar ordenados en directorios... Así como elaborar dos documentos a partir de toda la información disponible: el manual del usuario y el manual de mantenimiento, o si se prefiere, documentación para el usuario y documentación para el mantenimiento. Al igual que en un coche nos hace falta el manual para saber qué hace tal botón o cómo desmontar tal pieza, es necesario dejar documentación que nos informe del funcionamiento y posibilidades del programa así como de su estructura de cara al mantenimiento.

Terminada la mejora estética y antes de preparar la documentación conviene realizar las mejoras finales:

### 1.- Mejora de la presentación del código y sus comentarios.

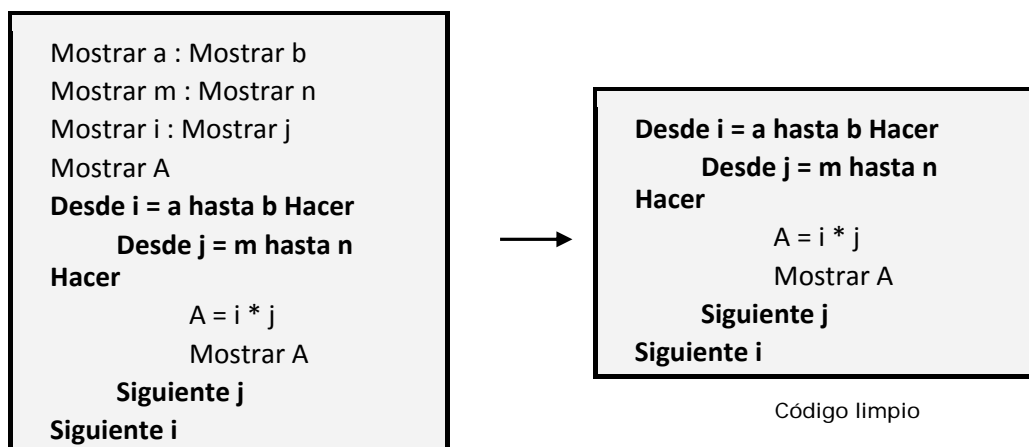
Muchas veces durante el desarrollo del programa hacemos comentarios abreviados o poco explícitos. Conviene reemplazar aquellos comentarios poco entendibles por otros claros y fáciles de entender.

Por ejemplo [Si a negativo peligro saltamos]

Cambiarlo por [Si a es negativo imposible calcular su raíz cuadrada. En este caso saltamos al módulo Transforma]

En otras ocasiones hay partes del código que se utilizaron para una verificación y que hay que eliminar en la versión definitiva del programa. Puede ocurrir que hasta el momento final hayamos estado haciendo verificaciones y tengamos el código de la verificación como comentario. Igualmente conviene eliminarlo: el programa ha de quedar limpio.

### Ejemplo:



Código contaminado por una verificación de variables y contadores

```
Desde i = a hasta b
Hacer
    [Mostrar i]
    PEC = i * b
    Mostrar PEC
Siguiente
```

Código contaminado por comentario transitorio no útil, en este caso un seguimiento de un contador conservado como comentario



```
Desde i = a hasta b
Hacer
    PEC = i * b
    Mostrar PEC
Siguiente
```

Código limpio

La presentación del código se revisa y mejora para facilitar su lectura, como si de un texto se tratase.

**Ejemplo:**

```
Si FOM > FUM Entonces
    Mostrar A
    Mostrar B
    Mostrar C
FinSi
Si FEM > FUM Entonces
    Mostrar FEM
FinSi
```



```
Si FOM > FUM Entonces
    Mostrar A
    Mostrar B
    Mostrar C
FinSi

Si FEM > FUM Entonces
    Mostrar FEM
FinSi
```

Se revisa en cuanto a sangrías, separación entre bloques, etc

## **2.- Organizar los módulos.**

Si existen módulos muy interrelacionados que se encuentran distanciados, convendrá agruparlos para facilitar la lectura y comprensión del programa. Igualmente el orden en que aparecen los módulos se intentará que sea lo más aproximado a la forma de ejecución del programa.

## **3.- Mejorar el diagrama de flujo.**

Si existe un diagrama de flujo, mejoramos igualmente su presentación: tamaño de letra y elementos gráficos adecuados, separación entre elementos adecuada, reorganización de páginas para evitar excesivos apelotonamientos o espacios en blanco, etc.

## **4.- Actualizar el esquema descendente o índice descendente.**

Comprobar que el esquema descendente refleje la estructura última del programa, incluyendo los últimos cambios que haya habido, y que su presentación es clara y correcta.

## **5.- Organizar los archivos.**

Si el programa consta o usa varios archivos, estos deberán quedar bien ordenados en directorios con nombres descriptivos que permitan una fácil localización y entendimiento del contenido.

Terminado este proceso de “pulimento” tenemos un producto que funciona, de agradable estética, ordenado y organizado. Sobre él podemos desarrollar la documentación del programa. En realidad debemos decir la documentación externa del programa, ya que el programa en sí mismo contiene información equivalente a documentación en forma de nombres descriptivos de variables, nombres descriptivos de módulos, comentarios, organización y claridad. A todo ello lo llamamos documentación intrínseca, interna o autodocumentación. La documentación interna es de gran importancia y no debe descuidarse pensando que con un buen manual de mantenimiento todo queda resuelto.

Podemos considerar una buena práctica de programación apoyar la facilidad de lectura y comprensión del programa en todas sus vertientes. No debemos sustituir información básica como el nombre descriptivo para una variable o módulo por un comentario que es información accesorio. Tampoco debemos sustituir un comentario necesario por una extensa y prolija explicación en un manual.

**Próxima entrega: CU00250A**

**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)