



## Estilo de programación

### Necesidad de un buen estilo

Conocer los detalles técnicos de uno o más lenguajes de programación no es suficiente garantía de crear buen código. El código de un programa no debe verse sólo como un conjunto de instrucciones, sino casi como literatura: está hecho por humanos y lo van a mantener (modificar cuando se precise) humanos. Por tanto, hay que escribirlo pensando en las personas y no en los compiladores. Éstos harán su trabajo para convertir el código fuente en ejecutable sin importarles cómo está escrito, bastará que sea correcto técnicamente; pero los humanos tendrán que leer el código y entender qué hace (y por qué) en el menor tiempo posible.



Así pues, es muy importante conocer unas pocas normas que orienten en la escritura de buen código. Para un principiante puede ser sorprendente que se le dé tanta importancia a un aspecto que luego no tendrá repercusión en el funcionamiento del programa, pero la experiencia de más de cincuenta años escribiendo programas ha llevado a los técnicos en la materia a reconocer que es un aspecto fundamental: a lo largo de la vida de un programa, la labor que más tiempo consume es su modificación, por eso es fundamental que esté bien escrito.



### Escritura del código

Hay que intentar escribir el código de modo que se entienda qué hace en el menor tiempo posible; debe tener físicamente buen aspecto, para facilitar la lectura: ni todo muy junto ni todo muy separado. Algunos consejos más concretos:

- ◆ **Líneas en blanco.** Hay que usar líneas en blanco para separar en el código las partes que sean lógicamente diferentes. Por ejemplo, separar la declaración de variables de una función y el código con instrucciones de ejecución.
- ◆ **Longitud de las líneas.** Nunca superior a 80 caracteres, para poder trabajar el código con cualquier editor e imprimirlo en cualquier impresora.
- ◆ **Indentación coherente.** Cuando se utilizan sentencias de control anidadas es fácil perderse y no saber a qué nivel pertenece cada porción de código. Eso se puede solventar indentando (es decir, metiendo un poco hacia la derecha) un poco más cada nivel respecto al anterior. Pero hay que hacerlo de forma coherente, siempre igual. Las llaves tan utilizadas en C deben colocarse siempre de la misma forma. Hay varios estilos de hacerlo: cada programador debe elegir el suyo y seguirlo fielmente.
- ◆ **Nombres de las variables.** Deben ser cortos pero descriptivos. Conviene pensarlos bien y no poner el primero que aparezca en la imaginación. Como se pueden mezclar mayúsculas y minúsculas en los nombres, hay que hacerlo coherentemente.



### Introducción de comentarios

En muchas ocasiones el código solo es difícil de entender, aunque esté bien escrito. Entonces hay que añadir comentarios que faciliten la comprensión. Otras veces se utilizan comentarios simplemente para añadir información extra al código, como las partes de que se compone el fichero, quién lo ha escrito, etc. Consejos más precisos:

- ◆ En los comentarios debe escribirse qué hace el código, no cómo lo hace, puesto que eso es justo lo que ya se ve.
- ◆ Aunque se escriban comentarios, hay que procurar que el código sea legible por sí mismo.
- ◆ No deben entorpecer la lectura del código, sino mejorarla.

- ◆ Deben escribirse con coherencia, no unas partes con muchos comentarios y otras sin ninguno.
- ◆ Sólo hay que escribir comentarios que sean realmente útiles y correctos: el que no es útil, molesta; el que no es correcto, equivoca.
- ◆ Conviene añadir comentarios físicamente evidentes para distinguir con más facilidad las distintas partes de un programa: ficheros de cabecera, declaración de funciones, etc.
- ◆ Es imprescindible que haya comentarios con los datos básicos del programa: autor, fecha de creación, explicaciones generales, etc.
- ◆ Cada función debe llevar un comentario que la anteceda, especificando sus entradas, salidas, y cualquier otro dato que sea pertinente.

