



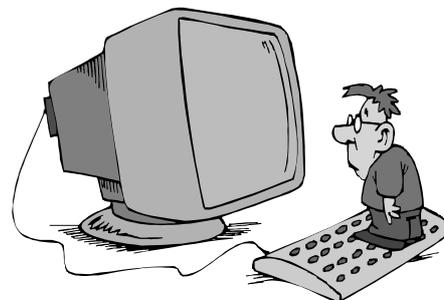
Ciclo de vida

Ciclo de vida de un programa

Se llama así al conjunto de fases por las que pasa un programa, desde que no existe hasta que deja de usarse. Para crear un buen programa es imprescindible conocer estas fases, y respetarlas. Por supuesto, para hacer un programa muy pequeño no es necesario ser tan estricto, puesto que los programas pequeños apenas requieren esfuerzo, pero cuando los proyectos de programación van siendo mayores, enseguida se nota la necesidad de una buena planificación.

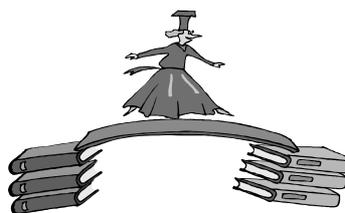
Nunca se debe pasar a la siguiente fase sin haber concluido la anterior. Se ahorra mucho tiempo trabajando así, porque, aunque el software es muy flexible, modificarlo cuando está escrito es muchísimo más complicado que hacerlo cuando todavía no se ha hecho.

No todas las fases se realizan en el ordenador: de hecho, las partes más importantes se llevan a cabo con ordenadores apagados. Normalmente se piensa que en informática se comienza cualquier tarea encendiendo el ordenador, sin embargo en programación se comienza apagándolo.



Fases del ciclo

- ◆ Análisis de requerimientos.
- ◆ Diseño.
- ◆ Codificación.
- ◆ Depuración.
- ◆ Explotación.



Análisis de requerimientos

Para poder escribir un programa antes hay que saber qué se espera de él, cuál será su cometido. Por tanto, hay que comenzar por estudiar qué se desea exactamente que haga el programa. Esto se llama analizar los requerimientos, y en casos reales lo realizan los analistas de sistemas.

Hay que llegar a definir con la mayor exactitud posible qué entradas recibirá el programa, y qué salidas se espera de él.

Diseño

A partir del análisis de requerimientos se trabaja en el diseño del programa, pensando en qué partes habrá que dividirlo, así como qué estructuras serán necesarias para poder llevar a cabo todo lo que se pide.

Hay que procurar que haya partes claramente distinguibles, con responsabilidades bien definidas; se implementarán en funciones independientes. Se procura usar pocas variables globales, porque suelen ser fuente de errores.



Codificación

Si el programa está bien diseñado, codificarlo es bastante sencillo; pero si el diseño es pobre y se han pasado por alto muchos detalles, la codificación puede consumir mucho tiempo.

Hay que dividir el código en funciones pequeñas, para que sea fácil de entender y modificar. Cada función debe hacer una sola cosa.

Depuración

Mientras se va escribiendo el código, hay que ir comprobando si funciona bien. No hay que esperar a que esté todo escrito, porque entonces habrá demasiados errores. Hay que ir detectando errores y corrigiéndolos.

Esta etapa se llama en inglés *debug*, “eliminar bichos”, ya que en uno de los primeros ordenadores un error estaba causado por un insecto que se había quedado enredado en los cables.

Explotación

Esta es la fase en la que se usa el programa. Es habitual que durante esta fase aún se detecten más errores y haya que arreglarlos o bien tomar nota de ellos para corregirlos en la siguiente versión del programa.



Nuevas versiones

Cuando se decide que aparezca la nueva versión de un programa, el ciclo de vida comienza otra vez, pero con una particularidad: la mayor parte del código anterior se conserva o se adapta.

Se calcula que la mayoría del tiempo de programación se pasa retocando código, bien para arreglarlo o para mejorarlo; esto demuestra la importancia de prestar mucha atención a una codificación con buen estilo.

