

CARACTERÍSTICAS DE LOS ALGORITMOS

- **Finitos:** Debe acabar en algún momento.
- **Eficientes:** Deben ocupar la mínima memoria y minimizar el tiempo de ejecución.
- **Legibles:** El texto que lo describe debe ser claro, tal que permita entenderlo y leerlo fácilmente.
- **Modificables:** Estarán diseñados de modo que sus posteriores modificaciones sean fáciles de realizar, incluso por programadores diferentes a sus propios autores.
- **Modulares:** La filosofía utilizada para su diseño debe favorecer la división del problema en módulos pequeños.
- **Único punto de entrada, único punto de salida:** A los algoritmos y a los módulos que lo integran se entra por un sólo punto, inicio, y se sale por un sólo punto también, fin.

LA ESTRUCTURA GENERAL PARA LA RESOLUCIÓN INFORMÁTICA DE UN PROBLEMA SERÁ LA SIGUIENTE:

1.- Análisis del problema.

Haremos un estudio previo de la tarea a realizar. Si nos la dan verbalmente, haremos preguntas hasta tener claro lo que se ha de hacer, es decir, hasta obtener la **escritura detallada de todas las especificaciones**. Si la tarea nos la dan por escrito, podemos comenzar subrayando palabras o frases decisivas, escribiendo al margen ideas y dudas, o cualquier otra cosa que nos ayude a concretar especificaciones, además de realizar verbalmente todas las preguntas que den respuesta a nuestras dudas. Algunas preguntas elementales serán las siguientes:

- ¿Cuáles serán los datos de entrada?.
- ¿Qué datos se piden de salida?.
- ¿Qué apariencia (formato) tendrán los datos?.
- ¿Cuántas veces se deben repetir los procesos?.
- ¿Hay que hacer suposiciones?. En caso afirmativo se hará un listado de ellas.
- ¿Pueden aparecer condiciones especiales de error?
- ¿Hemos resuelto algún problema parecido o sabemos que existe una solución para una tarea análoga?. Si la respuesta es afirmativa usemos esa solución, aunque hubiera que modificarla.
- ¿Hay juegos de ensayo adecuados?. Búsquelos.

2.- Escribir en pseudocódigo el programa principal.

Empezaremos dividiendo el problema en módulos más sencillos, hasta llegar a un nivel de detalle razonable utilizando nombres con significado para ellos. Si de momento no sabemos resolver alguno de estos módulos, no debemos preocuparnos pensemos que alguien nos lo resolverá o que mañana tendremos una idea genial para solucionarlo y sigamos adelante con nuestro diseño, tal vez tengamos la suerte de encontrarlo ya hecho. Recordemos que si los problemas son razonablemente pequeños, como ocurrirá en los primeros temas no será necesaria esta división en subtareas del problema.

2.- Escribir en pseudocódigo los módulos restantes.

Escribiremos cada uno de los módulos, detallando la secuencia de instrucciones que deben realizar, en el caso de que los hubiera.

3.- Lectura del pseudocódigo.

Recorrer el algoritmo escrito en pseudocódigo inspeccionando, reordenando, detectando y corrigiendo errores, revisando todo lo que sea necesario, incluso será posible que haya que planificar algún cambio.

No hay que tener miedo a empezar de nuevo si fuera necesario.

4.- Seguir la traza.

Haremos la prueba del algoritmo siguiendo la traza con los datos elegidos como juego de ensayo. Observaremos los resultados minuciosamente, tratando de detectar y corregir los errores si estos resultados no son los esperados.

5.- Documentación.

Una vez diseñado y bien estructurado un algoritmo, la siguiente actividad a desarrollar, sumamente importante en programación, es producir programas bien documentados, para que se cumplan los objetivos básicos, que los programas sean legibles, comprensibles y fácilmente modificables. La documentación incluye descripciones, comentarios, especificaciones, incluso un breve manual de uso si el algoritmo es suficientemente extenso.

Tenemos dos tipos de documentación en el desarrollo de algoritmos bien diseñados:

- a) **Documentación externa:** Incluye la escritura de información que es exterior al cuerpo del algoritmo. Puede incluir, además de todas las especificaciones detalladas, una explicación extensa del desarrollo del algoritmo y de las modificaciones sufridas. Deben añadirse igualmente descripciones detalladas del problema, de los módulos que lo componen, adjuntando un diagrama jerárquico del diseño descendente y por último un manual de usuario donde se explicará al cliente (o al profesor en su defecto) como debe usarse el programa. Esta documentación debe adjuntarse sólo en el caso de que los algoritmos sean suficientemente grandes, lo que se consideran *aplicaciones informáticas* de cierta envergadura, divididas en varios programas.
- b) **Documentación interna:** Incluye la escritura de información que se suministra con el cuerpo del algoritmo. Estará formada por comentarios, código autodocumentado e impresión agradable del texto del pseudocódigo.