

Programación modular

Hasta ahora, en los videos sobre las clases hemos escrito programas que resuelven problemas relativamente pequeños y de manera lineal. Para ello hemos escrito instrucciones en secuencia dentro de un archivo que termina con la extensión **.cpp**.

Cuando el problema a resolver es considerablemente grande, igualmente lo será la solución del mismo. Si un problema es suficientemente complejo, de igual forma lo será la solución que debe implementarse. Esto puede volver la programación bastante engorrosa, así que lo que se aplica en estos casos es lo que se conoce como **programación modular** o **"divide y vencerás"**.

Si un problema es suficientemente grande y complejo, lo que se realiza para facilitar la programación de la solución es dividir el problema original en problemas más pequeños. Es decir, se identifican las diferentes partes del problema: se separa en sub problemas, para luego programar la solución de cada uno de estos sub problemas identificados. Así, analizamos el problema **llendo de lo general a lo particular**, y así también planteamos su solución. Por tanto, la unión de las soluciones de los sub problemas, es la solución del problema original.

Este tipo de programación tiene la característica de que, dado que los diferentes módulo son independiente entre sí, pueden ser asignados a diferentes programadores y así terminar el proceso de programación en menos tiempo. También permite que se alteren las instrucciones de un módulo sin que el programa completo se vea afectado. Así que es uno de los métodos más potentes para mejorar la productividad del equipo de programadores.

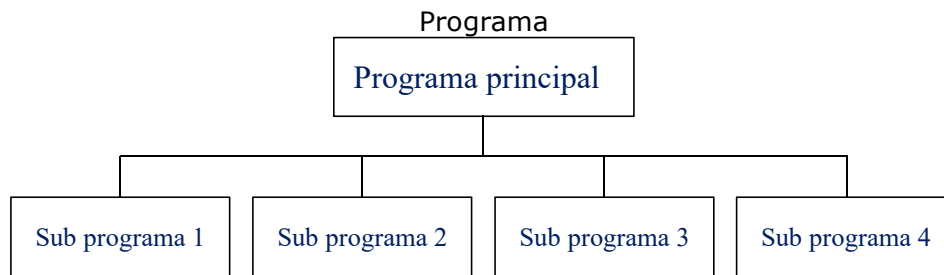
Cada programa contiene un módulo especial llamado programa principal, que controla todo lo que sucede a través de ejecutar algunas instrucciones y transferir el control a los otros módulos o sub programas, los cuales regresarán el control de nuevo al programa principal.

Si un módulo tiene a cargo la resolución de una tarea compleja, éste podrá también dividirse en sub módulos hasta lograr que cada uno tenga solamente una tarea sencilla que resolver. Las tareas más comunes son: recepción de entradas, manipulación de datos, control de módulos, entrega de salida y alguna combinación de las anteriores.

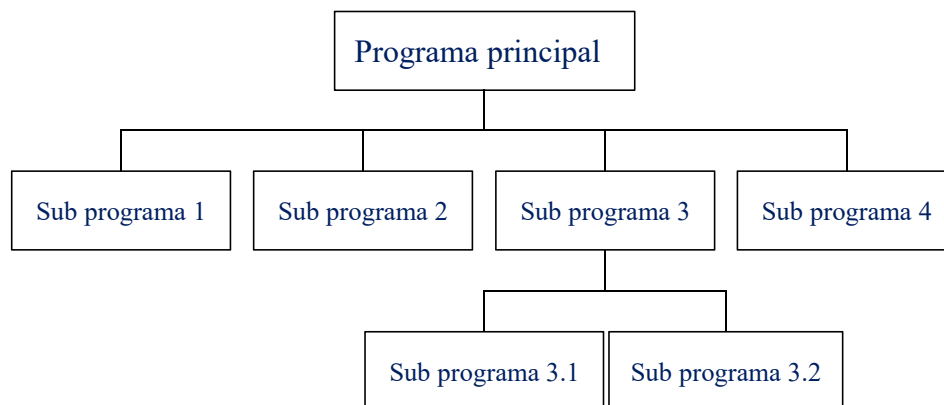
Por ejemplo, si se tiene el siguiente problema, del cual se han identificado cuatro partes que lo componen:

Problema	
Sub problema 1	Sub problema 2
Sub problema 3	
Sub problema 4	

Entonces podríamos esquematizar la solución del problema así:



Si a su vez alguno de los sub problemas fuera lo suficientemente complejo como para dividirse a sí mismo en sub problemas, lo recomendable es hacerlo. Por ejemplo, si el sub problema 3 de la figura se puede dividir en más sub problemas, podríamos tener algo como lo que se ve en la siguiente figura:



Donde cada uno de los sub problemas de la figura es de una complejidad muchísimo menor que el problema original en su conjunto.

La programación modular nos proporciona muchos beneficios, entre ellos podemos mencionar:

- a) Hace menos engorrosa la programación.
- b) Ayuda a ordenar el pensamiento.
- c) Permite concentrarse en una pequeña tarea a la vez, haciendo a un lado el resto del problema.
- d) Permite repartir el trabajo en un equipo de programadores, cada quien solo necesita saber que recibe su módulo, o rutina, y qué debe retornar.
- e) La programación repetitiva de tareas similares se reduce a un solo bloque de código, reduciendo el tamaño del programa.