

Tema 4

Metodología De Desarrollo De Programas (I)

La programación como resolución de problemas

La labor de programación puede considerarse como un caso particular de la resolución de problemas. Resolver un problema consiste esencialmente en encontrar una estrategia a seguir para conseguir la solución. Una estrategia se expresará como una colección de reglas o recomendaciones que, si se siguen, conducirán a la solución.

Descomposición de un problema en subproblemas

El método más general de resolución de problemas no triviales consiste en descomponer el problema original en subproblemas más sencillos, continuando el proceso hasta llegar a subproblemas que puedan ser resueltos en forma directa.

Desarrollo de programas por refinamientos sucesivos

La técnica de *refinamientos sucesivos* consiste en expresar inicialmente el programa a desarrollar como una acción global, que si es necesario se irá descomponiendo en acciones más sencillas hasta llegar a acciones simples, que pueden ser expresadas directamente como sentencias del lenguaje de programación.

La forma en que varias acciones se combinan en una acción compuesta constituye el *esquema o estructura* de la acción compuesta. El *esquema secuencial* consiste en realizar una acción compuesta a base de realizar una tras otra, en secuencia, dos o más acciones componentes.

- **Desarrollo de un esquema secuencial:** La metodología de refinamientos incluye el ir desarrollando a la vez las sentencias del programa que realizan las acciones de la parte ejecutable, y la definición de las variables necesarias para almacenar la información manipulada por dichas acciones. Para desarrollar una acción compuesta según el esquema secuencial se necesitará:

1. Identificar las acciones componentes de la secuencia. Identificar las variables necesarias para disponer de la información adecuada al comienzo de cada acción, y almacenar el resultado.
2. Identificar el orden en que deben ejecutarse dichas acciones.

Aspectos de estilo

Una buena metodología de desarrollo de programas debe atender no sólo a cómo se van refinando las sucesivas acciones, sino a cómo se expresan las acciones finales en el lenguaje de programación. El estilo de redacción del programa en su forma final es algo fundamental para conseguir que sea claro y fácilmente comprensible por parte de quienes hayan de leerlo.

- **Encolumnado:** Un programa aparece como un texto. El estilo de presentación de dicho texto o documento debe destacar claramente su organización en partes.

Un recurso de estilo de presentación es el sangrado o *encolumnado*, el cual se consigue ampliando el margen izquierdo de las partes internas de un elemento compuesto.

- **Comentarios. Documentación del refinamiento:** Otro recurso utilizable para mejorar la claridad de un programa es el empleo de comentarios. Esto se consigue escribiéndolos entre los símbolos (* y *). Algunas clases de comentarios son:

- Cabeceras de programa.
- Cabeceras de sección.
- Comentarios-orden.
- Comentarios al margen.

- **Elección de nombres:** Otro aspecto de estilo, fundamental para la claridad de un programa, es la elección correcta de los nombres o identificadores utilizados para designar sus diferentes elementos. Los nombres que haya de inventar el programador deben ser elegidos con un criterio mnemotécnico, de manera que recuerden fácilmente el significado de los elementos nombrados.
Para que los nombres o identificadores resulten significativos hay que procurar que tengan la categoría gramatical adecuada al elemento nombrado. En concreto:
 - Los valores (constantes, variable, etc.) deben ser designados mediante sustantivos.
 - Las acciones (procedimientos, etc.) deben ser designadas mediante verbos.
 - Los tipos deben ser designados mediante nombres genéricos.
- **Uso de letras mayúsculas y minúsculas:** Se puede marcar claramente a que clase de elemento del programa se está refiriendo un nombre, escribiéndolos en una combinación de mayúsculas y minúsculas.
- **Constantes con nombre:** La posibilidad de declarar constantes con nombres simbólicos puede aprovecharse para mejorar la claridad del programa. En lugar de usar directamente valores numéricos en las expresiones de algunos cálculos, puede resultar ventajoso definir determinados coeficientes o factores de conversión con un nombre simbólico que tenga un buen significado mnemotécnico, y usar la constante con ese nombre en los cálculos.