Recuerde que estos son apuntes muy simplificados que deberá completar con la bibliografía recomendada

PROGRAMACIÓN 10 Prof. Dolores Cuiñas H.
APUNTES Nº 1.
METODOLOGÍA PARA LA CONSTRUCCIÓN DE PROGRAMAS.

Presentaremos de forma muy general los principales pasos que se deben seguir para resolver problemas aplicando técnicas de programación. Esta metodología será desarrollada a lo largo de todo el curso, en la medida que se estudien las restantes unidades del mismo.

1. FORMULACIÓN Y ANÁLISIS DEL PROBLEMA

Consiste en entender de qué se trata el problema planteado y esbozar su posible solución, concluyendo con una clara definición de tres aspectos: 1º qué es lo que nos piden, es decir, definición del resultado o solución deseada (**para qué**). 2º cómo obtener lo que nos piden (**qué hacer**). 3º qué necesitamos para obtener los resultados pedidos (**con qué**). Esto último nos facilitará la construcción de lo que denominaremos Especificación Funcional.

1.1.- Especificación Funcional: Consiste en determinar las funciones que se van a realizar (**qué hacer**) y sus respectivas entradas (**con qué**) y salidas (**para qué**):

Donde: **entrada** son los argumentos (variables o constantes) que se requieren para resolver un problema, **salida** son los resultados (argumentos) que se desean obtener una vez resuelto el problema y **proceso** es el procedimiento(s) u operación(es) que deben efectuarse sobre las entradas para obtener las salidas deseadas.

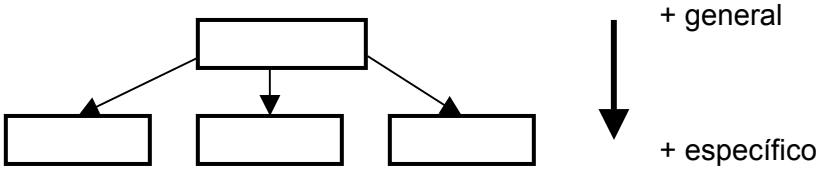
1.2.- Especificación de los Argumentos o Parámetros: Consiste en la documentación de los argumentos o parámetros (sean estos de entrada, salida o intermedios) requeridos en la solución del problema, mediante la elaboración de una tabla que contemple los siguientes aspectos:

descripción del argumento	identificador	tipo	longitud	condición	restricción

1.3.- Establecimiento de Restricciones y Atributos: Consiste en determinar bajo qué restricciones se ha de operar y cuales son las medidas de rendimiento y calidad que debe tener el sistema (programa). Este aspecto no se realizará en este curso dada la simplicidad de los problemas tratados.

2. DISEÑO.

Consiste en diseñar **cómo** hace el programa la tarea solicitada. En forma general consiste en dividir el programa en subprogramas y cada subprograma en módulos.



Este tópico se tratará con mayor detalle en la **unidad 6**. Inicialmente **no** dividiremos el programa en subprogramas.

El criterio de descomposición más utilizado es el de tipo funcional, el cual produce una estructura jerárquica en la que cada módulo ejecuta una o más funciones y para cada módulo se produce una **especificación de programa o módulo**, la cual contiene lo siguiente:

Nombre del Programa o Módulo.
Función que desarrolla.

Parámetros o Argumentos.
Parámetros o Argumentos de Entrada.
Parámetros o Argumentos de Salida.
Estructura de Datos Requerida.
Lenguaje de Programación.
Algoritmo

Estructura de datos no se utilizará en este curso dado que el mismo es dado a nivel introductorio

Donde **ALGORITMO** es un conjunto finito de pasos en secuencia que indican como se resuelve un determinado problema.

Propiedades de los algoritmos estructurales:

a) Número finito de pasos sin ambigüedades.
b) Numeración de cada paso en orden secuencial
c) La acción a realizar se indica con un verbo o con un gráfico.
d) Condicionado a las estructuras básicas de la programación estructurada.
e) Eficiente (menor número de pasos pero lo más claro posible)

Herramientas para diseñar algoritmos:

a) Diagramas de Flujo: representación gráfica de un algoritmo.
b) Pseudocódigo: lenguaje de especificación de algoritmos (el algoritmo se representa mediante palabras similares al inglés o al español, para facilitar tanto la lectura como la escritura de programas.

Los símbolos o la notación para elaborar, respectivamente, los diagramas de flujo o los pseudocódigos, serán indicados en los siguientes apuntes, en la medida que se estudie las correspondientes estructuras de programación.

3. CODIFICACIÓN.

Es la escritura en un lenguaje de programación de la representación del algoritmo desarrollado en la etapa de diseño. El resultado de la codificación es un programa fuente.

4. COMPILACIÓN Y EJECUCIÓN.

Es el proceso de traducción del programa fuente al lenguaje de máquina. Este proceso se realiza con el compilador y el Sistema Operativo. El resultado, si no hay errores, es la obtención del programa objeto que todavía no es ejecutable directamente. Luego, mediante el Sistema Operativo se realiza la carga del programa objeto con las librerías del programa compilador, el resultado es un programa ejecutable. Cuando el programa ejecutable se ha creado, se puede ejecutar el programa desde el Sistema Operativo generalmente con sólo teclear su nombre. Si no hay errores se obtiene como salida los resultados del programa. **NOTA**: El Turbo Pascal compila y ejecuta con una sola orden.

5. VERIFICACIÓN Y DEPURACIÓN.

Es el proceso de probar que el programa trabaje correctamente y cumpla con los requerimientos del usuario.
VERIFICAR: es el proceso de ejecución del programa con una amplia variedad de datos de entrada (test o pruebas) que determinarán si el programa tiene errores.
DEPURAR: es el proceso de encontrar los errores del programa y corregir o eliminar dichos errores.

TIPOS DE ERRORES:

a) Errores de compilación: suelen ser errores de sintaxis.
b) Errores de ejecución: suelen ser instrucciones que la computadora comprende pero que no puede ejecutar. Ejemplo: divisiones por cero, raíces negativas, etc.
c) Errores de lógica: se producen en la lógica del programa, en el diseño del algoritmo. Se detectan porque los resultados son incorrectos.

6. DOCUMENTACIÓN Y MANTENIMIENTO.

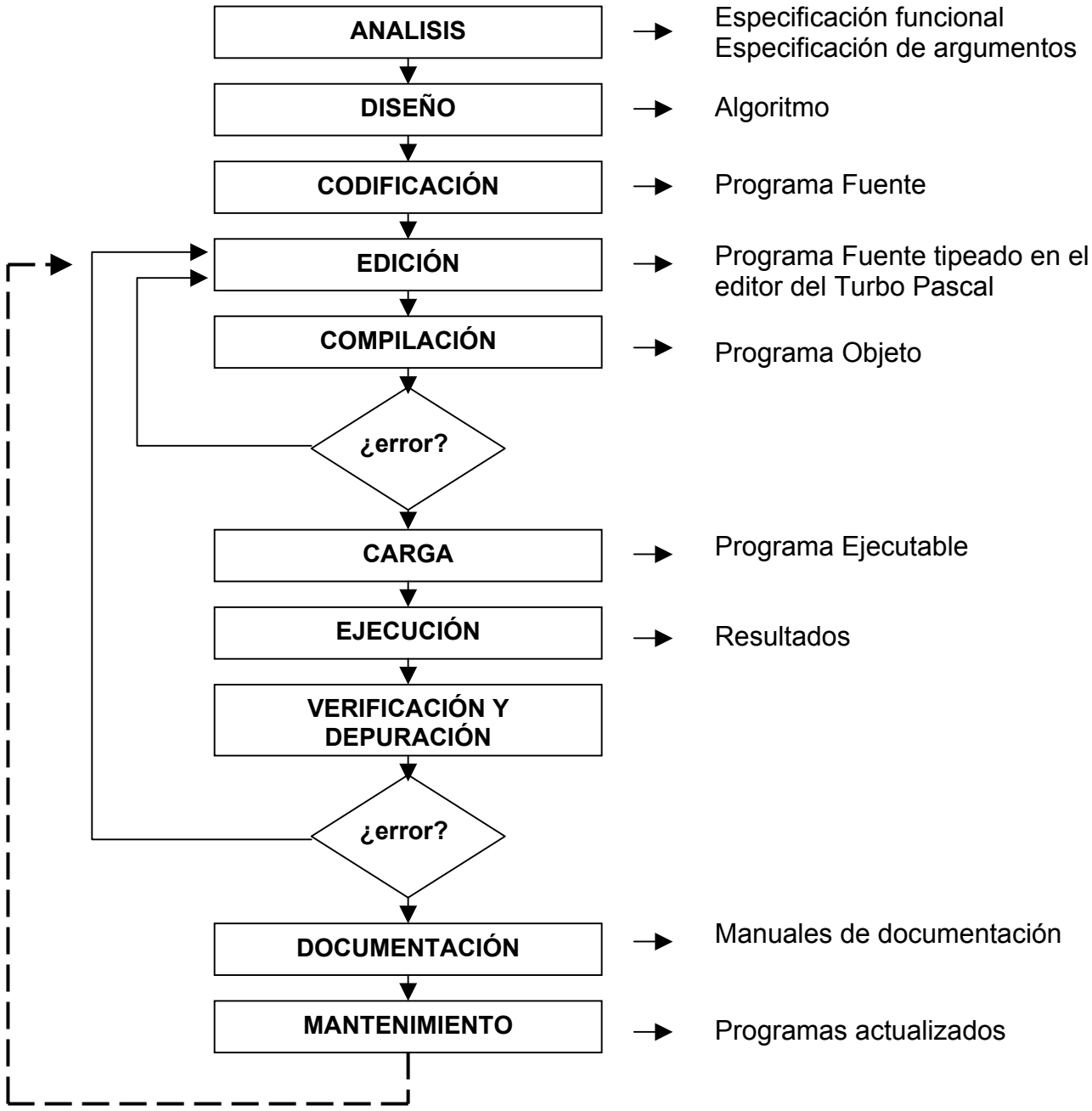
Consta de la descripción de los pasos a dar en el proceso de resolución de un problema. La documentación de un programa puede ser:

Interna: contenida en las líneas de comentarios del propio programa.

Externa: Incluye el análisis, la especificación del programa, el algoritmo, los manuales de los usuarios, etc.

El mantenimiento consiste en la actualización de los programas con los cambios requeridos por el usuario o corrección de posibles errores futuros.

Gráficamente la metodología puede ilustrarse de la siguiente manera:



NOTA: Esta metodología es una simplificación de la utilizada para el desarrollo de sistemas más complejos. En este curso será aplicada aún más esquemáticamente tal como se plantea en el siguiente ejemplo y en los restantes explicados en clase.

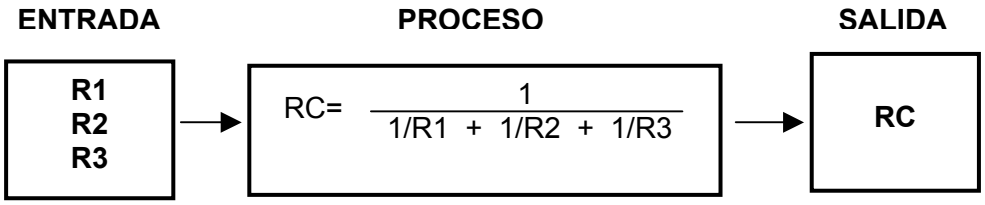
EJERCICIO PARA ILUSTRAR LA METODOLOGÍA .

1. Planteamiento del problema: calcular la resistencia combinada cuando tres resistencias (R1, R2 y R3) están conectadas en paralelo. La fórmula de la resistencia combinada es:

$$\frac{1}{1/R1 + 1/R2 + 1/R3}$$

El programa debe producir la siguiente salida: Resistencia Combinada en Ohmios es: xxxxxx

2. Análisis.
   a) Especificación funcional: suponiendo que llamo **RC** a la Resistencia Combinada, del análisis deduzco que requiero de **R1, R2** y **R3** (entradas) para poder realizar el cálculo (proceso) de la **RC** (salida).



   b) Especificación de los argumentos: se trata de la documentación de los argumentos utilizados.

argumentos	identificador	tipo	longitud	condición	restricción
Resistencia 1	R1	real	5.1	variable	>0
Resistencia 2	R2	real	5.1	variable	>0
Resistencia 3	R3	real	5.1	variable	>0
Resistencia Combinada	RC	real	7.1	variable	>0

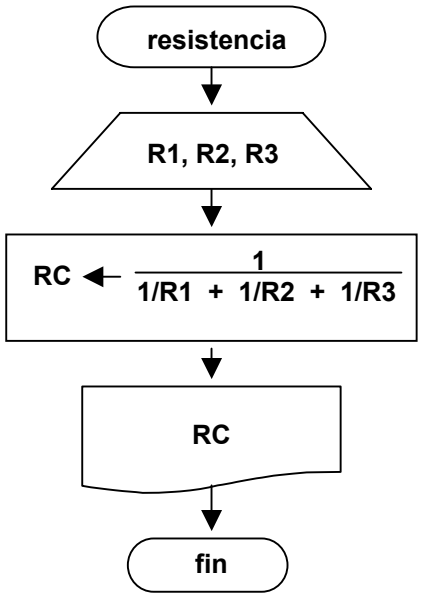
3. Diseño.

Especificación del Programa:
Nombre del Programa: resistencia
Función: Calcular la Resistencia Combinada de 3 resistencias conectadas en paralelo-
Argumentos: R1, R2, R3, RC
Argumentos de Entrada: R1, R2, R3
Argumentos de Salida: RC
Lenguaje de Programación: Turbo Pascal
Algoritmo: (nota: puede utilizarse pseudocódigo o diagrama de flujo).

Pseudocódigo o algoritmo escrito:

1. Comienzo (resistencia)
2. Leer (R1, R2, R3)
3. RC ← 1/((1/R1)+(1/R2)+(1/R3))
4. Escribir (Resistencia Combinada en Ohnios es:, RC)
5. Fin (resistencia)

Diagrama de flujo o algoritmo gráfico:



4. Codificación en Turbo Pascal

```
PROGRAM RESISTENCIA;
  {Programa escrito por: xxxxxxxxxxxxxxxxx}
  {Fecha: xx / xx / xx}
  {Este programa calcula la resistencia combinada de tres resistencias conectadas en paralelo}
USES
    CRT;
VAR
    R1,R2,R3,RC:REAL;
BEGIN
    CLRSCR;
    WRITELN('INTRODUZCA EL VALOR DE R1, R2 Y R3 CONSECUTIVAMENTE');
    READLN(R1,R2,R3);
    RC:= 1/(1/R1 + 1/R2 + 1/R3);   {CALCULO DE LA RESISTENCIA COMBINADA}
    CLRSCR;
    WRITELN('LA RESISTENCIA COMBINADA EN OHMIOS ES=',RC:8:1)
END.
```

5. Edición: tipear el programa anterior (programa fuente) en el computador.
6. Compilación y Ejecución del programa editado: para la obtención de los resultados previa corrección de posibles errores de sintaxis o de ejecución.
7. Verificacación y Depuración de los resultados: para corregir los posibles errores de lógica que puedan existir.
8. Documentación.
9. Mantenimiento.

NOTA: Los conceptos anteriores son tomados básicamente de:

- Programación en Turbo/Borland. Pascal 7. Luis Joyanes Aguilar
- Diseño de Algoritmos. Isabel Besembel
- Metodología para el Desarrollo de Sistemas. Jonas Montilva