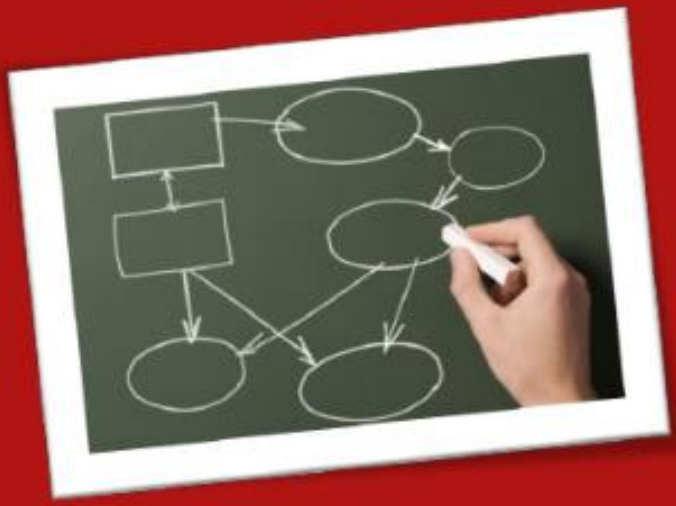
Universidad Politécnica de la Región Ribereña



Algoritmos, Diagramas de Flujo y Pseudocódigos.

Cd. Miguel Alemán a 07 de septiembre de 2015.

LÓGICA DE PROGRAMACÓN

PROFESOR MCA. IEC. HERIBERTO RENÉ SALDAÑA SALDAÑA.

ALUMNO GAMALIEL MUÑOZ HERNÁNDEZ.

INGENIERÍA INDUSTRIAL

GENERACIÓN 14.

4to CUATRIMESTRE.

Índice

INTRODUCCIÓN

El estudio de la Lógica de Programación no exige ningún conocimiento previo de computadores ni de tecnología en general, tampoco exige la presencia de algún lenguaje de programación específico aunque no se puede negar que éste podría permitirle implementar y ver convertida en realidad las soluciones lógicas a sus objetivos, en el siguiente trabajo de Investigación Documental, se abordarán temas tales como los Algoritmos, dentro de los cuales destacaremos su definición, sus características, sus partes entre otras cosas, también se hablará acerca de los Diagramas de Flujo, destacando el objetivo para lo que están hechos, como se conforman y las reglas que se deben seguir para su estructuración, por ultimo también se investigó acerca de los Pseudocódigos, ¿Qué son? y ¿Para qué sirven?, aunque como se menciona al principio que no se exige un previo conocimiento de computadoras, es necesario realizar a cabo esta investigación para poder entender algunos de los temas básicos de la lógica de programación, ya que muchas personas confunden la Programación con la Lógica de Programación, la primera involucra el conocimiento de técnicas e instrucciones de un determinado Lenguaje a través de los cuales se hace sencillo lograr que la Computadora obtenga unos resultados mucho más rápidos que una persona. La segunda involucra, de una manera técnica y organizada, los conceptos que permiten diseñar en términos generales, la solución a problemas que pueden llegar a ser implementados a través de una computadora.

1.1 ALGORITMOS

DEFINICIÓN:

Un Algoritmo, se puede definir como una secuencia de instrucciones que representan un modelo de solución para determinado tipo de problemas. O bien, también como un conjunto de instrucciones que realizadas en orden conducen a obtener la solución de un problema. Por lo tanto, podemos decir que es un conjunto ordenado y finito de pasos que nos permite solucionar un problema.

Fig. 1.1 El algoritmo es la infraestructura de cualquier solución, escrita en cualquier

Los algoritmos son independientes de los lenguajes de programación. En cada problema el algoritmo puede escribirse y luego ejecutarse en un lenguaje de diferente programación. (Fig. 1.1)

PROGRAMA:

Un programa es una serie de instrucciones ordenadas, codificadas en lenguaje de programación que expresa un algoritmo yque puede ser ejecutado en un computador.

CLASIFICACIÓN DE ALGORITMOS:

Los algoritmos se pueden clasificar en cuatro tipos:

Algoritmo computacional:

Es un algoritmo que puede ser ejecutado en una computadora. Ejemplo: Fórmula aplicada para un cálculo de la raíz cuadrada de un valor x. (fig. 1.2)

Fig. 1.2 algoritmo computacional

Algoritmo no computacional:

Es un algoritmo que no requiere de una computadora para ser ejecutado. Ejemplo: Instalación de un equipo de sonido.

Algoritmo cualitativo:

Un algoritmo es cualitativo cuando en sus pasos o instrucciones no están involucrados cálculos numéricos. Ejemplos: las instrucciones para desarrollar una actividad física, encontrar un tesoro.

Algoritmo cuantitativo:

Un algoritmo es cuantitativo cuando en sus pasos o instrucciones involucran cálculos. Ejemplo: Solución de una ecuación de segundo grado. (fig. 1.3)

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

Fig. 1.3 Algoritmo cuantitativo

CARACTERISTICAS DE UN ALGORITMO:

Todos los algoritmos deben tener las siguientes características (fig. 1.4):

Fig. 1.4 Características

- **Debe ser Preciso**, porque cada uno de sus pasos debe indicar de manera precisa e inequívoca que se debe de hacer.
- **Debe ser Finito,** porque un algoritmo debe tener un número limitado de pasos.
- **Debe ser Definido,** porque debe producir los mismos resultados para las mismas condiciones de entrada.
- **Puede tener** cero o más elementos de entrada.
- **Debe producir un resultado**. Los datos de salida serán los resultados de efectuar las instrucciones.

PARTES DE UN ALGORITMO:

Todo Algoritmo debe de tener las siguientes partes:

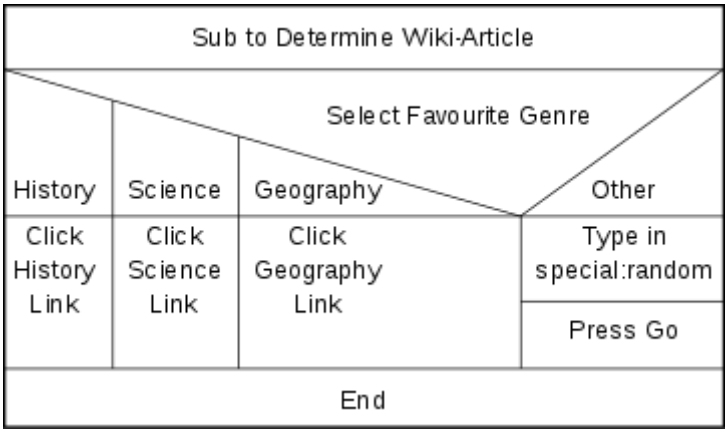
- **Entrada de datos,** son los datos necesarios que el algoritmo necesita para ser ejecutado.
- **Proceso**, es la secuencia de pasos para ejecutar el algoritmo.
- **Salida de resultados**, son los datos obtenidos después de la ejecución del algoritmo.

Fig. 1.5 Partes de un algoritmo

TECNICA DE REPRESENTACIÓN:

Para la representación de un algoritmo, antes de ser convertido a lenguaje de programación, se utilizan algunos métodos de representación escrita, gráfica o matemática. Los métodos más conocidos son:

- Diagramación libre (Diagramas de flujo).
- Diagrama Nassi-Shneiderman.
- Pseudocódigo.
- Lenguaje natural (español, inglés, etc.)
- Fórmulas matemáticas.



Ejemplo de Diagrama Nassi-Shneiderman.

1.2 DIAGRAMAS DE FLUJO

Un DIAGRAMA DE FLUJO es una representación gráfica de un proceso. Cada paso del proceso es representado por un símbolo diferente que contiene una breve descripción de la etapa del proceso. (fig. 16)

OBJETIVOS DEL DIAGRAMA DE FLUJO:

Los diagramas de flujo tienen como objetivos:

- ✓ Ofrecer una descripción visual de las actividades de un proceso mostrando la relación secuencial ante ellas.
- ✓ Facilitar la rápida comprensión de cada actividad y su relación con las demás, el flujo de la información, las ramas en el proceso, el número de pasos del proceso, etc.
- ✓ Facilitar la sección de indicadores de proceso.
- ✓ Estimula el pensamiento analítico en el momento de estudiar un proceso, haciendo más factible generar alternativas útiles.

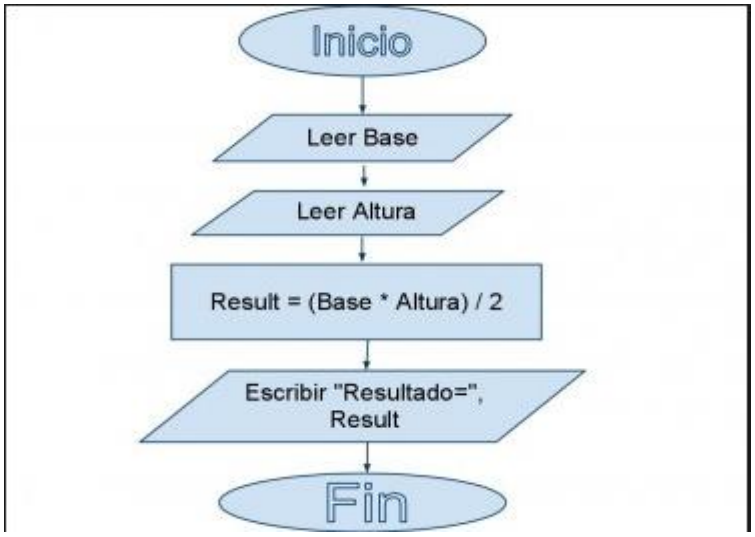


Fig. 16 diagrama de flujo

- ✓ Un diagrama de flujo ayuda a establecer el valor agregado de cada una de las actividades que compone el proceso.

TIPOS DE DIAGRAMA DE FLUJO

Hay varios tipos de distintos de flujograma que pueden usarse:

- Flujograma de primer nivel o de dirección descendente

Un flujograma de primer nivel muestra los pasos principales de un proceso y puede incluir también los resultados intermedios de cada paso y los sub-pasos correspondientes. (fig. 1.6)

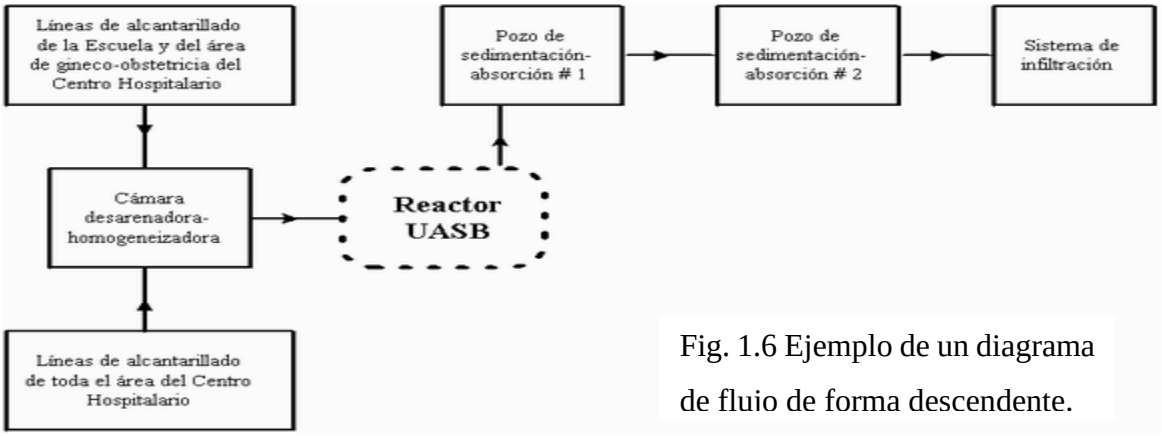
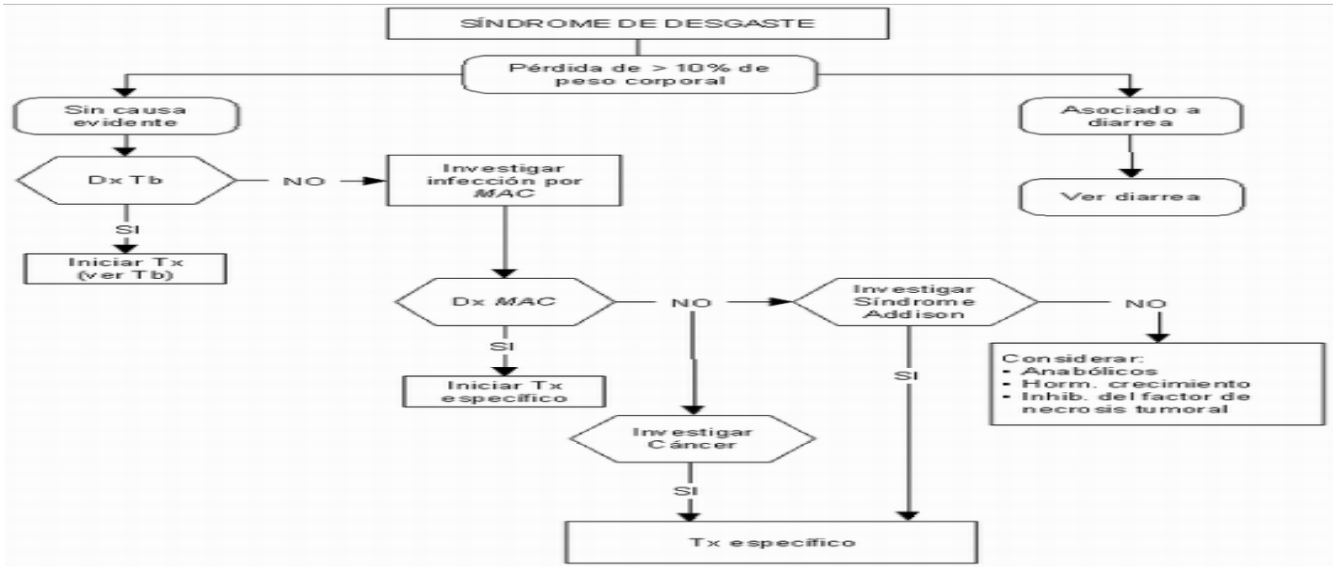


Fig. 1.6 Ejemplo de un diagrama de fluio de forma descendente.

- Flujograma de segundo nivel o detallado

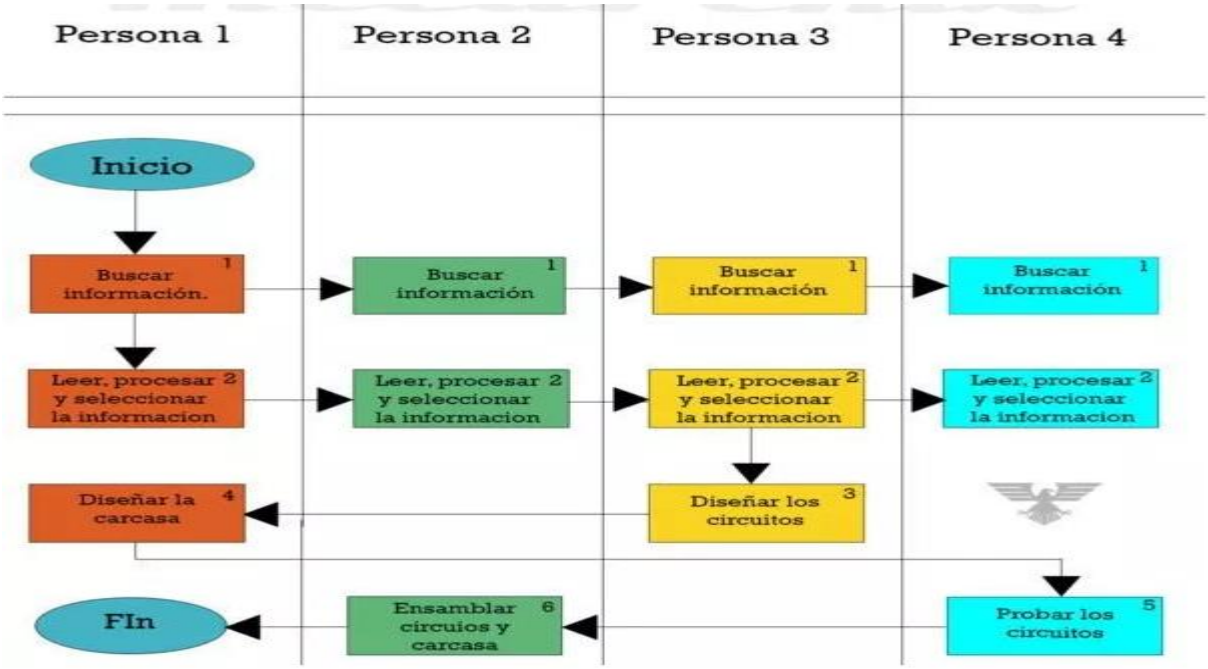
Un flujograma detallado indica los pasos o actividades de un proceso e incluye puntos de decisión, períodos de espera, tareas que se tienen que volver a hacer y ciclos de retroalimentación. (fig. 1.7)



Diagramas de flujo Fig. 1.7 Ejemplo de un diagrama de flujo de segundo nivel o detallado.

- Flujograma de ejecución o matriz:

Un flujograma de ejecución representa en forma gráfica el proceso en términos de quién se ocupa de realizar los pasos, Tiene forma de matriz e ilustra los diversos participantes y el flujo de pasos entre esos participantes. (fig. 1.7)



SÍMBOLOS DE DIAGRAMAS DE FLUJO EN COMPUTACIÓN:

Los diagramas de flujo parten de unos símbolos que permiten decidir lo mismo que en los algoritmos pero de una manera gráfica y, por supuesto, un poco más entendible. (fig. 1.8)

	Inicio/Final Se utiliza para indicar el inicio y el final de un diagrama; de Inicio sólo puede salir una línea de flujo y al final sólo debe llegar una línea		**Decisión** Indica la comparación de dos datos y dependiendo del resultado lógico (falso o verdadero) se toma la decisión de seguir un camino del diagrama u otro
	Entrada/Salida Entrada/Salida de datos por cualquier dispositivo (scanner, lector de código de barras, micrófono, parlantes, etc.)		**Impresora/Documento.** Indica la presentación de uno o varios resultados en forma impresa
	Entrada por teclado. Entrada de datos por teclado. Indica que el computador debe esperar a que el usuario teclee un dato que se guardará en una variable o constante		**Pantalla** Instrucción de presentación de mensajes o resultados en pantalla
	Acción/Proceso Indica una acción o instrucción general que debe realizarse (operaciones aritméticas, asignaciones, etc.)		**Conector Interno** Indica el enlace de dos partes de un diagrama dentro de la misma página
	Flujo/Flecas de Dirección Indica el seguimiento lógico del diagrama. También indica el sentido de ejecución de las operaciones		**Conector Externo** Indica el enlace de dos partes de un diagrama en páginas diferentes

Fig. 1.8 Simbología utilizada para elaborar un diagrama

REGLAS PARA ESTRUCTURAR UN DIAGRAMA DE FLUJO

1) Los diagramas de flujo se escriben de arriba hacia abajo, de izquierda a derecha y deben de tener un inicio y un fin
2) Todo símbolo (excepto las líneas de flujo) llevará en su interior información que indique su función exacta y unívoca. (Fig. 1.9)

Ejemplos:

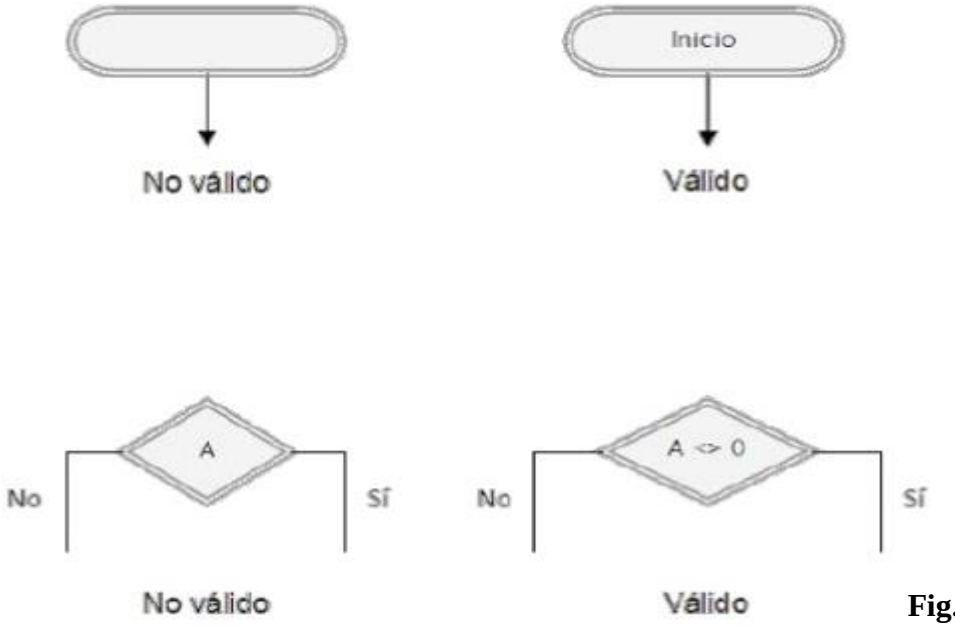


Fig. 1.9

3) Un elemento de un diagrama no puede tener más de una salida si no es un elemento de decisión.

Ejemplos:

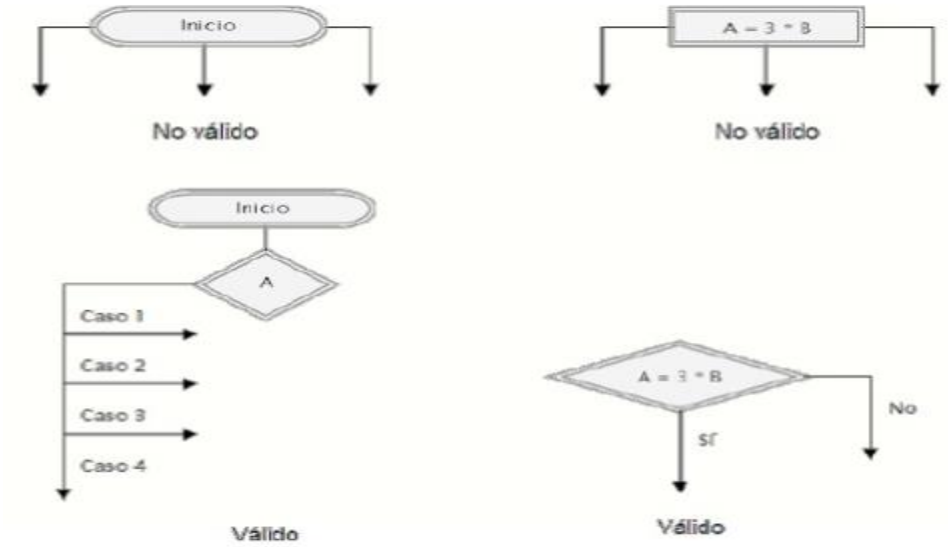
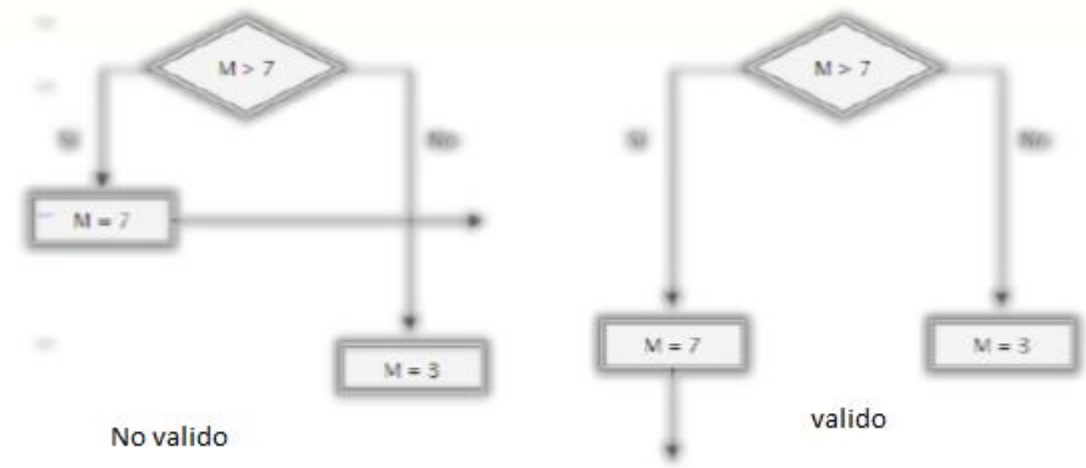


Fig. 1.10

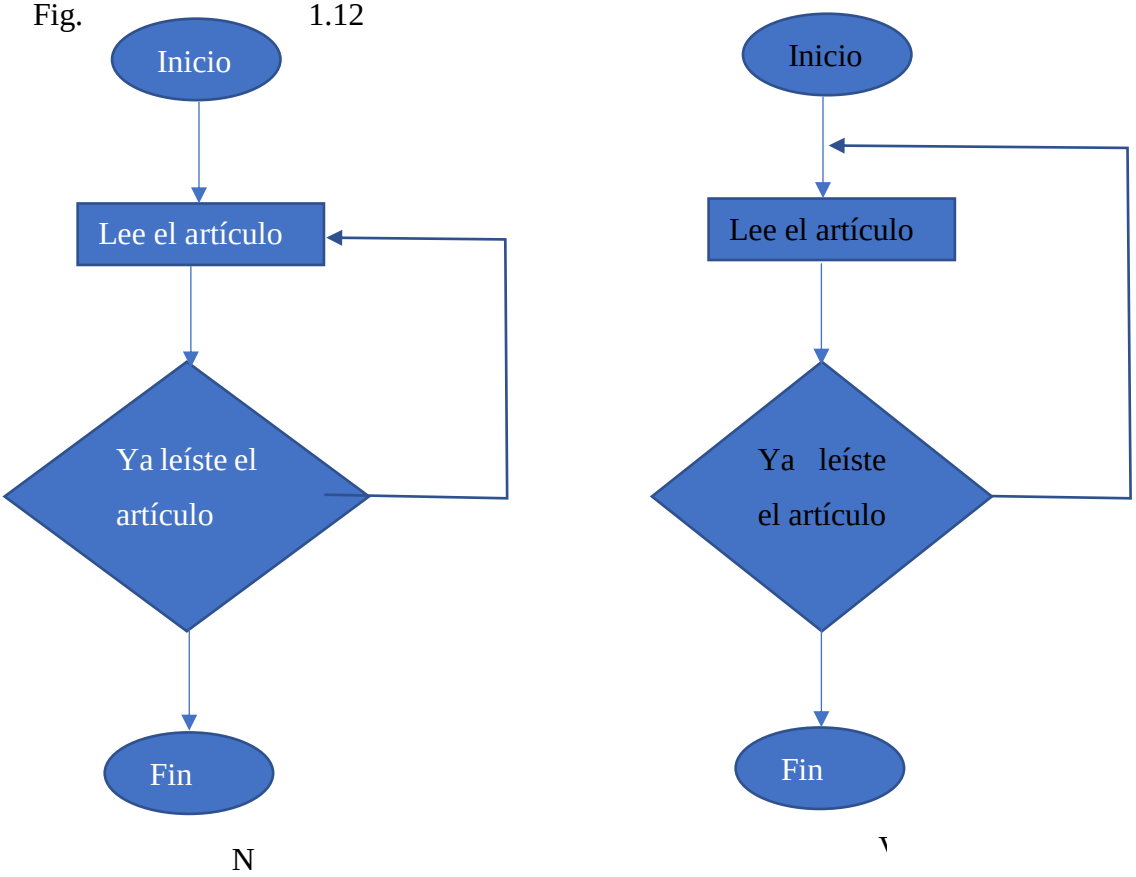
4) Las líneas de flujo no deben cruzarse. (Fig 1.11)

Fig. 1.11



5) No puede llegar más de una línea a un símbolo (Fig. 1.12)

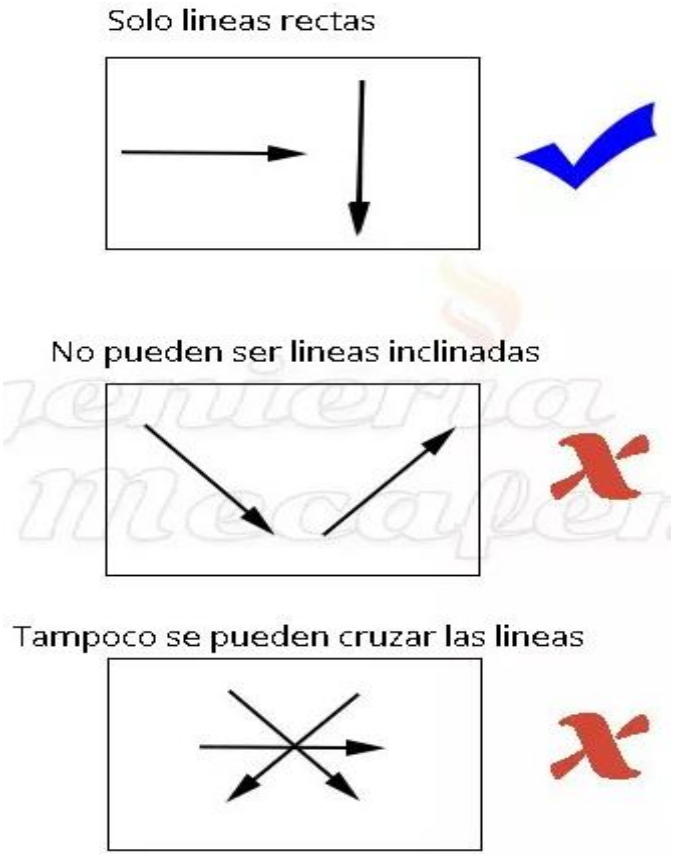
Fig. 1.12



N

6) Las líneas utilizadas para indicar la dirección del flujo del diagrama deben ser rectas verticales y horizontales. (fig. 1.13)

Fig. 1.13



7) Todas las líneas utilizadas para indicar la dirección del flujo del diagrama deben estar conectadas, la conexión debe ser a un símbolo que exprese lectura, proceso, decisión, conexión o fin del diagrama. (fig. 1.14)

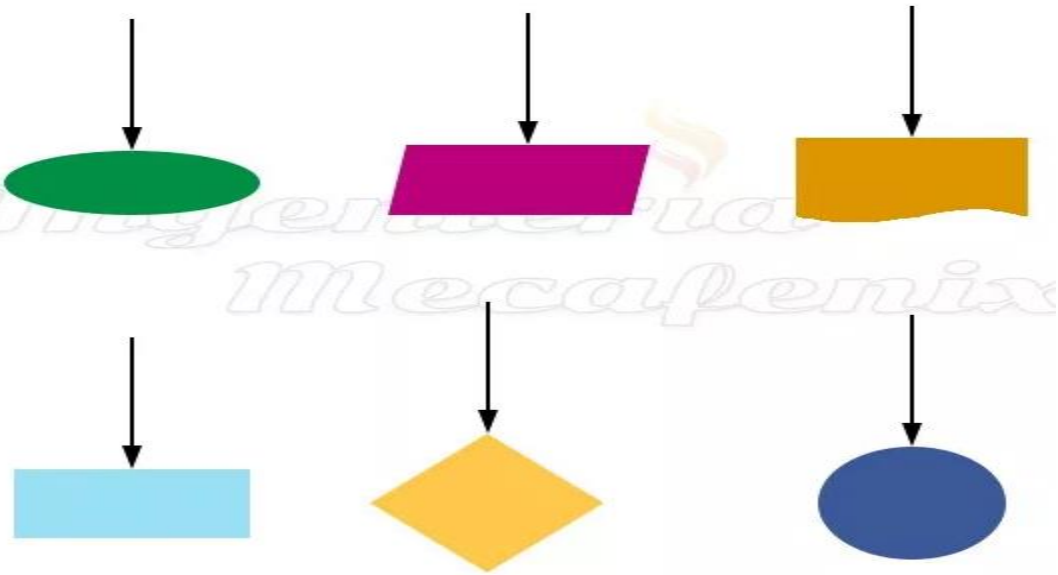


Fig. 1.14

1.3 PSEUDOCÓDIGOS

DEFINICIÓN

El **pseudocódigo** (o lenguaje de descripción algorítmico) es una descripción de alto nivel compacta e informal del principio operativo de un programa informático u otro algoritmo. Hay quienes también lo definen como: falso lenguaje que es comúnmente utilizado por los programadores para omitir secciones de código o para dar una explicación del paradigma que tomó el mismo programador para hacer sus códigos, esto quiere decir que el pseudocódigo no es programable sino facilita la programación.

OBJETIVO

El principal objetivo del pseudocódigo es el de representar la solución a un algoritmo de la forma más detallada posible, y a su vez lo más parecido posible al lenguaje que posteriormente se utilizará para la codificación del mismo.

El pseudocódigo utiliza para representar las acciones sucesivas palabras reservadas en inglés (similares a sus homónimos en los lenguajes de programación), tales como star, begin, end, stop, if-then-else, while, repeat-until….etc.

Es un lenguaje de especificaciones de algoritmos. El uso de tal lenguaje hace el paso de codificación final (esto es, traducción a un lenguaje de programación) relativamente fácil.

Las principales características de este lenguaje son:

- Se puede ejecutar en un ordenador
- Es una forma de representación sencilla de utilizar y de manipular.
- Facilita el paso del programa al lenguaje de programación.
- Es independiente del lenguaje de programación que se vaya a utilizar.
- Es un método que facilita la programación y solución al algoritmo del programa.

Todo documento en pseudocódigo debe permitir la descripción de:

- Instrucciones primitivas
- Instrucciones de proceso
- Instrucciones de control
- Instrucciones compuestas
- Instrucciones de descripción

Estructura a seguir en su realización:

Cabecera:

- Programa
- Modulo:
- Tipos de datos:
- Constantes:
- Variables:

Cuerpo:

- Inicio
- Instrucciones
- Fin

```
Procedimiento Ordenar (L)
  //Comentario: L = (L1, L2, ..., Ln) es una lista con n elementos//
  k ← 0;
  Repetir
    intercambio ← Falso;
    k ← k + 1;
    Para i ← 1 Hasta n − k ConPaso 1 Hacer
      Si Li > Li+1 Entonces
        intercambiar (Li, Li+1)
        intercambio ← Verdadero;
      FinSi
    FinPara
  Hasta Que intercambio = Falso;
FinProcedimiento
```

Para comentar en pseudocódigo se le antepone al comentario dos asteriscos (*)

Ejemplo:

Programa que calcula el área de un cuadrado a partir de un lado dado por el teclado.

```
Programa: area_cuadrado
Modulo: main **(también se puede llamar principal)
Variables:
lado: natural
area: natural
Inicio
Visualizar "Introduce el lado del cuadrado"
Leer lado
Area<- lado * lado
Visualizar "El área del cuadrado es", área
Fin
```