RESOLUCIÓN DE PROBLEMAS Y ALGORITMOS

La principal razón para que las personas aprendan lenguajes de programación es utilizar una computadora como una **herramienta para la resolución de problemas**. Cinco fases pueden ser identificadas en el proceso:

1. **Análisis del problema:** primer paso para encontrar la solución a un problema es el análisis del mismo.
2. **Estudio de su solución:** Se debe examinar cuidadosamente el problema a fin de obtener una idea clara sobre lo que se solicita y determinar los datos necesarios para conseguirlo.
3. **Diseño del Algoritmo:** Un algoritmo puede ser definido como la secuencia ordenada de pasos, sin ambigüedades, que conducen a la resolución de un problema dado y expresado en lenguaje natural, por ejemplo el castellano. Todo algoritmo **debe ser**:
   - **Preciso**: Indicando el orden de realización de cada uno de los pasos.
   - **Definido**: Si se sigue el algoritmo varias veces proporcionándole (consistente) los mismos datos, se deben obtener siempre los mismos resultados.
   - **Finito**: Al seguir el algoritmo, este debe terminar en algún momento, es decir tener un número finito de pasos.

> *En un algoritmo se deben de considerar tres partes:*
> - *Entrada: Información dada al algoritmo.*
> - *Proceso:Operaciones o cálculos necesarios para encontrar la solución del problema.*
> - *Salida: Respuestas dadas por el algoritmo o resultados finales de los procesos realizados.*
>
> *Es aconsejable utlizar antes de la codificación:*
> - *Diagrama de flujos*
> - *Pseudolenguaje*

4. **Codificación del programa:** en un lenguaje de programación.
5. **Depuración y prueba:** verificación, corrección de errores y ejecución.

Como ejemplo supongamos que desea desarrollar un algoritmo que calcule la superficie de un rectángulo proporcionándole su base y altura. Lo primero que debemos hacer es plantearnos las siguientes preguntas:

Especificaciones de entrada:
- ¿Que datos son de entrada?
- ¿Cuántos datos se introducirán?
- ¿Cuántos son datos de entrada válidos?

Especificaciones de salida:
- ¿Cuáles son los datos de salida?
- ¿Cuántos datos de salida se producirán?
- ¿Qué formato y precisión tendrán los resultados?

El algoritmo que podemos utilizar es el siguiente:
- Paso 1. Entrada desde el teclado, de los datos de base y altura.
- Paso 2. Cálculo de la superficie, multiplicando la base por la altura.
- Paso 3. Salida por pantalla de base, altura y superficie calculada.

EJEMPLOS INTUITIVOS DE ALGORITMOS

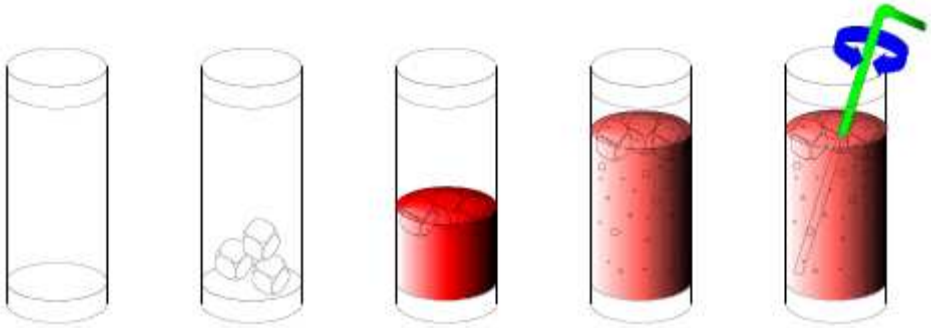
- PROBLEMA: calcular la media aritmética de tres números cualesquiera utilizando una calculadora básica.
  - Fases 1 y 2: Análisis y Estudio de la solución (ejemplo 16, 9, 4)

- Fase 3: Diseño del **Algoritmo**

1. Pulsar la tecla "ON"
2. Teclear el primer número
3. Pulsar la tecla "+"
4. Teclear el segundo número
5. Pulsar la tecla "+"
6. Teclear el tercer número
7. Pulsar la tecla "÷"
8. Pulsar la tecla "3"
9. Pulsar la tecla "="
10. La media de los tres números aparece en la pantalla
11. Pulsar la tecla "OFF"

OTRO EJEMPLO

- EJEMPLOS INTUITIVOS DE ALGORITMOS
  - PROBLEMA: Preparación de un "Tinto de Verano"
    - Fases 1 y 2: Análisis y Estudio de la solución



- Fase 3: Diseño del **Algoritmo**

1. Tomar un vaso.
2. Colocar algunos cubitos de hielo en el vaso.
3. Echar vino tinto en el vaso.
4. Añadir gaseosa al contenido del vaso.
5. Agitar el contenido.

- Fase 3: Diseño del **Algoritmo**

1. Tomar un vaso **vacío**.
2. Colocar **tres** cubitos de hielo en el vaso.
3. Echar vino tinto **hasta la mitad** del vaso.
4. Añadir gaseosa **hasta llenar** el vaso.
5. Agitar **tres segundos** el contenido.

ENTONCES

- Algoritmo
  - Dado un procesador y un entorno bien definido, es el enunciado de una **secuencia finita** de **acciones primitivas** que **resuelven** un determinado problema

DÓNDE..

Procesador es…

Toda entidad capaz de comprender un enunciado y ejecutar el trabajo indicado en el mismo.
Un procesador conveniente puede ser una persona que entienda las tareas descriptas y cuente con los elementos necesarios para ejecutarlas.
El procesador no puede realizar el trabajo demandado si no cuenta con los recursos necesarios.

Ambiente o entorno es …

El conjunto de todos los recursos necesarios para la ejecución de un trabajo.
El ambiente para un procesador dado es, por consiguiente específico del trabajo a ejecutar.
La elección hecha para efectuar el trabajo o no, depende de los recursos puestos a su disposición.

Acción es …

Un evento que modifica el ambiente.
Una acción transforma el ambiente desde un determinado estado, el estado inicial, a otro que puede ser un estado intermedio o el estado final.
El procesador debe respetar la secuencia de acciones.
Las acciones deben ser ejecutadas en el orden en que aparecen en el enunciado del procedimiento y deben escribirse en líneas sucesivas.
El enunciado de una misma acción puede aparecer más de una vez, en la descripción de un trabajo.

Primitiva es …

Para un procesador dado, una acción es primitiva si su enunciado es suficiente, para que pueda ejecutarla sin información adicional.
Una acción no primitiva debe ser descompuesta en acciones primitivas para un procesador dado.
Existen distintas técnicas para descomponer una acción en acciones primitivas.
Uno de los métodos es denominado técnicas de refinamientos sucesivos.

- Fase 3: Diseño del **Algoritmo**

1. Tomar un vaso vacío.
2. Colocar tres cubitos de hielo en el vaso.
   a) Sacar la cubitera del congelador.
   b) Rociar la parte inferior con agua.
   c) REPETIR
   d) Extraer un cubito.
   e) Echarlo al vaso.
   f) HASTA QUE el nº de cubitos sea 3.
   g) Rellenar los huecos de la cubitera con agua.
   h) Meter de nuevo la cubitera en el congelador.
3. Echar vino tinto hasta la mitad del vaso.
4. Añadir gaseosa hasta llenar el vaso.
5. Agitar tres segundos el contenido.

Ejemplo incorrecto de algoritmo…	*Ejemplo correcto de algoritmo…*
1. Hacer la división decimal exacta de 1 entre 3. 2. Imprimir	1. Hacer la división decimal exacta de 1 entre 3 hasta obtener un resultado con 2 cifras decimales. 2. Imprimir
Ejemplo incorrecto pués la primera acción no tiene fin. No es un algoritmo.	Ejemplo correcto pués la primera acción tiene fin. Constituye un algoritmo.

ELEMENTOS DE PROGRAMACIÓN

El lenguaje algorítmico debe ser independiente de cualquier lenguaje de programación particular, pero fácilmente traducible a cada uno de ellos. Alcanzar estos objetivos conducirá al empleo de métodos normalizados para la representación de algoritmos, tales como los diagrama de flujo o pseudocódigo.

PSEUDOCÓDIGO

La **formalización de las acciones** se realiza a través de la estructura del algoritmo en pseudocódigo. El pseudocódigo se considera una herramienta para el diseño que permite obtener una solución mediante aproximaciones sucesivas.

Se denomina notación de pseudocódigo a aquella que permite describir la solución de un problema en forma de algoritmo dirigido al computador utilizando palabras y frases del lenguaje natural sujetas a determinadas reglas.

Ejemplo 1:
{Suma el entero 5 y el entero 7 }

```
principal
comienza
int a ←5 //en la variable a se almacena el valor de 5
int b ←7 //en la variable b se almacena el valor de 7
int c ←a + b //en c se almacena la suma de a con b
termina
```

Ejemplo 2:
{suma valores ingresados por teclado}

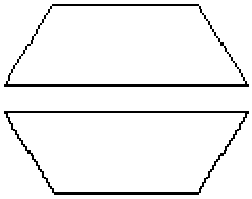
```
principal
comienza
int a, b, c
escribe “De un valor entero”
lee a
escribe “de otro valor entero”
lee b
c ← a + b
escribe “La suma de”
escribe a
escribe “con”
escribe b
escribe “es”
escribe c
termina
```

DIAGRAMAS DE FLUJO

También conocidos como flowchart, en estos se utilizan símbolos estándar, en el que cada paso para la elaboración del programa se representa con el símbolo y orden adecuados, unidos o conectados por flechas, también llamadas líneas de flujo, esto por que indican el sentido en el que se mueve el proceso. En resumen el diagrama de flujo es un medio de presentación visual y gráfica del flujo de datos a través de un algoritmo.

Bloques terminales

Bloques de inicio y fin de programa
Indican los límites del procedimiento considerado como principal. Generalmente se trata de un programa completo o de un módulo funcionalmente autónomo.

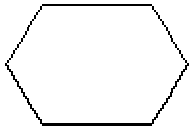


Bloques de inicio y fin de procedimiento
Indican los límites de un procedimiento considerado como una parte dependiente de otro mayor. Delimitan la explosión de un grupo de acciones que han sido consideradas como un procedimiento en otra parte del diagrama. Generalmente se trata de una función que hace una tarea específica.

Bloques de acción

Bloque de acción simple
Representa una acción sencilla que puede ser considerada como única y que generalmente se codifica con una sola instrucción. Por ejemplo: incrementar contador, ubicar cursor, abrir archivo, etc.

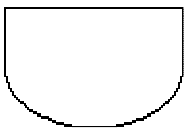
Bloque de entrada/salida
Representa una acción simple de entrada o salida de datos, generalmente desde o hacia un dispositivo periférico como el teclado, la pantalla o el disco. Por ejemplo: ingresar valor, leer registro, mostrar resultado, etc.



Bloque de procedimiento
Representa un conjunto de acciones que se consideran juntas, sin analizar su detalle. Este grupo de acciones se describe generalmente como procedimiento en otra parte del diagrama. Por ejemplo: buscar elemento, ordenar conjunto, procesar dato, etc.

Bloques de decisión

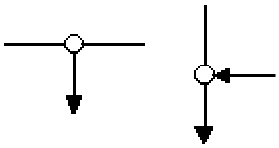
Bloque de decisión simple
Representa la acción de analizar el valor de verdad de una condición, que sólo puede ser verdadera o falsa (selección simple). Según el resultado de esta evaluación se sigue uno u otro curso de acción. Por lo tanto, de un bloque de decisión simple siempre salen exactamente dos flujos, uno por V (sí) y otro por F (no).



Bloque de decisión múltiple
Representa la acción de analizar el valor de una variable, que puede tomar uno entre una serie de valores conocidos (selección múltiple). Según el resultado de esta evaluación, se sigue uno entre varios cursos de acción. Por lo tanto, de un bloque de decisión múltiple siempre salen varios flujos, uno por cada valor esperado de la variable analizada.

Flujos y conectores

Flecha o flujo
Indica la secuencia en que se van ejecutando las acciones al pasar de un bloque a otro.



Conector
Indica la convergencia de dos o más flujos. En la práctica determina el comienzo o el fin de una estructura.

Veamos a continuación un conjunto de reglas o normas que nos permiten construir un diagrama de flujo.

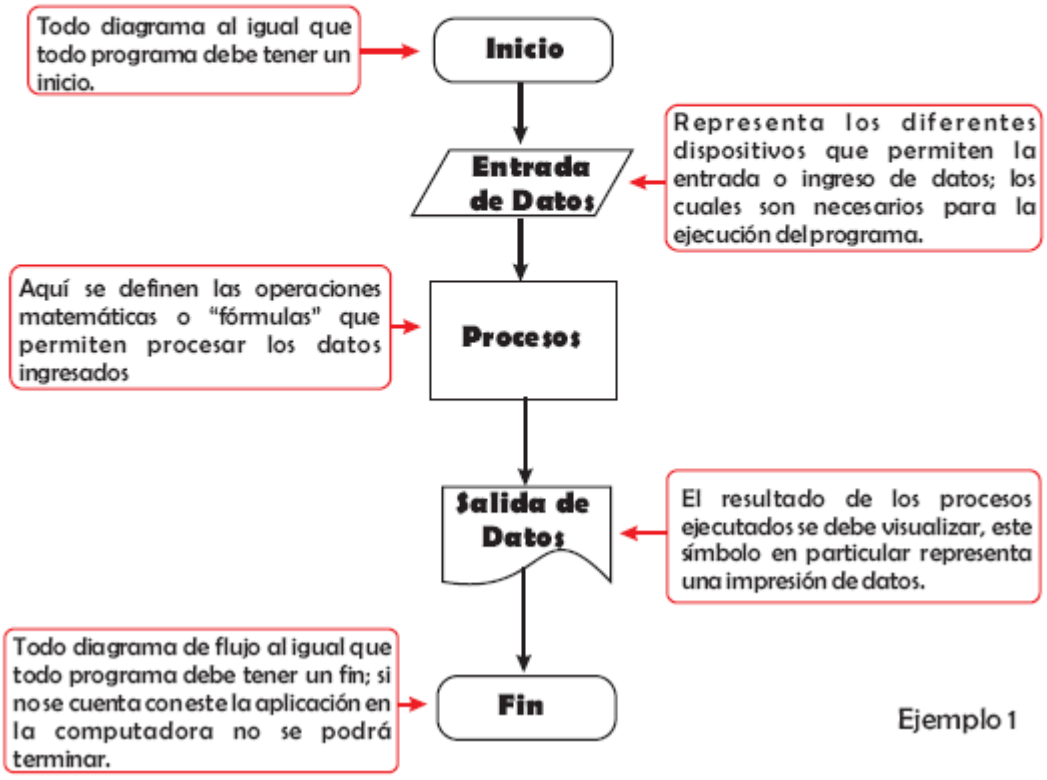
1. Todo diagrama de flujo debe tener un inicio y un fin.
2. Las líneas utilizadas para indicar la dirección del diagrama deben ser rectas, horizontales o verticales, nunca se deben cruzar entre si.

3. No deben haber líneas sin conexión a los demás elementos del diagrama de flujo.
4. Un diagrama de flujo se debe construir de arriba hacia abajo y de requerirse de izquierda a derecha.
5.La notación o símbolos utilizados en el diagrama de flujo son independientes del lenguajes de programación utilizado para la elaboración del programa o aplicación.
6. No puede llegar más de una línea de conexión a un símbolo.

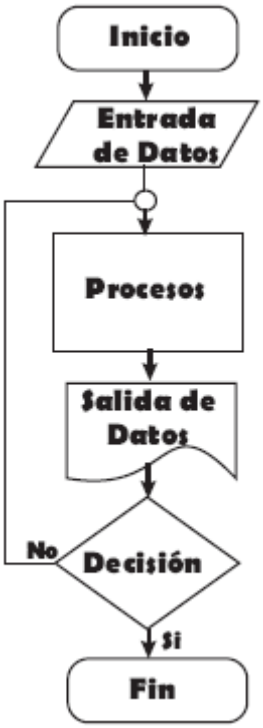
Para tener en cuenta: En programación utilizamos las llamadas expresiones lógicas, que están constituidas por números, constantes o variables y operadores lógicos o relacionales, que pueden tomar un valor de falso o verdadero dependiendo del resultado de la evaluación para seguir por un camino determinado. Los operadores relacionales son aquellos que permiten la comparación y los operadores lógicos son los que permiten formular condiciones.

ESTRUCTURAS BÁSICAS DE UN DIAGRAMA DE FLUJO

Esta es una de las estructuras más simples que se pueden dar para resolver un algoritmo, en donde se observa una entrada de datos, el procesamiento de estos y la salida de los datos.



Ejemplo 1



En esta estructura de Diagrama de Flujo encontramos los mismos componentes del ejemplo anterior, pero se utiliza un símbolo de "Decisión", el cual permite tomar uno y solo uno de los dos caminos disponibles, siempre y cuando se cumpla cierta condición, al dejar de cumplirse la condición inicial se continua por la otra alternativa.

A este tipo de estructuras cíclicas o repetitivas también se les llama bucles o loops.

Permiten la ejecución del mismo proceso todas las veces de forma idéntica mientras se de la condición u operación lógica planteada.

Ejemplo 2