

## Pasos de resolución de un algoritmo

Para la realización de un algoritmo es necesario realizar una secuencia de pasos para llegar a cumplir con la resolución de un problema dado. Para ello, definiremos los siguientes pasos:

1. **Análisis previo del Problema:** Se debe realizar un análisis del funcionamiento del problema antes que se realice cualquier algoritmo.
2. **Definición de Requerimientos:** Los problemas a solucionar, esto es, por ejemplo, el sumar dos números, multiplicar dos matrices, ordenar una lista de números, generar un reporte, etc.
3. **Identificación de los Módulos:** La identificación de los módulos es tan importante como la identificación correcta de los requerimientos, esto porque la correcta identificación de los módulos simplifica considerablemente la realización de los algoritmos que darán solución a los requerimientos identificados en el paso anterior.
4. **Realización de los Algoritmos:** El algoritmo deberá cumplir con las características que se indicaron para posteriormente implementarse en un lenguaje de programación comprensible por una computadora.
5. **Implementación de los Algoritmos:** La implementación de los algoritmos se debe realizar en un lenguaje de programación para que una computadora pueda comprender las instrucciones que el algoritmo modela para así poder ejecutarlas y lograr el resultado esperado.

En el último paso ya podemos hablar de una aplicación o programa de computadora, que estará compuesto por una serie de instrucciones que ordenadas una tras otra logran representar los algoritmos diseñados y dar así solución a los requerimientos identificados.

## Proceso para la solución de problemas utilizando algoritmos

Los seres humanos realizamos de manera cotidiana una serie de pasos, procedimientos o acciones que nos permiten realizar una actividad, alcanzar un resultado o resolver un problema.

Esta serie de pasos los realizamos diariamente desde que comienza el día, por ejemplo, cuando nos levantamos, tomamos un baño, desayunamos, preparamos la comida, entre muchas otras actividades, seguimos una serie de pasos que al final nos permiten alcanzar un resultado. La realidad es que a cada momento aplicamos un algoritmo para resolver un problema o alcanzar un resultado.

### ● Un algoritmo se define como

Un conjunto de pasos, procedimientos o acciones que nos permiten alcanzar un resultado o resolver un problema.

El concepto de algoritmo, aunque similar y obviamente relacionado, no debe confundirse con el concepto de programa. Mientras el primero es la especificación de un conjunto de pasos (operaciones, instrucciones, órdenes,...) orientados a la resolución de un problema (método), el segundo es un conjunto de operaciones especificadas en un determinado lenguaje de programación, susceptible de ser ejecutado (o compilado o interpretado).

Al momento de resolver un algoritmo se debe tener en cuenta la información existente, los procesos que se deben realizar y los resultados esperados. Si lo representamos en un diagrama de bloques tenemos que:

#### ● Datos de entrada

Son todos aquellos datos iniciales que serán manipulados con el fin de resolver el problema.

#### ● Procesos Algorítmicos

Luego de tener los datos de entrada, se requiere tener las instrucciones precisas que nos permitan manipularlos y transformarlos para obtener la salida deseada.

#### ● Datos de Salida

Son los datos que permiten visualizar la solución al problema que se deseaba resolver.

### Definición del problema

#### Análisis del problema

#### Diseño del programa

#### Características

#### Herramientas

#### Diagramas de flujo

## Definición del problema

Como ya mencionamos un problema se define cuando surge una necesidad. Por lo que se debe identificar claramente el problema y comprender cuál será la utilidad de la solución que se alcance. Se debe tener una visión general del problema estableciendo las condiciones iniciales y además, los límites del problema, es decir donde inicia y donde termina.

- Si necesitamos saber el sueldo total de un empleado. La solución que se alcance debe permitir conocer con exactitud la cantidad que debe pagarse.

- Definición del problema
- Análisis del problema
- Diseño del programa
- Características
- Herramientas
- Diagramas de flujo

## Análisis del problema

Es necesario entender con detalle el problema en cuestión, para obtener una radiografía del mismo en términos de los datos disponibles, y definir el proceso necesario para convertir los datos en la información requerida.

### La primera etapa consiste en definir los resultados esperados

- En este problema se desea calcular el sueldo total a pagar a un empleado

### La segunda etapa consiste en identificar los datos que se requieren para producir esa salida

- Para calcular el sueldo total a pagar del empleado se requiere conocer el nombre del empleado, el sueldo diario y los días trabajados.

### La tercera etapa consiste en determinar el proceso para convertir los datos de entrada en la salida esperada

- ¿Cómo se calcula el nombre? El nombre se obtendrá directamente del usuario y no sufrirá proceso alguno.
- ¿Cómo se calcula el sueldo total? El sueldo diario del trabajador se multiplica por los días trabajados.

- Definición del problema
- Análisis del problema
- Diseño del programa
- Características
- Herramientas
- Diagramas de flujo

## Diseño del programa

En el análisis se determina qué hace el programa, en la etapa de diseño se determina cómo se hace el programa. Esta fase se procede a diseñar la lógica para la solución del problema, aquí se plantean los pasos a incluir y el orden de los mismos, a través del siguiente procedimiento:

Entre los métodos más eficaces para el proceso de diseño está el diseño modular el cual se basa en dividir el problema en subproblemas y a continuación dividir estos subproblemas en otros niveles más bajos hasta que pueda ser implementada la solución. Cada subprograma es resuelto mediante un módulo que tiene un solo punto de entrada y un solo punto de salida. De tal manera que un programa diseñado correctamente consta de un programa principal (el módulo de nivel superior) que llama a uno o varios subprogramas que a su vez pueden llamar a otros subprogramas.

Los programas estructurados de esta manera tienen un diseño modular y el método de descomponer un programa en módulos más pequeños se llama programación modular. Los módulos pueden ser planeados, codificados, comprobados y depurados de manera independiente y posteriormente integrarlos. El proceso de romper el problema en cada etapa y expresar cada paso en forma más detallada se denomina refinamiento sucesivo.

El proceso que convierte los resultados del análisis de un problema en un diseño modular con refinamiento sucesivo que permita una posterior codificación en un lenguaje de programación se denomina diseño de algoritmo.

Cabe mencionar que el diseño de algoritmo es independiente del lenguaje de programación en el que se vaya a codificar.

## Las características de un algoritmo

Definición del problema	De las definiciones proporcionadas, podemos ver que un algoritmo para que ser catalogado como tal, debe exhibir ciertas prioridades:
Análisis del problema	<b>Ser definido.</b> - Sin ambigüedad, cada paso del algoritmo debe indicar la acción a realizar sin criterios de interpretación.
Diseño del programa	<b>Ser finito.</b> - Un número específico y numerable de pasos debe componer al algoritmo, el cual deberá finalizar al completarlos.
Características	<b>Tener una o más salidas.</b> - Debe siempre devolver un resultado; de nada sirve un algoritmo que hace algo y nunca sabemos que fue. El devolver un resultado no debe ser considerado como únicamente "verlos" en forma impresa o en pantalla, como ocurre con las computadoras. Existen muchos otros mecanismos susceptibles de programación que no cuentan con una salida de resultados de esta forma. Por salida de resultados debe entenderse todo medio o canal por el cual es posible apreciar los efectos de las acciones del algoritmo.
Herramientas	
Diagramas de flujo	<b>Efectividad.</b> - El tiempo y esfuerzo por cada paso realizado debe ser preciso, no usando nada más ni nada menos que aquello que se requiera para y en su ejecución.

Definido	Preciso	Finito
Si se sigue dos veces se obtiene el mismo resultado cada vez.	Se indica el orden de realización de cada paso.	Tiene un número determinado de pasos, siempre llega a un fin.

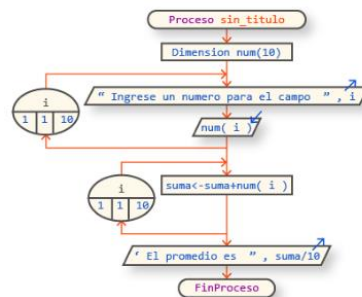
Por otra parte, un algoritmo consta de tres secciones o módulos principales.

[Ver Módulos >>](#)

Definición del problema
Análisis del problema
Diseño del programa
Características
Herramientas
Diagramas de flujo

### Diagrama de flujo

Representación gráfica de un algoritmo. Muestra gráficamente los pasos que se deben seguir para lograr la solución



### Pseudocódigo

También conocido como lenguaje de especificaciones de algoritmos. Herramienta de programación en la que las instrucciones se escriben en palabras similares al inglés o español.

```

Proceso Promedio
  Dimension num [10];
  Para i<-1 Hasta 10 Con Paso 1 Hacer
    Escribir "Ingrese un numero para el campo " , i
    leer num [i];
  Fin Para
  Para i<-1 Hasta 10 Con Paso 1 Hacer
    suma=suma+num[i]
  Fin Para
  escribir "El promedio es " , suma/10;
FinProceso
  
```

Definición del problema

Análisis del problema

Diseño del programa

Características

Herramientas

Diagramas de flujo

Para la construcción de diagramas de flujo se deben seguir una serie de reglas a continuación serán mencionadas:

1

2

3

4

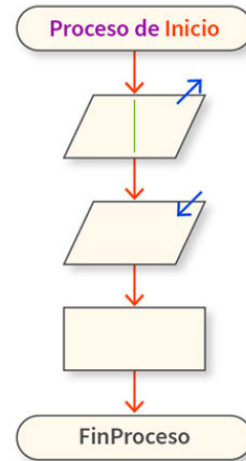
5

6

7

8

Todo diagrama de flujo tiene un inicio y un fin.



Las líneas utilizadas para indicar el flujo del diagrama deben ser rectas, verticales y horizontales. No deben cruzarse.

1

2

3

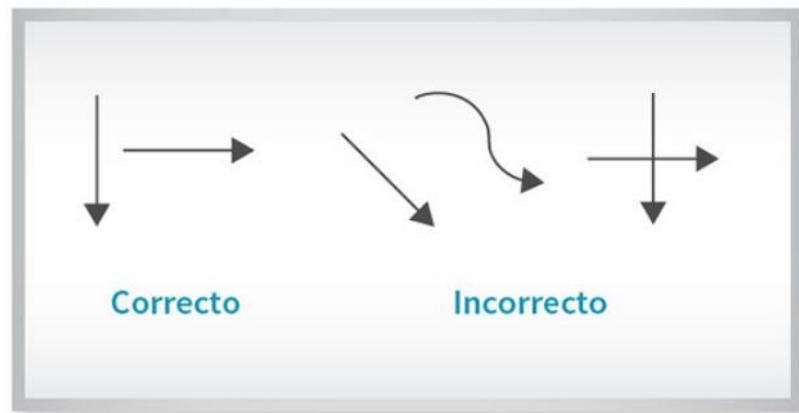
4

5

6

7

8



Todas las líneas utilizadas para indicar el flujo del diagrama deberán estar conectadas a un símbolo.

1

2

3

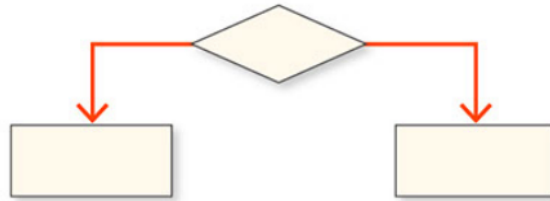
4

5

6

7

8



El diagrama tiene una construcción de arriba hacia abajo y de izquierda a derecha.

1

2

3

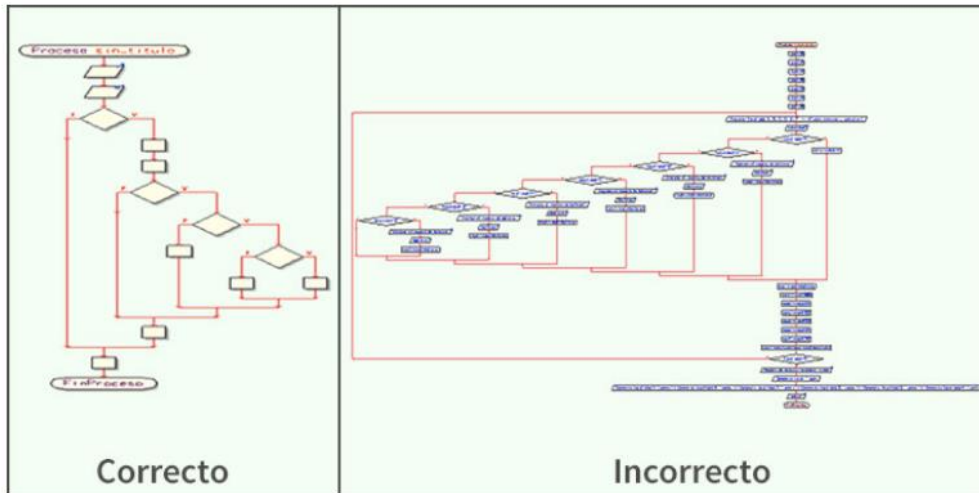
4

5

6

7

8



Correcto

Incorrecto

1

2

3

4

5

6

7

8

La notación utilizada en el diagrama de flujo debe ser independiente a cualquier lenguaje de programación.

1

2

3

4

5

6

7

8

Los comentarios pueden servir para clarificar una tarea compleja.

Si el diagrama de flujo requiere más de una hoja para su construcción se deben utilizar los conectores adecuados y enumerar las páginas.

1

2

3

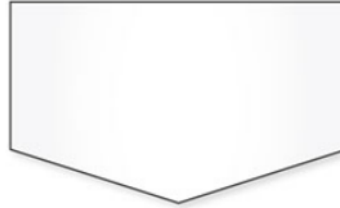
4

5

6

7

8



No puede llegar más de una línea a un símbolo. Deben llegar a la línea inmediata anterior.

1

2

3

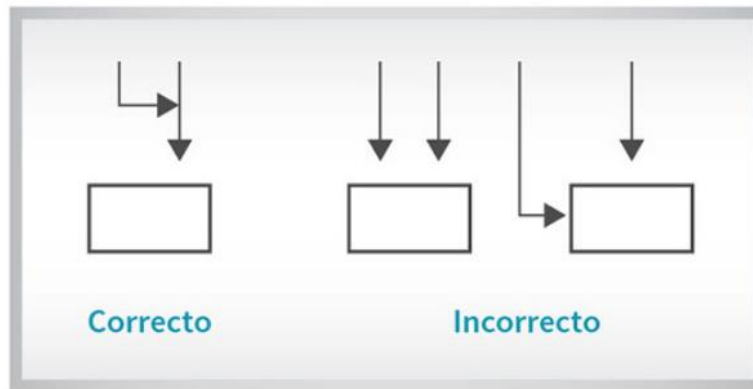
4

5

6

7

8





Definición del problema

Análisis del problema

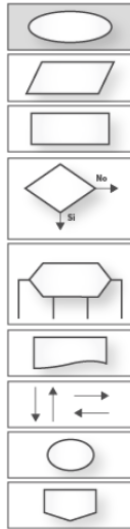
Diseño del programa

Características

Herramientas

Diagramas de flujo

## Símbolo



Y se utilizan los siguientes símbolos. Se debe aclarar que cada software maneja sus propios símbolos así que varían de uno a otro.

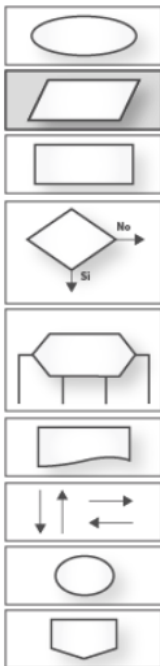
## Descripción

Símbolo utilizado para indicar el inicio y el fin del diagrama de flujo.

## Símbolo en PSEInt



## Símbolo



Y se utilizan los siguientes símbolos. Se debe aclarar que cada software maneja sus propios símbolos así que varían de uno a otro.

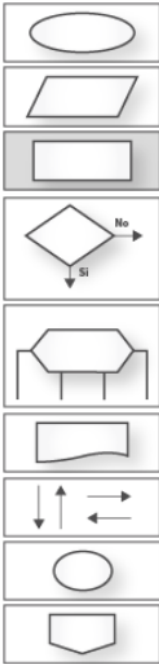
## Descripción

Símbolo utilizado para introducir datos al diagrama de flujo. Expresa entrada de datos o lectura.

## Símbolo en PSEInt



### Símbolo



Y se utilizan los siguientes símbolos. Se debe aclarar que cada software maneja sus propios símbolos así que varían de uno a otro.

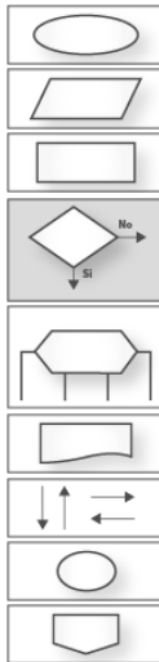
### Descripción

Símbolo para representar un proceso. En su interior debe ir una asignación, operación, cambio de valor, etc.

### Símbolo en PSEInt



### Símbolo

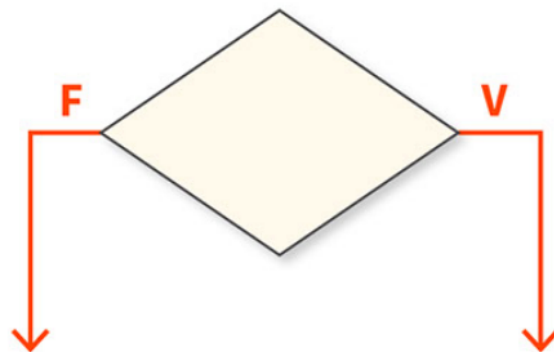


Y se utilizan los siguientes símbolos. Se debe aclarar que cada software maneja sus propios símbolos así que varían de uno a otro.

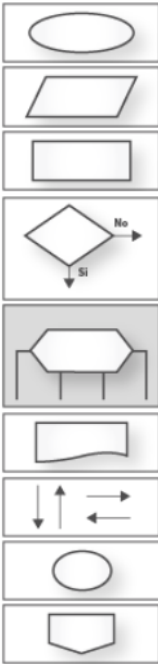
### Descripción

Símbolo utilizado para representar una decisión. En su interior debe ir una condición la cual debe ser evaluada, dependiendo del resultado se toma el camino verdadero o falso. Este símbolo se utiliza en las estructuras selectivas si y si entonces/sino.

### Símbolo en PSEInt



### Símbolo

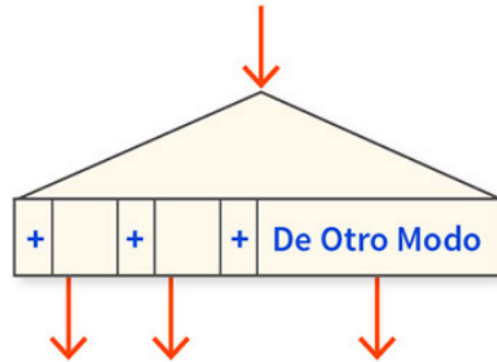


Y se utilizan los siguientes símbolos. Se debe aclarar que cada software maneja sus propios símbolos así que varían de uno a otro.

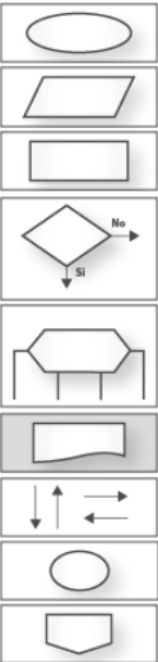
### Descripción

Símbolo utilizado para representar una decisión múltiple. En su interior va un selector, dependiendo del valor que este tenga se seguirá por alguno de los caminos. Este símbolo se utiliza en la estructura selectiva si múltiple.

### Símbolo en PSEInt



### Símbolo



Y se utilizan los siguientes símbolos. Se debe aclarar que cada software maneja sus propios símbolos así que varían de uno a otro.

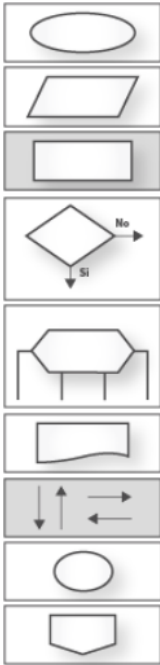
### Descripción

Símbolo que expresa escritura o impresión y se utiliza para imprimir un resultado.

### Símbolo en PSEInt



### Símbolo

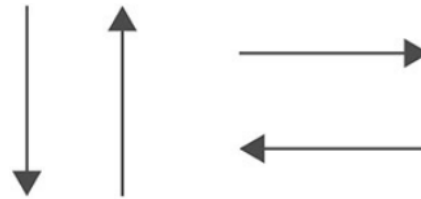


Y se utilizan los siguientes símbolos. Se debe aclarar que cada software maneja sus propios símbolos así que varían de uno a otro.

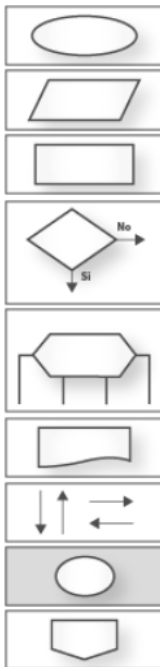
### Descripción

Se utilizan para expresar el flujo del diagrama. Es decir la dirección que siguen los datos.

### Símbolo en PSEInt



### Símbolo



Y se utilizan los siguientes símbolos. Se debe aclarar que cada software maneja sus propios símbolos así que varían de uno a otro.

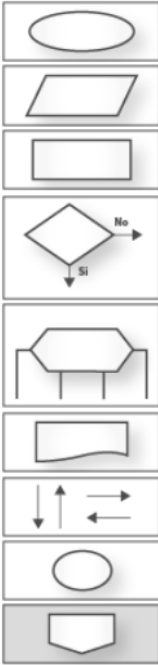
### Descripción

Símbolo utilizado para expresar conexión dentro de una misma página.

### Símbolo en PSEInt

**No se utiliza, el software acomoda automáticamente el diagrama de flujo para hacer las conexiones adecuadamente.**

### Símbolo



Y se utilizan los siguientes símbolos. Se debe aclarar que cada software maneja sus propios símbolos así que varían de uno a otro.

### Descripción

Símbolo utilizado para expresar conexión entre páginas diferentes.

### Símbolo en PSEInt

**No se utiliza, el software desarrolla el diagrama de flujo dentro de una misma página.**