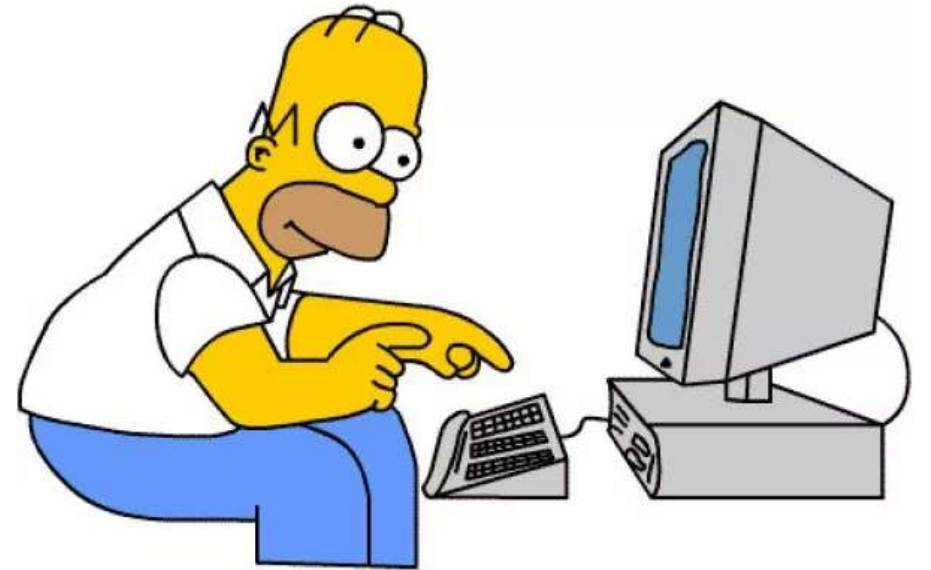


INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.



INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

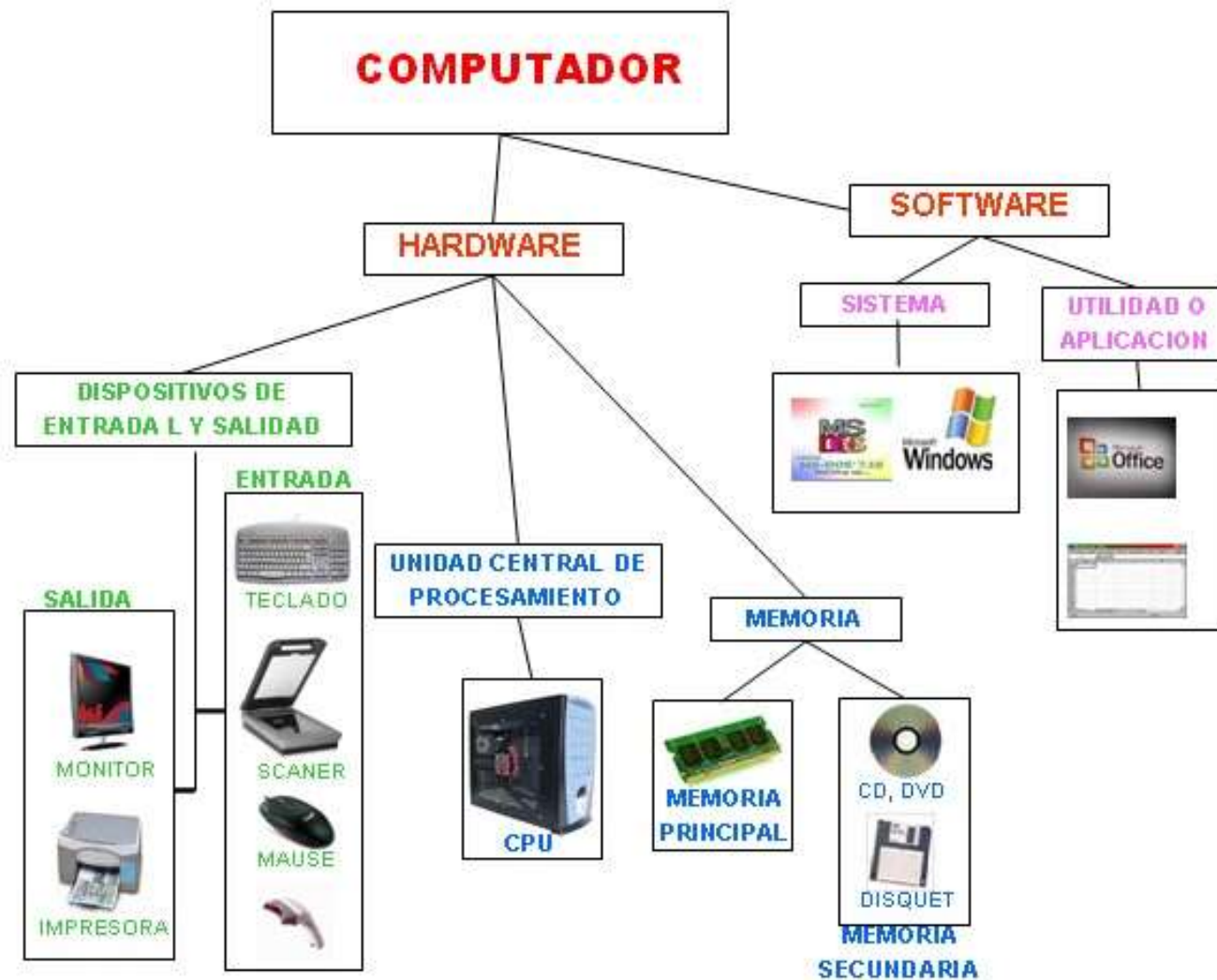
¿Qué es un ordenador?

Máquina electrónica capaz de realizar un tratamiento automático de la información y de resolver con gran rapidez problemas matemáticos y lógicos mediante programas informáticos.



INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.



INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

¿Qué significa programar?

Programar es darle órdenes a una computadora para que lleve a cabo una tarea determinada.

Computadora: dispositivo electrónico programable que puede almacenar, recuperar y procesar datos.

Programar: planificar una secuencia de instrucciones que ha de seguir una computadora. Debemos especificar exactamente lo que queremos hacer y el orden en el que debe hacerse (***algoritmo***).

Programa: la secuencia de instrucciones escritas en algún L.P.

Lenguaje de programación: conjunto de reglas, símbolos y palabras especiales utilizadas para construir programas.

INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

Un **programa** no es más que **una secuencia de instrucciones**, escritas en algún **lenguaje de programación** y pensadas para resolver algún tipo de **problema**. Eso sí, si no sabemos resolver este problema, no podremos escribir el programa.

Por eso cuando te plantees crear un programa primero tendrás que saber para qué servirá y cómo lo resolverá. A ti se te puede ocurrir una manera de resolverlo, a tu compañero, otra. Y las dos formas pueden ser correctas. Este **método con el que resolvéis el problema**, es lo que se llama **algoritmo**.

INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

¿Qué es un algoritmo?

Un algoritmo es una sucesión de pasos no ambiguos y que se pueden llevar a cabo en un tiempo finito a través del cual podemos hallar la solución a un problema.

- ✓ La descripción de cada paso no me lleva a ambigüedades, los pasos son absolutamente explícitos y no inducen a error.
- ✓ El número de pasos es finito. Tienen un principio y un fin.

INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

No es necesario ir a un libro de programación para entender lo que es un algoritmo.

Los algoritmos nos acompañan a todo momento en nuestras vidas, desde el momento en que nos levantamos a la mañana, tomamos una ducha, vamos al trabajo, a la universidad, y un sinnúmero de actividades cotidianas que hacemos con la finalidad de resolver algún problema específico, por más natural que sea.

INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

Lavarnos los dientes es un procedimiento que realizamos varias veces al día. Veamos la forma de expresar este procedimiento como un Algoritmo:

1. Tomar la crema dental.
2. Destapar la crema dental.
3. Tomar el cepillo de dientes.
4. Aplicar crema dental al cepillo.
5. Tapar la crema dental.
6. Abrir la llave del lavamanos.
7. Remojar el cepillo con la crema dental.
8. Cerrar la llave del lavamanos.
9. Frotar los dientes con el cepillo.
10. Abrir la llave del lavamanos.
11. Enjuagarse la boca.
12. Enjuagar el cepillo.
13. Cerrar la llave del lavamanos.
14. Secarse la cara y las manos con una toalla.

INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

Expresar en forma de Algoritmo la tarea de cambiar una bombilla.



INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

Expresar en forma de Algoritmo la tarea de cambiar una bombilla.

- 1 Colocar una escalera debajo de la bombilla a cambiar.
- 2 Tomar una bombilla nueva.
- 3 Subir por la escalera.
- 4 Sujetar y girar la bombilla a cambiar hacia la izquierda hasta retirarla del portalámparas.
- 5 Enroscar la bombilla nueva hacia la derecha en el portalámparas hasta apretarla.
- 6 Bajar de la escalera.

INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

En primer lugar tenemos un **Problema**, una situación que debe ser resuelta de alguna manera usando un programa informático.

Luego, realizamos un **Análisis de este problema**, una investigación, para saber cómo y con qué resolverlo.

Una vez tenemos determinado cómo y con qué resolver nuestro Problema, debemos elegir y **construir un Algoritmo**: una serie de pasos finitos y determinados para tener como resultado la resolución exacta del Problema.

INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

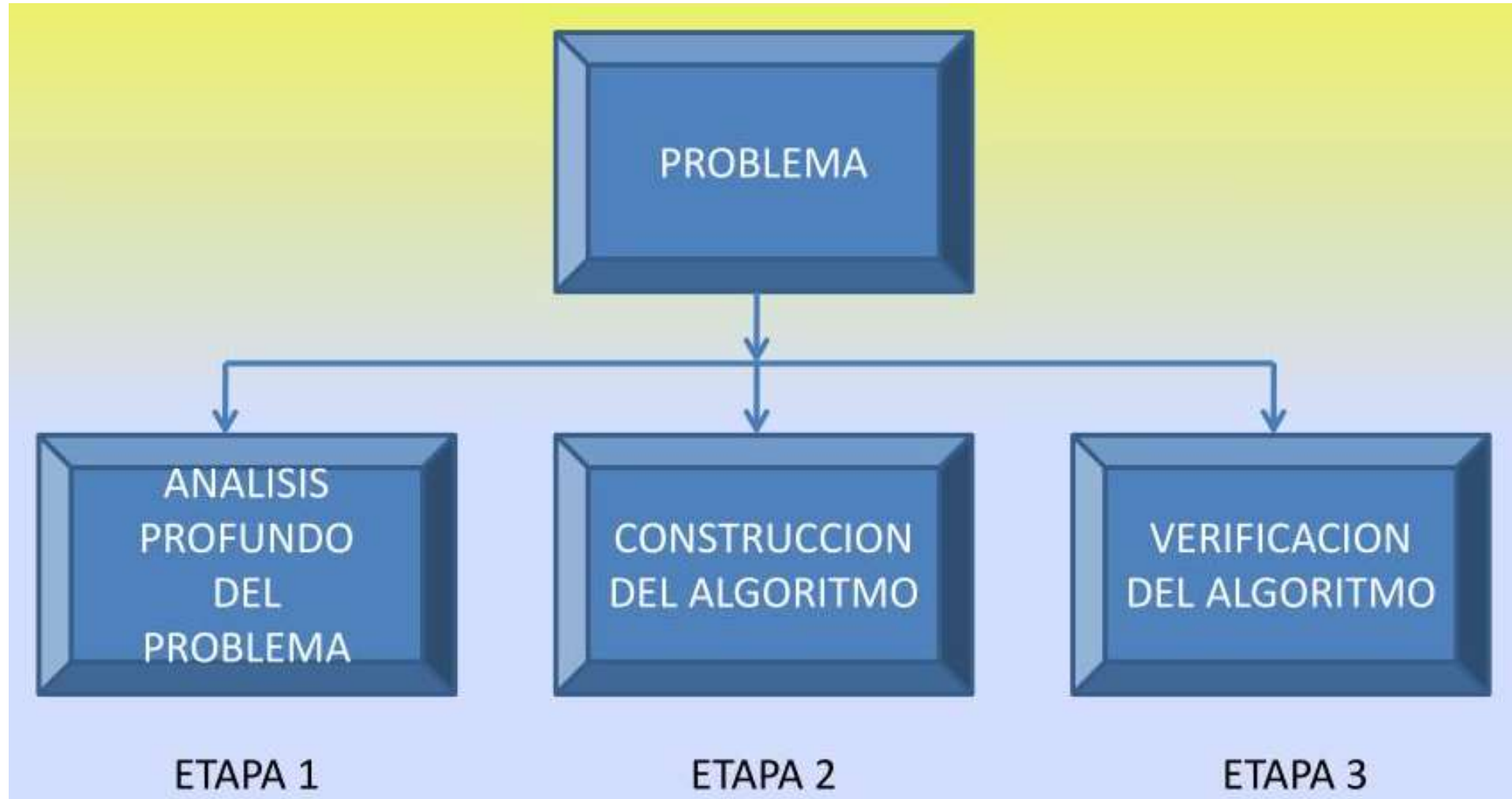
Como último paso, y antes de aplicar nuestro algoritmo, debemos **verificarlo**, es decir, probarlo, para estar seguros de que resuelve nuestro Problema.

Si al probarlo vemos que no resuelve nuestro Problema, debemos volver al Análisis para cambiarlo.

No es necesario aplicar ningún lenguaje de programación para crear un algoritmo. Si usamos un lenguaje, entonces ya deja de ser un algoritmo, sino que estamos programando nuestro algoritmo creado previamente.

INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

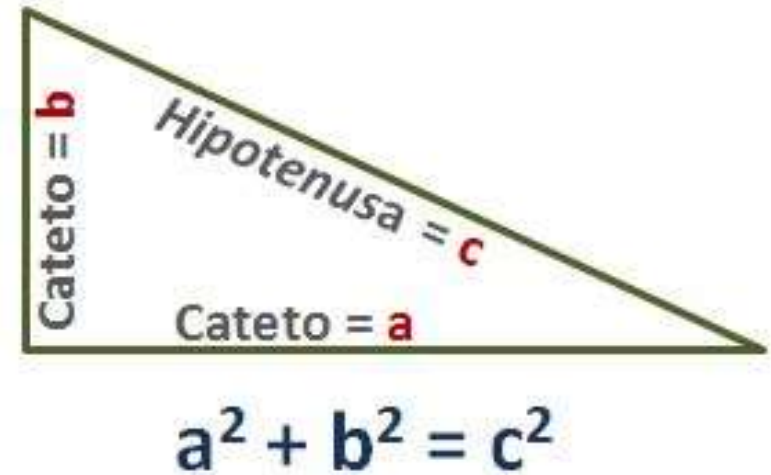


INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

Problema

Calcular la hipotenusa de un triángulo rectángulo, sabiendo el valor de sus catetos.



INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

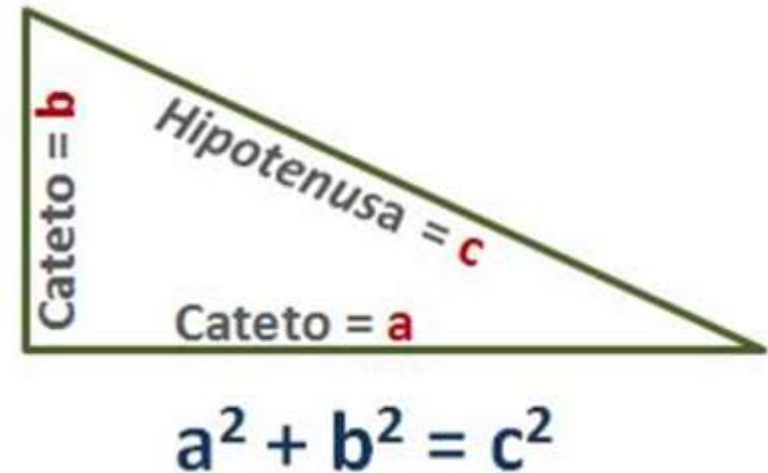
Análisis del problema

Tenemos que leer la longitud de los dos catetos y calcular la hipotenusa aplicando el Teorema de Pitágoras.

Variables de entrada: cateto a, cateto b.

Procesamiento: Teorema de Pitagoras.

Variables de salida: hipotenusa.



INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

Diseño del algoritmo

1º Pedir cateto a

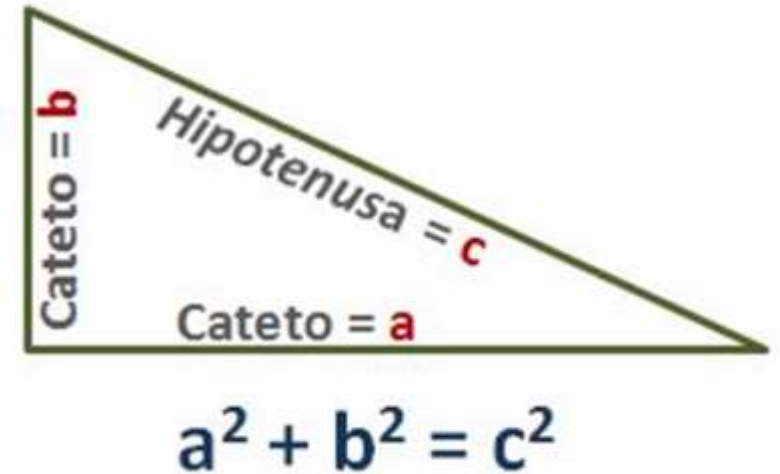
2º Leer cateto a

3º Pedir cateto b

4º Leer cateto b

5º Calcular hipotenusa En un triángulo rectángulo la hipotenusa es igual a la raíz cuadrada de la suma de los cuadrados de los catetos.

6º Mostrar la hipotenusa



INTRODUCCIÓN A LA PROGRAMACIÓN

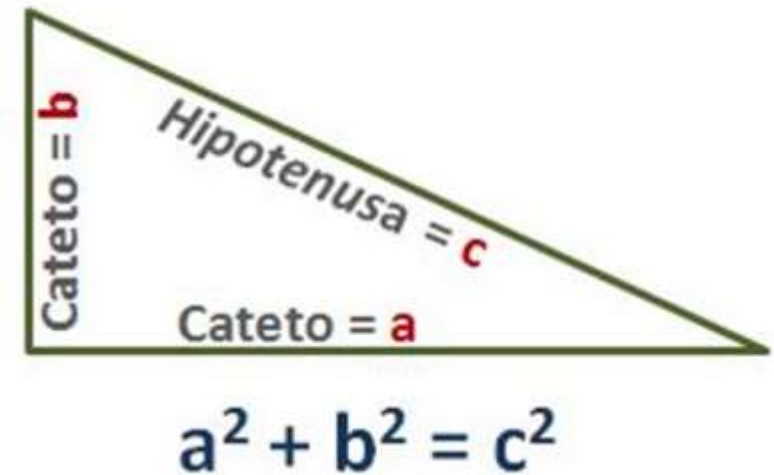
Algoritmos.

Verificar el algoritmo

cateto a = 5

cateto b = 3

hipotenusa = 5,83



¿La hipotenusa es correcta? Si la respuesta es sí, entonces el algoritmo es correcto. Caso contrario, el algoritmo es inválido.

INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

Todo algoritmo consta de tres partes:

- ❖ **Entrada:** se introducen todos aquellos datos que el algoritmo necesite para operar.
- ❖ **Procesamiento:** con lo recibido en la entrada, el algoritmo ejecutara una serie de instrucciones para resolver el problema.
- ❖ **Salida:** se mostrarán los resultados obtenidos en el procesamiento.

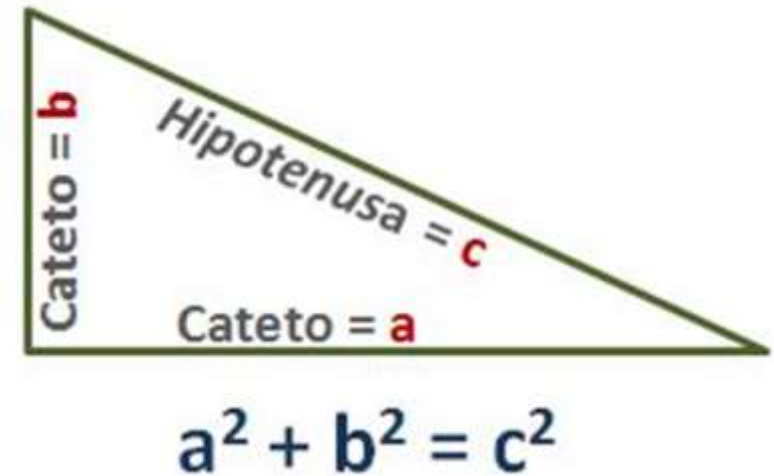
INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

DATOS DE ENTRADA: a y b

PROCESAMIENTO: hipotenusa = $\sqrt{a^2 + b^2}$

RESULTADO: imprimir hipotenusa



INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

$\text{datos} + \text{algoritmos} = \text{programas}$

Programar es escribir algoritmos en un lenguaje que entienda la máquina

INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

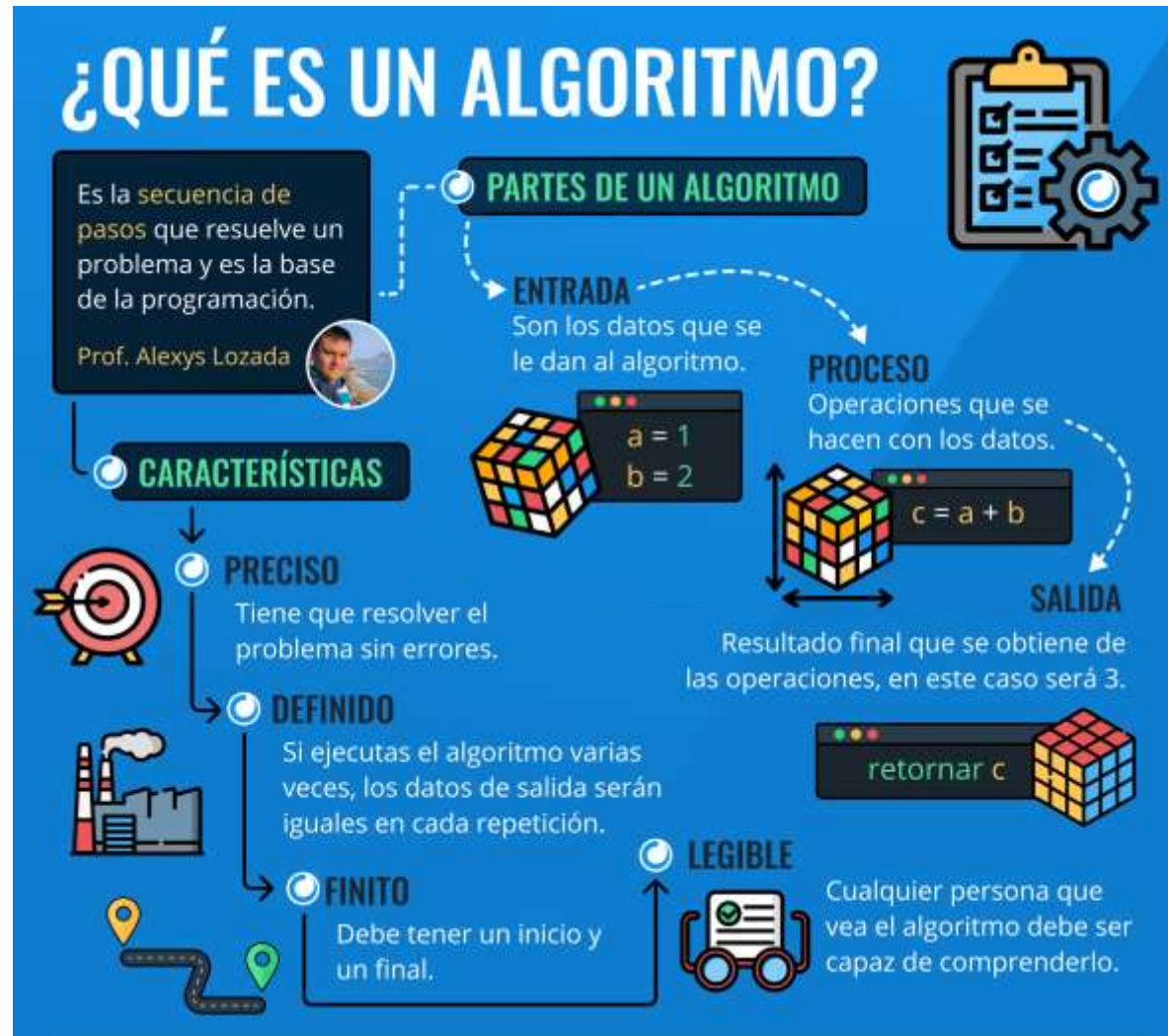
Características de un algoritmo:

- **Preciso:** Debe indicar el orden en el que se tiene que realizar cada paso.
- **Definido:** Ante los mismos datos de entrada siempre deben obtenerse los mismos resultados.
- **Finito:** Debe tener un fin, es decir, número determinado de pasos.

Para llegar a la realización de un programa es necesario el diseño previo de un algoritmo, de modo que sin algoritmo no puede existir un programa.

INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.



INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

Formas de representar un algoritmo:

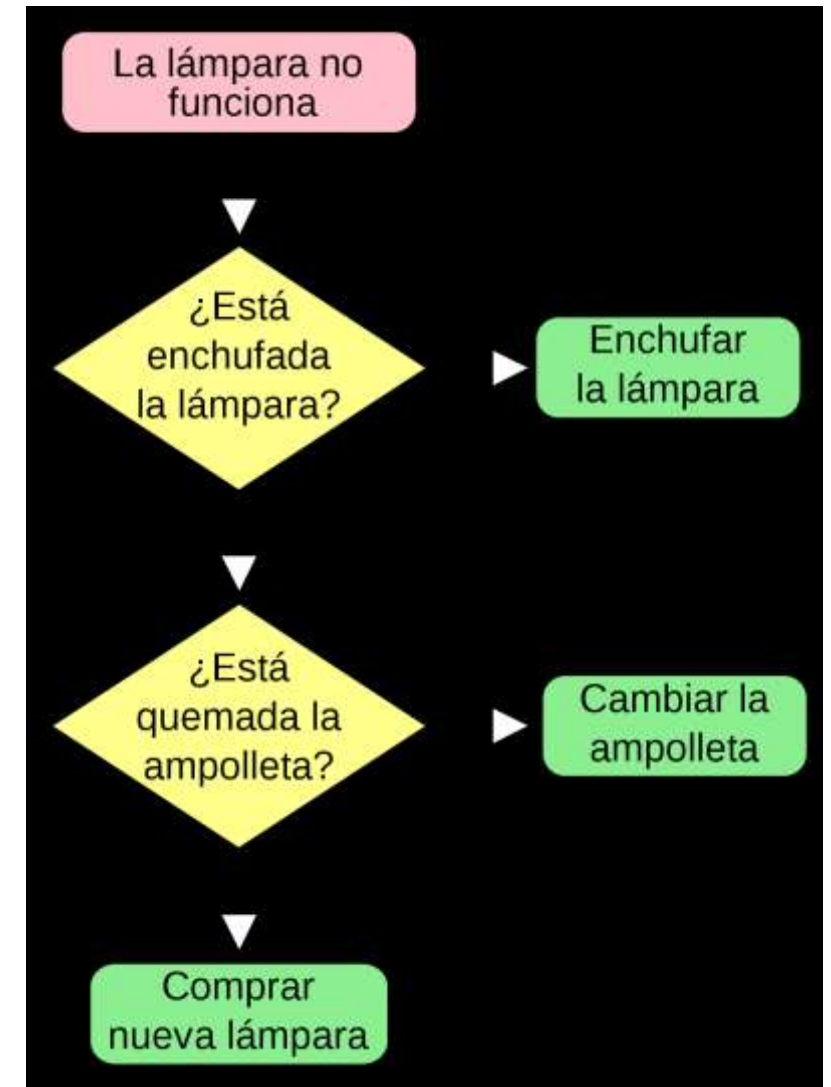
- ❑ **Diagramas de Flujo:** Son descripciones gráficas de algoritmos; usan símbolos conectados con flechas para indicar la secuencia de instrucciones, están regidos por ISO (*International Organization for Standardization*) y ANSI (*American National Standard Institute*).
- ❑ **Pseudocódigo:** Es una descripción que emplea una mezcla de lenguaje natural con algunas convenciones sintácticas propias de lenguajes de programación. No está regido por ningún estándar.

INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos - Diagramas de Flujo

Los Diagramas de Flujo se basan en símbolos que debemos usar para representar acciones, estados, procesos, datos e información para nuestro programa.

Cada elemento tiene un significado determinado y está unido a los demás por flechas que especifican el orden en que se realiza la secuencia.



INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos - Diagramas de Flujo

Los símbolos que se usan para realizar los diagramas de flujo son los siguientes:



Inicio/Final se utiliza para marcar el inicio y fin de un diagrama de flujo, es decir, siempre vamos a comenzar y terminar un diagrama con este símbolo.



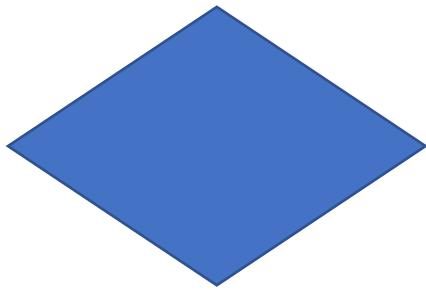
Entrada de datos se utiliza para introducir los datos de entrada, es decir, cuando queremos representar la entrada de datos (guardar una variable) por parte de un usuario.

INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos - Diagramas de Flujo



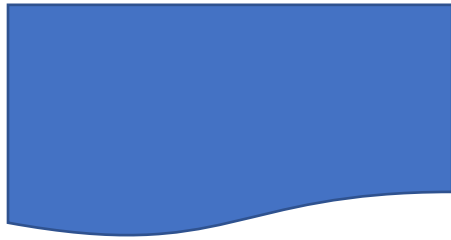
Proceso se utiliza para representar un proceso, el lugar donde se realiza las acciones correspondientes con los datos de entrada y generará un resultado. Aquí pueden existir operaciones aritméticas, asignaciones, cálculos y expresiones de todo tipo.



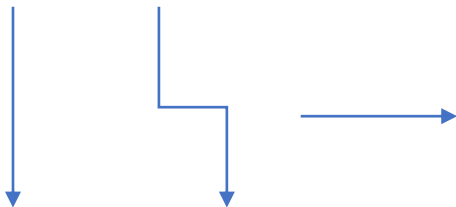
Decisión se utiliza para representar una decisión, es decir, implementa una condición que puede ser verdadera o falsa. Dependiendo el resultado, el flujo se dirige por un camino u otro.

INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos - Diagramas de Flujo



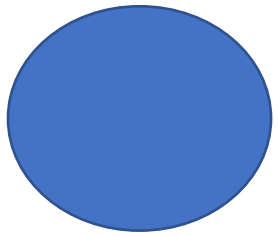
Salida impresa se utiliza para representar una impresión de resultado. Es decir, denota una escritura. Podemos usarla para escribir texto, valores, etc. Todo lo que contenga una variable o constante.



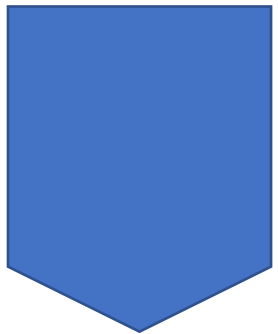
Flechas de flujo se utilizan para representar los movimientos (direcciones) de los datos a través del Diagrama. Se pueden usar en horizontal o vertical.

INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos - Diagramas de Flujo



Conector se utiliza para representar la conexión dentro de una página. Podemos usar dentro, letras o números o caracteres especiales.



Conector se utiliza para representar la conexión a páginas diferentes, es decir, si se nos terminó la hoja podemos usar otra y mostrar que el flujo sigue en esa otra hoja.



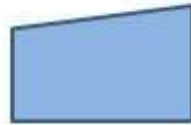
Inicio/Final

Se utiliza para indicar el inicio y el final de un diagrama; del Inicio sólo puede salir una línea de flujo y al Final sólo debe llegar una línea.



Entrada General

Entrada/Salida de datos en General(en esta guía, solo la usaremos para la Entrada).



Entrada por teclado

Instrucción de 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.



Llamada a subrutina

Indica la llamada a una subrutina o proceso predeterminado.



Acción/Proceso General

Indica una acción o instrucción general que debe realizar el computador (cambios de valores de variables, asignaciones, operaciones aritméticas etc.).



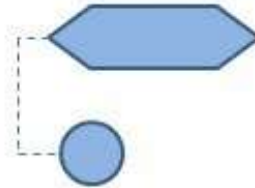
Flujo

Indica el seguimiento lógico del diagrama. También indica el sentido de ejecución de las operaciones.



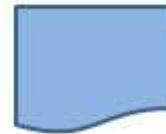
Decisión

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.



Iteración (repetición)

Indica que una instrucción o grupo de instrucciones deben ejecutarse varias veces..



Salida Impresa

Indica la presentación de uno o varios resultados en forma impresa.



Salida en Pantalla

Instrucción de presentación de mensajes o resultados en pantalla



Conector

Indica el enlace de dos partes de un diagrama dentro de la misma página.



Conector

Indica el enlace de dos partes de un diagrama en páginas diferentes.

INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos - Diagramas de Flujo

Reglas Básicas Para la Construcción de un Diagrama de Flujo:

- ✓ Por principio todo algoritmo debe tener un solo símbolo de inicio y uno de fin. Aunque el diagrama sea extenso o complejo, siempre debe terminar.
- ✓ Todos los posibles caminos deben conducir, eventualmente, al fin del algoritmo.
- ✓ Todos los símbolos han de estar conectados. No se pueden dejar líneas que no conecten nada.
- ✓ A un símbolo de proceso pueden llegarle varias líneas.
- ✓ A un símbolo de decisión pueden llegarle varias líneas, pero sólo saldrán dos (Si o No, Verdadero o Falso).

INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos - Diagramas de Flujo

Reglas Básicas Para la Construcción de un Diagrama de Flujo:

- ✓ A un símbolo de inicio nunca le llegan líneas.
- ✓ De un símbolo de fin no parte ninguna línea.
- ✓ Siempre construiremos los diagramas de arriba hacia abajo y de izquierda a derecha.
- ✓ Si el diagrama ocupa más de una hoja, usar conectores.

INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos - Diagramas de Flujo

Las Etapas en la construcción, sea de un Diagrama complejo o simple, son siempre las mismas:

- Siempre empezamos con una elipse indicando nuestro INICIO y luego indicamos el flujo del Diagrama indicando la flecha hacia abajo.
- Continuamos con el paralelogramo, indicando la lectura de datos. No importa en este momento qué tipo de datos o la cantidad, importa que aquí introduciremos los datos necesarios para resolver el problema.

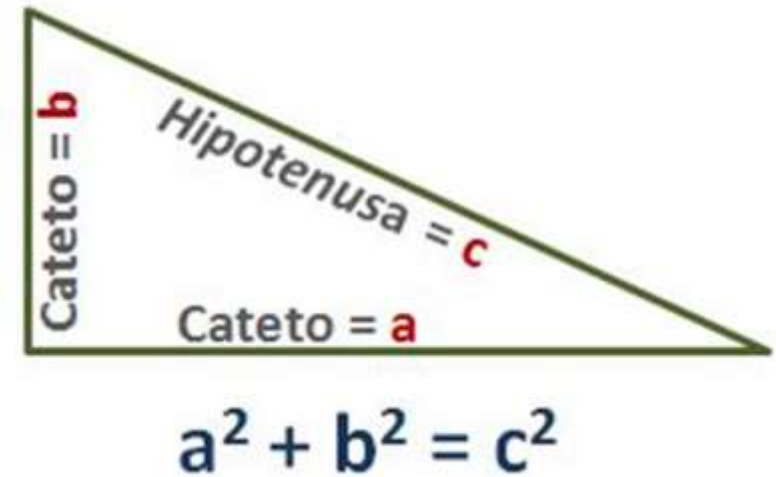
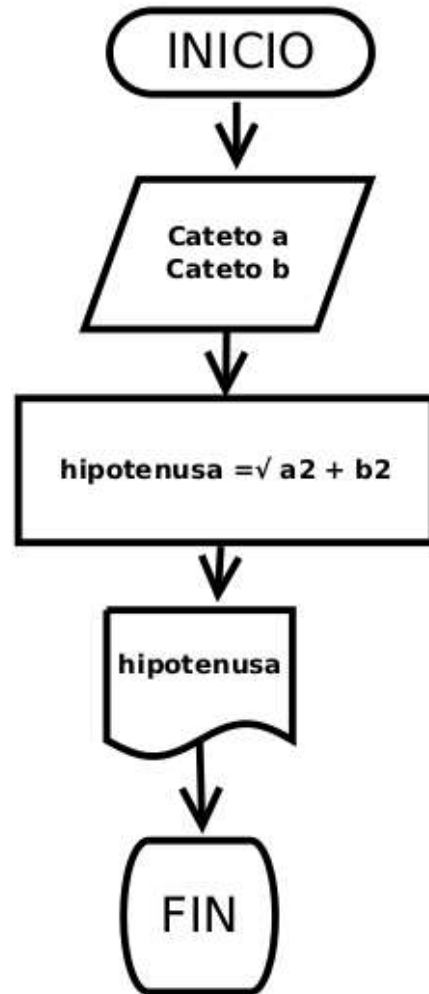
INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos - Diagramas de Flujo

- Luego, siguiendo la flecha que representa nuestro flujo, procederemos a procesar los datos, utilizando asignaciones, instrucciones, cálculos, operaciones aritméticas, etc.
- Una vez procesados los datos, la información resultante ingresa en la ante última figura que representa la impresión de la información.
- Por último todo Diagrama finaliza con el símbolo elipse de FIN.

INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.



INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos - Seudocódigo

El **pseudocódigo** es una forma de escribir los pasos, pero de una forma más cercana a lo que significaría escribir un programa directamente en un lenguaje de programación. Su sintaxis es menos específica que la de un lenguaje de programación, pero mantiene una estructura muy similar.

Normalmente esta representación se utiliza por quienes están muy familiarizados con el diseño de algoritmos, y es una representación estándar para comunicar algoritmos de manera efectiva y específicas, pero sin entrar en la sintaxis de un lenguaje de programación en particular.

INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos - Seudocódigo

Lo más común que podemos escribir en pseudocódigo es:

INICIO: Marca el inicio del algoritmo.

FIN: Marca el fin del algoritmo.

VAR: Se utiliza para declarar variables, es decir, los elementos donde guardamos los datos para su manipulación.

CONST: Se utiliza para declarar constantes, es decir, valores que se mantienen fijos en todo el programa.

LEER: Pide un dato al usuario.

ESCRIBIR: Muestra un mensaje por pantalla.

← Operador de asignación. Se emplea para asignar un valor a una variable.

INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos - Seudocódigo

INICIO CalcularHipotenusa

VAR cateto a,cateto b,hipotenusa;

ESCRIBIR "Introduce el cateto a:";

LEER cateto a;

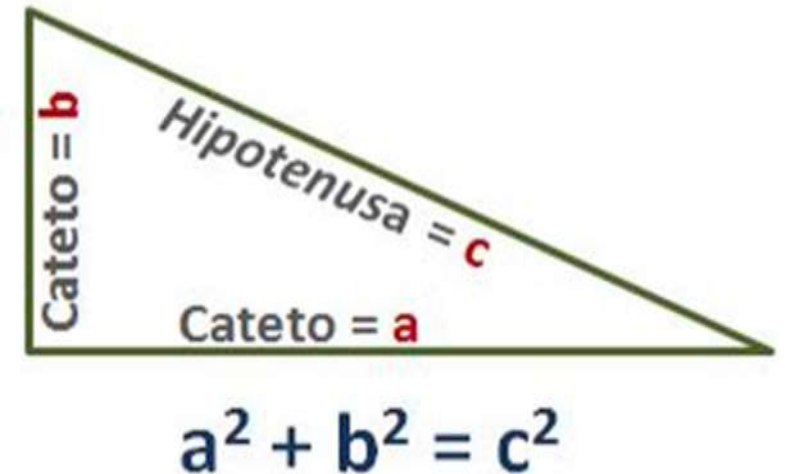
ESCRIBIR "Introduce la cateto b:";

LEER cateto b;

hipotenusa <- $\sqrt{a^2 + b^2}$;

ESCRIBIR "La hipotenusa es ",hipotenusa;

FIN



INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

Sin importar cuál sea la representación del algoritmo en cuestión, lo más importante es escribirlo de la manera más precisa y clara posible.

INTRODUCCIÓN A LA PROGRAMACIÓN

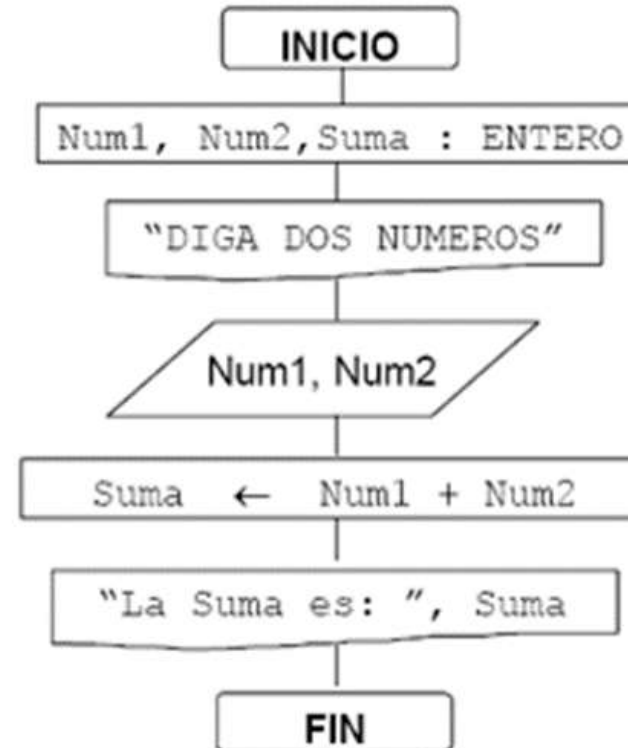
Algoritmos.

Se desea implementar un algoritmo para obtener la suma de dos números cualesquiera. Realizar el pseudocódigo y el diagrama de flujo que representen el algoritmo correspondiente.

Pseudocódigo:

```
INICIO
  Num1, Num2, Suma : ENTERO
  ESCRIBA "Diga dos números: "
  LEA Num1, Num2
  Suma ← Num1 + Num2
  ESCRIBA "La Suma es:", Suma
FIN
```

Diagrama de flujo:

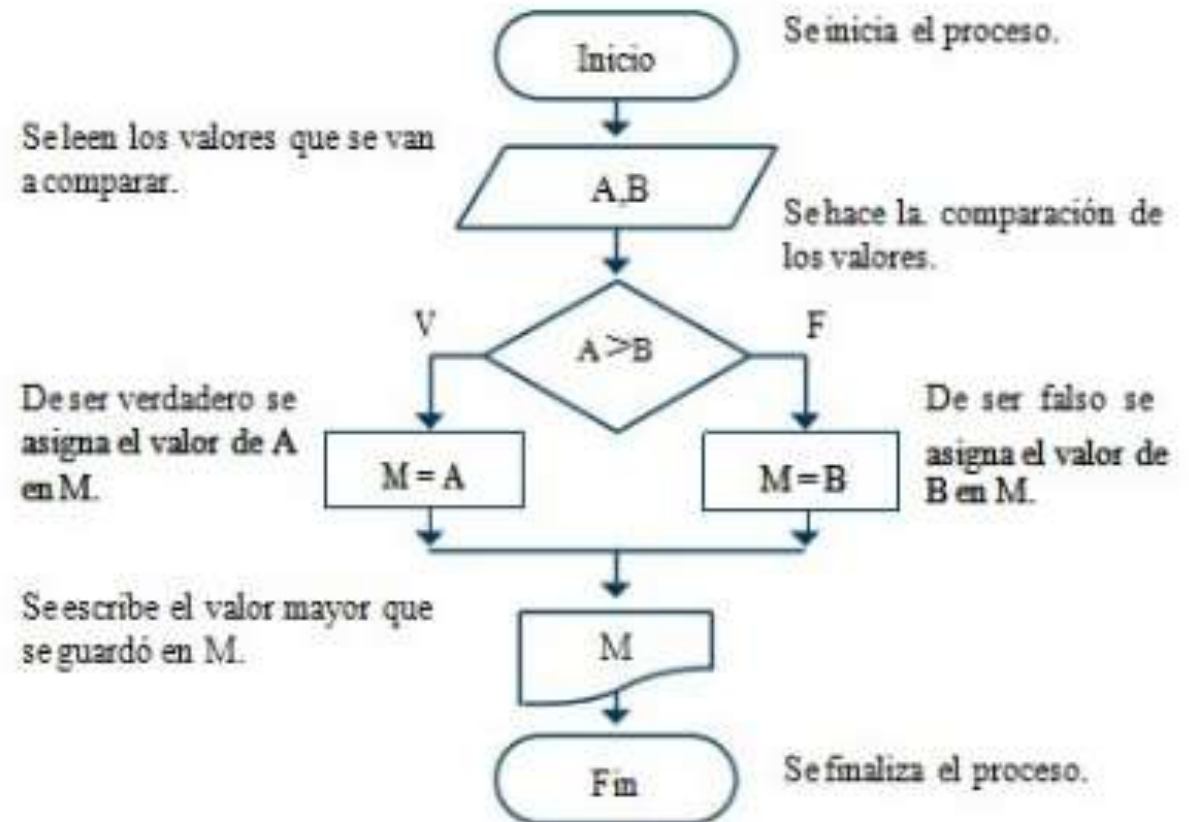


INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

Escribir el pseudocódigo y representar el diagrama para un algoritmo que determina el valor mayor para dos valores dados.

```
1. Inicio
2. Leer A, B
3. Si A > B
    Entonces
        Hacer M = A
    Si no
        Hacer M = B
    Fin de comparación
4. Escribir "el mayor es", M
5. Fin
```

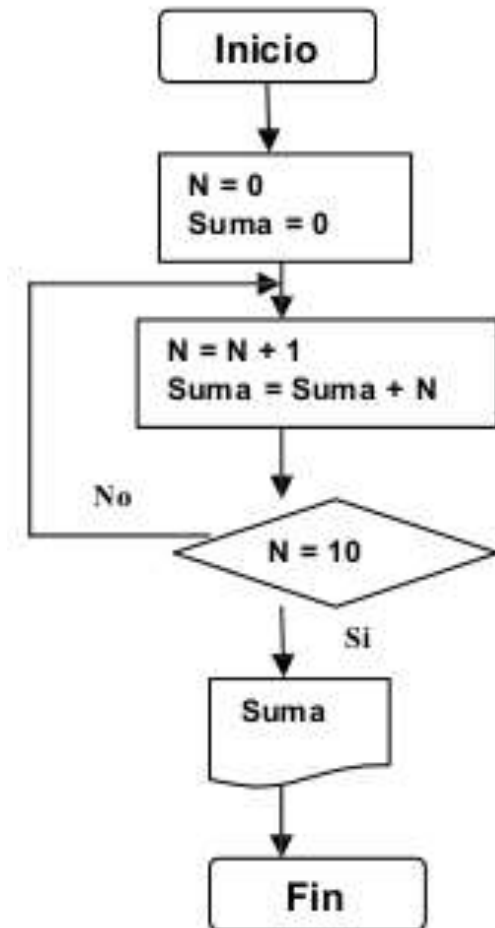


INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

Desarrolle un algoritmo que realice la sumatoria de los números enteros comprendidos entre el 1 y el 10, es decir, $1 + 2 + 3 + \dots + 10$.

1. Inicio
2. Declaración de variables: $N = 0$, $\text{Suma} = 0$
3. Asignación Contador : $N = N + 1$
4. Asignación Acumulador: $\text{Suma} = \text{Suma} + N$
5. Si $N = 10$ Entonces: Escribir Suma
De lo contrario, Repetir desde el paso 3
Fin_Si
6. Fin



INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

Ciclo de vida de desarrollo de Software

El ciclo de vida es el conjunto de fases por las que pasa el sistema que se está desarrollando desde que nace la idea inicial hasta que el software es retirado o remplazado (muere).



INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

Análisis: Comprender y definir el problema. Incluye un estudio de viabilidad (se evalúan los costes, la rentabilidad y la factibilidad del proyecto) y una definición de los requisitos (funcionalidades, limitaciones y exigencias de partida).

Diseño: Desarrollar el algoritmo que conducirá a la solución del problema.

Codificación: Transformar el algoritmo en un programa usando un lenguaje de programación.

Validación: Comprobar que el programa cumple con los requisitos especificados, tiene el funcionamiento esperado y resuelve el problema para el cual ha sido desarrollado.

Mantenimiento: Seguir dando soporte, solucionar errores no detectados en pasos anteriores y realizar mejoras con vistas a su evolución.

INTRODUCCIÓN A LA PROGRAMACIÓN

Algoritmos.

“Fin del tema”

