

TEMA1: Algoritmos y Programación.

Contenido

1.- Definiciones.....	2
2.- CARACTERÍSTICAS DE UN ALGORITMO:.....	2
3.- PARTES DE UN ALGORITMO:.....	2
3.1. EJEMPLO DE ALGORITMO:	2
4.- CICLO DE PROGRAMACIÓN:.....	3
5.- ALGORITMO EN PSEUDOCÓDIGO:	3
6.- Diagramas de Flujo.....	4
6.1.- Definición	4
6.2.- Simbología de los diagramas de flujo:	4
6.3.- Reglas para elaborar diagrama de flujo:	5
6.4.-Extensión del navegador Chrome para diagramas de flujo	5
6.5.-EJEMPLO: ALGORITMO DE EUCLIDE:	5
6.6.-Ejemplos completos de algoritmia.	6
EJEMPLO SOBRE CALCULAR EL ÁREA DE UN TRIÁNGULO.....	6
EJEMPLO 2: Complementar la solución del problema anterior ,pero ahora puedes calcular el perímetro aplicando la fórmula (afinamiento).....	7
7.- CONCEPTOS BÁSICOS DE PROGRAMACIÓN.....	8
7-1- Qué es la programación.....	8
7.2.- ¿Qué conceptos debemos conocer antes de ponernos a programar?.....	8
7.3. TIPOS DE DATOS.....	8
7.4.- OPERADORES	8
7.5.- ESTRUCTURA DE CONTROL.....	8
7.5.1. ESTRUCTURA SECUENCIAL	8
7.5.2.- ESTRUCTURA ITERATIVA	9
7.5.3 ESTRUCTURA CONDICIONAL	9
7.6.- DEPURACIÓN:.....	9

1.- Definiciones.

ALGORITMO: conjunto de instrucciones que realizados en orden conducen a obtener la solución de un problema.

PROGRAMA: serie de instrucciones ordenadas, codificadas en in lenguaje de programación, que expresa un algoritmo y que puede ser ejecutado en un ordenador.

2.- CARACTERÍSTICAS DE UN ALGORITMO:

PRECISO: cada paso debe indicar de manera precisa e inequívoca lo que debe de hacer.

FINITO: debe tener un número de pasos finitos.

DEFINIDO: debe de producir los mismos resultados para las mismas condiciones de entrada.

3.- PARTES DE UN ALGORITMO:

-Entrada de datos

-Proceso

-Salida de datos

3.1. EJEMPLO DE ALGORTIMO:

Título: "Hacer una tortilla de patatas."

ENTRADA-> huevos, aceite, patatas, sal, salten y fuego

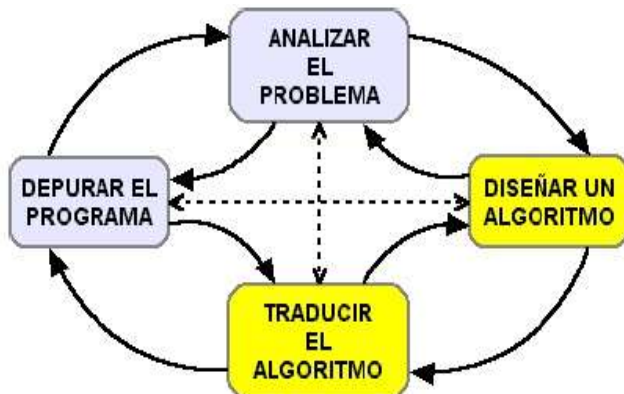
PROCESO->1ª pelar las patatas

2ª calentar el aceite

....

SALIDA-> Tortilla de patata

4.- CICLO DE PROGRAMACIÓN:



5.- ALGORITMO EN PSEUDOCÓDIGO:

El **pseudocódigo** (o falso lenguaje) es comúnmente utilizado por los programadores para omitir secciones de código o para dar una explicación de lo que debería hacer un algoritmo. En un lenguaje humano y no comprensible por la máquina.

Lo hemos visto antes con un ejemplo y a continuación os pongo otro sobre el algoritmo de Euclides para calcular el MCD.:

EJ: ALGORITMO DE EUCLIDE PARA CALCULAR EL MÁXIMO COMÚN DIVISOR DE DOS NÚMEROS ENTEROS MCD (200,30)

1. ALGORITMO EN PSEUDOCÓDIGO
 - 1.1 Leer los números "a" y "b". Ir al paso 2
 - 1.2 Compara si "a" ≠ "b", si son iguales, el MCD es "A". Ir al paso 4
 - 1.3 A < B
 - NO : A=A-B
 - SI: B=B-A
 - IR AL PASO 2
 - 1.4 El resultado es "A"

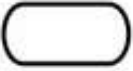


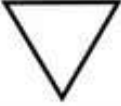




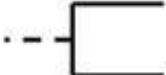


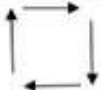


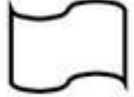

El pseudocódigo está compuesto por proposiciones informales en español (si hablamos de nuestro idioma), que permiten expresar detalladamente las instrucciones que llevan desde un estado inicial (problema) hasta un resultado deseado (solución). Los algoritmos suelen refinarse y descomponerse en subproblemas.

6.- Diagramas de Flujo.

6.1.- Definición

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 de proceso. Los símbolos gráficos del **flujo** del proceso están unidos entre sí con flechas que indican la dirección de **flujo** del proceso.

6.2.- Simbología de los diagramas de flujo:

SÍMBOLO	REPRESENTA	SÍMBOLO	REPRESENTA
	Terminal. Indica el inicio o la terminación del flujo, puede ser acción o lugar; además se usa para indicar una unidad administrativa o persona que recibe o proporciona información.		Documento. Representa cualquier tipo de documento que entra, se utilice, se genere o salga del procedimiento.
	Disparador. Indica el inicio de un procedimiento, contiene el nombre de éste o el nombre de la unidad administrativa donde se da inicio.		Archivo. Representa un archivo común y corriente de oficina.
	Operación. Representa la realización de una operación o actividad relativas a un procedimiento.		Conector. Representa una conexión o enlace de una parte del diagrama de flujo con otra parte lejana del mismo.
	Decisión o alternativa. Indica un punto dentro del flujo en que son posibles varios caminos alternativos.		Conector de página. Representa una conexión o enlace con otra hoja diferente, en la que continúa el diagrama de flujo.
	Nota aclaratoria. No forma parte del diagrama de flujo, es un elemento que se adiciona a una operación o actividad para dar una explicación.		Línea de comunicación. Proporciona la transmisión de información de un lugar a otro mediante?
SÍMBOLO	REPRESENTA	SÍMBOLO	REPRESENTA
	Operación con teclado. Representa una operación en que se utiliza una perforadora o verificadora de tarjeta.		Dirección de flujo o línea de unión. Conecta los símbolos señalando el orden en que se deben realizar las distintas operaciones.
	Tarjeta perforadora. Representa cualquier tipo de tarjeta perforada que se utilice en el procedimiento.		Cinta magnética. Representa cualquier tipo de cinta magnética que se utilice en el procedimiento.
	Cinta perforada. Representa cualquier tipo de cinta perforada que se utilice en el procedimiento.		Teclado en línea. Representa el uso de un dispositivo en línea para promocionar información a una computadora electrónica u obtenerla de ella.

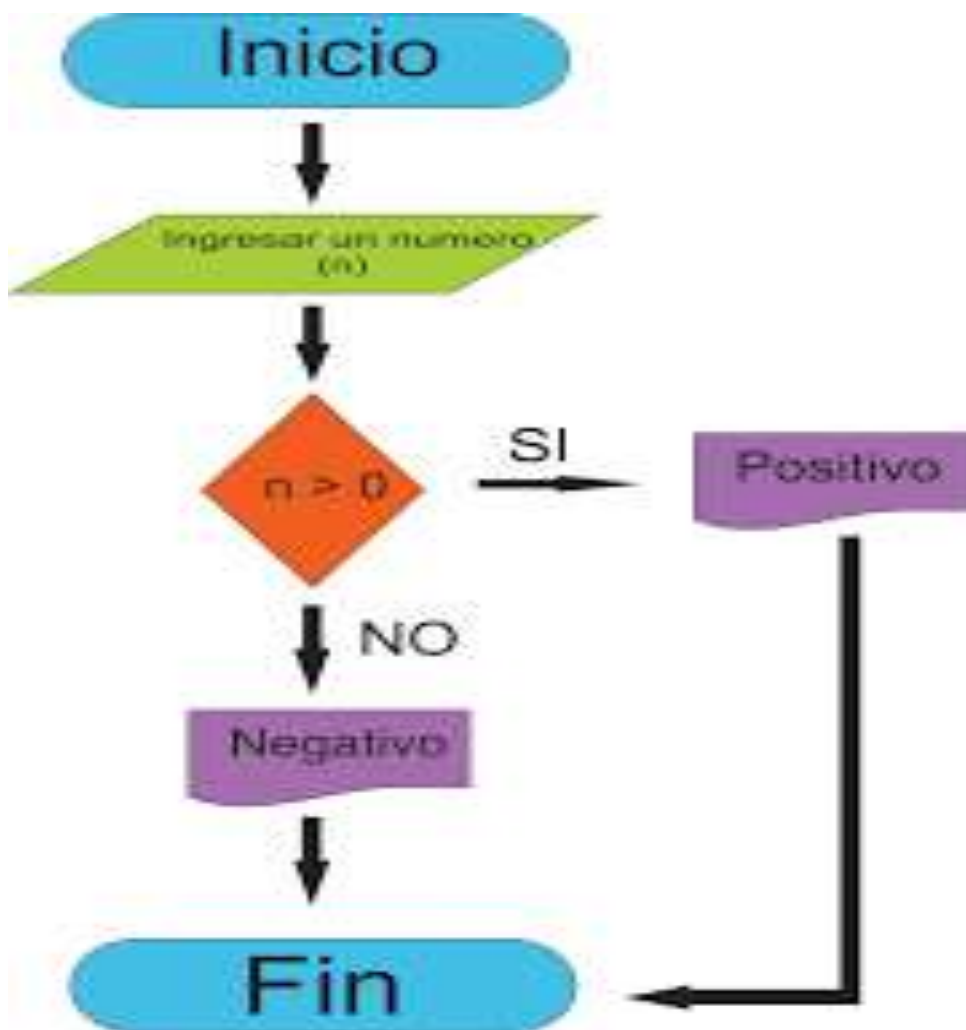
6.3.- Reglas para elaborar diagrama de flujo:

- Encabezado con el título, autor, fecha de elaboración y versión.
- Los diagramas se dibujan de arriba abajo y de izquierda a derecha.
- Inicio y final solo aparecen una vez.
- La dirección de flujo se representa por flechas.
- Cada rombo debe tener al menos dos líneas, si y no.
- Se debe probar con datos iniciales y simples.

6.4.-Extensión del navegador Chrome para diagramas de flujo

<https://chrome.google.com/webstore/detail/drawio-diagrams/onlkggianjhjenigcpigpjehhpplldkc>

6.5.-EJEMPLO: ALGORITMO DE EUCLIDE:



6.6.-Ejemplos completos de algoritmia.

EJEMPLO SOBRE CALCULAR EL ÁREA DE UN TRIÁNDULO.

Diseñamos un algoritmo (pseudocódigo y diagrama de flujo) para hallar el área de un triángulo cuya base mide 3cm, la altura 4cm y la hipotenusa 5cm.

1. ANÁLISIS DEL PROGRAMA.

1.1 Formulación del programa -> el problema está bien planteado.

1.2 Resultado esperado -> el resultado esperado es el área del triángulo.

1.3 Datos-> Base=3cm

Altura=4cm

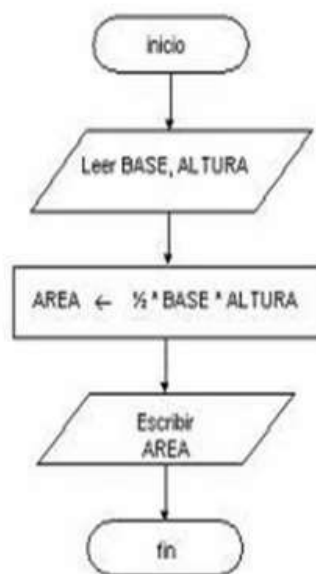
Hipotenusa=5cm

Área =base * altura \2

2. ALGORITMO EN PSEUDOCÓDIGO

1. Inicio
2. Asignamos número 2 a la constante "div"
3. Asignamos número 3 a la constante "base"
4. Asignamos número 4 a la constante "altura"
5. Guardar en la variable 2 "area" el resultado de base*altura/div
6. Imprimir el valor del área
7. Fin

3. DIAGRAMA DE FLUJO



EJEMPLO 2: Complementar la solución del problema anterior ,pero ahora puedes calcular el perímetro aplicando la fórmula (afinamiento)

$$P=B+A+\text{Hipotenusa}$$

$$\text{Sabiendo que la hipotenusa es } H^2=B^2+A^2$$

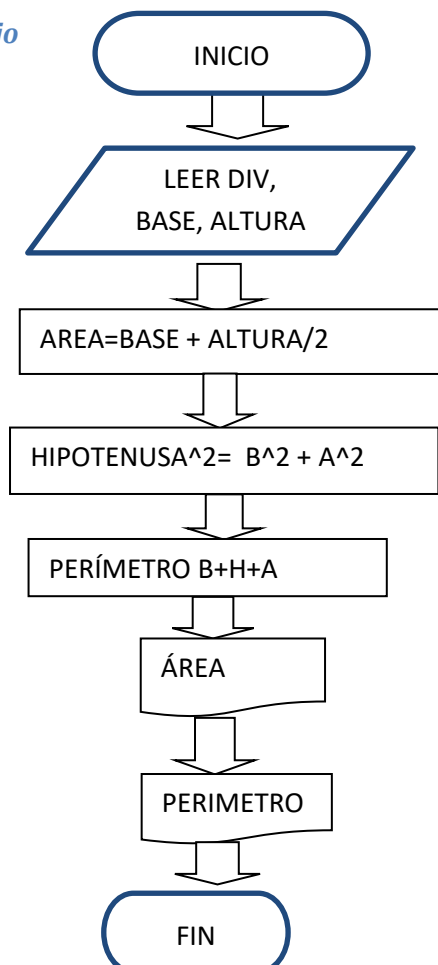
ANÁLISIS DEL PROBLEMA

- 1.1 Formulación del problema-> el problema está bien formulado
- 1.2 Resultado esperado -> el resultado esperado es el área y el perímetro del triángulo
- 1.3 Datos -> base=3cm
Altura=4cm
Hipotenusa²= B²+A²
Perímetro= b + a + h

ALGORITMO EN PSEUDOCODIGO

- 2.1 Inicio
- 2.2 Asignamos el valor 2 a la constante "div"
- 2.3 Asignamos el valor 3 a la constante base
- 2.4 Asignamos el valor 4 a la constante altura
- 2.5 Guardar en la variable "area" el resultado de b*a/div
- 2.6 Asignamos a la constante hipotenusa ² el valor b²+a²

3. Diagrama de flujo



7.- CONCEPTOS BÁSICOS DE PROGRAMACIÓN

7-1- Qué es la programación.

Es el proceso de diseñar, codificar, depurar y mantener el código fuente de programas de cualquier dispositivo electrónico que pueda interactuar con el ser humano. El código fuente es escrito en un lenguaje de **programación**.

7.2.- ¿Qué conceptos debemos conocer antes de ponernos a programar?

- **VARIABLE:** Identificador que puede tomar cualquier valor en la ejecución de un programa
- **CONSTANTE:** Datos que se almacenan bajo un nombre y que su valor no cambia durante el transcurso del programa
- **CONTADORES:** Variables que almacenan un valor que sube o baja de forma constante durante el programa
- **ACUMULADORES:** Variable que almacenan el acumulado de un valor

7.3. TIPOS DE DATOS

En ciencias de la computación, un tipo de dato informático o simplemente tipo, es un atributo de los datos que indica al ordenador (y/o al programador) sobre la clase de datos que se va a manejar. Esto incluye imponer restricciones en los datos, como qué valores pueden tomar y qué operaciones se pueden realizar.

Los tipos de datos más comunes son: números enteros, números con signo (negativos), números de coma flotante (decimales), cadenas alfanuméricas, fechas, lógicos y datos tipo OLE o multimedia.

7.4.- OPERADORES

- **ARITMETICOS** → +, -, *, =
- **ALFANUMÉRICOS** → Concatenar, etc.
- **RELACIONALES** → verdadero y falso , = , < ,
- **LÓGICOS:** verdadero , falso , negación , conjunción , y disyunción > True false Φ

7.5.- ESTRUCTURA DE CONTROL

7.5.1. ESTRUCTURA SECUENCIAL

Formada por instrucciones que deben ejecutarse una seguida de otra.

Suelen usarse mezclada con estructura iterativa y condicionales.

7.5.2.- ESTRUCTURA ITERATIVA

Permite ejecutar una o varias acciones , un número determinado de veces, mientras se cumpla una condición.

Hay dos tipos :

1. DEFINIDAS : Se repite un número determinado de veces
2. INDEFINIDAS : Mientras se cumpla una condición

7.5.3 ESTRUCTURA CONDICIONAL

Evalúa una expresión a partir del resultado. Y ejecuta el bloque al instrucciones correspondientes .

- SELECCION SIEMPLE: Si la condición es verdadera se hace algo
- SELECCION DOBLE : Se realiza un bloque si la condición es verdadera y otro bloque si es falsa
- SELECCIÓN ANIMADA: Una dentro de otra

7.6.- DEPURACIÓN:

Después de traducir el algoritmo en un lenguaje, el procedimiento resultante debe ser probado y los resultados validados. A este proceso se llama depuración.

Se identifica las fallas sintácticas o lógicas, que una vez aislada se vuelve a probar el programa.