

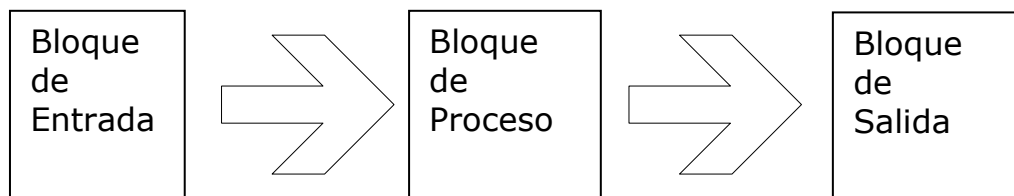
Introducción a la programación

Resolución de Problemas

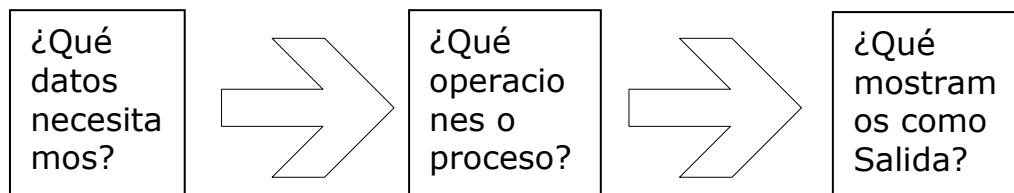
El objetivo principal para que las personas aprendan a programar en algún lenguaje de programación en particular es utilizar el computador como una herramienta para la resolución de problemas. Así, la resolución de un problema se puede dividir en tres etapas:

1. Un primer nivel de refinamiento **"Análisis del problema"**

Que dará como resultado un diagrama en bloques **Entrada, Proceso, Salida**. Requiere que el problema sea definido y comprendido claramente para que pueda ser analizado con todo detalle. Para poder definir con precisión el problema se requiere que las especificaciones de entrada y salida sean descriptas en detalle. El análisis debe responder a dos preguntas: Primero para definir el bloque de **Entrada** ¿Que datos se necesitan para resolver el problema? Segundo Para definir el bloque de **Salida** ¿Qué información debe proporcionar la resolución del problema? y Tercero para el bloque de **Proceso** ¿cuales son las operaciones o procesos que debo aplicar para convertir esos datos en esa información?



Aquí están las preguntas para cada bloque de manera de determinar que va en cada uno:



2. Un segundo nivel de refinamiento **"Diseño o desarrollo del algoritmo"**

Este es un procedimiento paso a paso para solucionar el problema dado utilizando estructuras de programación a través de herramientas tales como el diagrama de flujo y el pseudocódigo.

3. Un tercer nivel de refinamiento **"Resolución del algoritmo en el computador"**

Se necesita codificar el algoritmo en un lenguaje de programación (en este caso se usará el lenguaje C). Es decir, se convertirá el algoritmo en programa, luego ejecutarlo y comprobar que el programa soluciona verdaderamente el problema.

Ejemplo:

Leer el radio de un círculo y calcular e imprimir su superficie y circunferencia.

Análisis: Las entradas de datos en este problema se concentran en el radio del círculo. Dado que el radio puede tomar cualquier valor dentro del rango de los números reales, el tipo de datos radio debe ser real. Las salidas serán dos variables: superficie y circunferencia que también serán de tipo real

Entrada	Proceso	Salida
(Radio)	Calcular Área Calcular perímetro	(Área) (Perímetro)

El algoritmo sería así:

Leer radio
Calcular Área
Calcular Circunferencia
Escribir resultados

Mas refinado sería así:

Leer radio
área = 3.141592 * radio * radio
circunferencia = 2 * 3.141592 * radio
escribir área, circunferencia

Como vemos en este algoritmo mas refinado hemos utilizado algunas palabras tales como, "radio" "área" y "circunferencia" para denominar a las variables que van a contener los datos numéricos o que van a ser ingresados procesados y mostrados. Estos nombres, que hacen referencia a una posición de memoria del computador donde se encuentran almacenados estos datos, se denominan "**Identificadores**".

Identificadores

Representan los nombres de los objetos de un algoritmo o programa (constantes, variables, tipos de datos, procedimientos, funciones, etc.). Es una secuencia de caracteres que puede ser de cualquier longitud, aunque tiene ciertas reglas que hay que seguir, las cuales son:

Debe comenzar con una letra, luego una sucesión de letras, números y caracteres subrayados ("_") y no puede contener espacios en blanco.

Esta sucesión elegida no debe ser demasiado extensa no mas de 8 o a lo sumo 10 caracteres y debe dar idea de la función y el tipo de dato que va a almacenar para mejorar la legibilidad del algoritmo en revisiones posteriores.

En síntesis un identificador es un método para nombrar a las celdas de memoria en la computadora, en lugar de memorizarnos una dirección de memoria. Se utilizan para nombrar variables, constantes, procedimientos y funciones que vamos a aprender mas adelante.

Constantes: Las constantes son valores que no pueden cambiar en la ejecución del algoritmo o (prueba de escritorio) programa.

Variables: Las variables son valores que se pueden modificar durante la ejecución de un algoritmo (prueba de escritorio) o programa. Al contrario de las constantes estos reciben un valor, pero este valor puede ser modificado durante la ejecución.

DIAGRAMA EN BLOQUES

PROCESO DE ENTRADA SALIDA

Trabajo Práctico numero 1

Actividad 1:

Rellene la tabla siguiente de entrada/proceso/salida

Entrada	Proceso	Salida
	Calcular área de un triángulo rectángulo	
	Calcular área de un cilindro	
	Calcular volumen de un cilindro	
	Calcular el espacio recorrido por un móvil que se mueve con mru	
	Calcular el espacio recorrido por un móvil que se mueve con mruv	
	Calcular volumen de un cubo	
	Calcular área de un triangulo dando solo sus lados	

Actividad 2:

¿Cómo seria el algoritmo para cada fila de la tabla? describa cada una mediante un diagrama en bloques y luego por una sucesión de acciones refinando el algoritmo

Ahora estamos en condiciones de hacernos la pregunta ¿QUÉ ES UN ALGORITMO? y poder definirlo más precisamente:

La palabra algoritmo se deriva de la traducción al latín de la palabra árabe Al'Khwarizmi o Al'Khorezmi, nombre de un matemático y astrónomo árabe que escribió un tratado sobre manipulación de números y ecuaciones en el siglo IX.

Un algoritmo es un conjunto finito de instrucciones o pasos organizados que describe el proceso que se debe seguir, para dar solución a un problema específico.

De un modo más formal, un algoritmo es una secuencia finita de operaciones realizables, no ambiguas, cuya ejecución da una solución de un problema.

Sistema por el cual se llega a una solución, teniendo en cuenta que debe de ser: definido, finito y preciso. Por preciso entendemos que cada paso a seguir tiene un orden; finito implica que tiene un determinado número de pasos, o sea que tiene un fin; y definido que si se sigue el mismo proceso más de un vez llegaremos al mismo resultado.

Lenguajes Algorítmicos

Un Lenguaje algorítmico es una serie de símbolos y reglas que se utilizan para describir de manera explícita un proceso.

Tipos de Lenguajes

- **No Gráficos:** Representa en forma descriptiva las operaciones que debe realizar un algoritmo

Pseudocódigo

INICIO

Imp "Progr. que calcula el área de un triangulo"; /*Así se aclara(también en C)*/

Imp "Ingrese la base y la altura del triangulo"; /*esta es una salida */

Leer Base, h; /*esta es una entrada */

Sup=Base*h/2; /* expresión matemática */

Imp "La superficie calculada es:",Sup; /*esta es una salida */

Fin

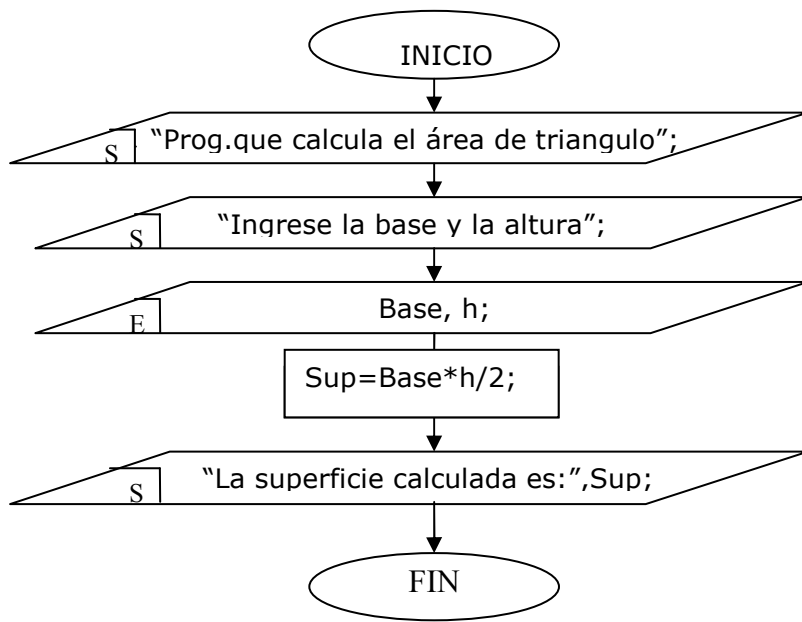
En la línea señalada hay una expresión matemática que es un conjunto de datos o funciones unidos por operadores aritméticos, que podemos utilizar en nuestros algoritmos (son los mismos en lenguaje C) y los cuales se muestran en la siguiente tabla:

Operadores Aritméticos

Operador	Descripcion	Ejemplo de aplicacion
=	Asignación: copia el contenido de var2 en var1	var1=var2
+	Suma	var1+var2
-	Resta	var2-var3
*	Multiplicacion	var3*var4
/	Division	var3/var4
++	incremento	++var3
--	Decremento	--var3

En esta tabla los operadores están ordenados de menor a mayor orden de predececion y todos salvo los monarios como el incremento y el decremento se ejecutan de izquierda a derecha como en matemáticas. Salvo el primero la asignación.

- **Gráficos:** Es la representación gráfica de las operaciones que realiza un algoritmo mediante una secuencia de ordinogramas o símbolos gráficos dada por las líneas que los unen representando entradas salidas y procesos ,todos los ordinogramas mas usados se encuentran la tabla que presentamos a continuación del ejemplo.

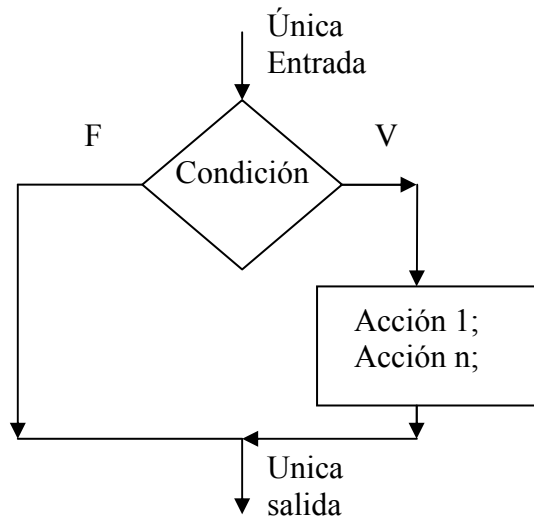
Diagrama de flujo ejemplo

Símbolo diagrama de flujo	Codificación pseudocódigo	Descripción
	Inicio	Inicio y declaración de variables
	Acción	Acción a realizar
	Si	Decisión
	Leer / Imp	Entrada / Salida Datos
	Para (Inic; Cond; Inc)	Bucle Repetitivo For cuando se conoce la cant de veces a repetir
	Según sea (Var) igual a caso 1 caso 2...ot Etc	Decisiones múltiples en donde la Cond es una variable que se compara con constantes

Estructura selectiva simple

En muchas ocasiones para poder resolver un problema debemos evaluar expresiones lógicas que tienen un valor de verdad que puede ser verdadero o falso. Si solo se desea ejecutar una o mas acciones si una expresión es verdadera, se debe utilizar esta estructura ya que cumple esta función. La forma en que se grafica y se escribe en los dos lenguajes algorítmicos es la siguiente:

Diagrama de Flujo



Codificación Pseudocódigo

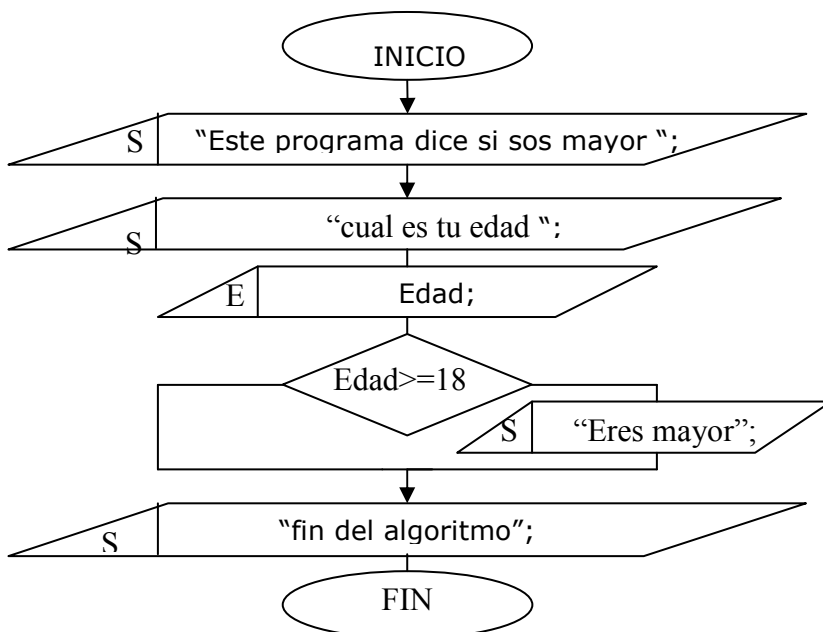
```

Si (Condición)
{
    Acción 1;
    Acción n;
}
  
```

Si es una sola la acción a ejecutar no hace falta poner las llaves para marcar el bloque.

Ejemplo:

Diagrama de flujo



Pseudocódigo

```

INICIO
  Imp "Este programa dice si sos mayor de Edad";
  Imp "cual es tu edad?";
  Leer Edad ;
  SI (Edad >=18)
    Imp "Eres mayor de Edad";
    Imp "fin del algoritmo" ;
  FIN
  
```

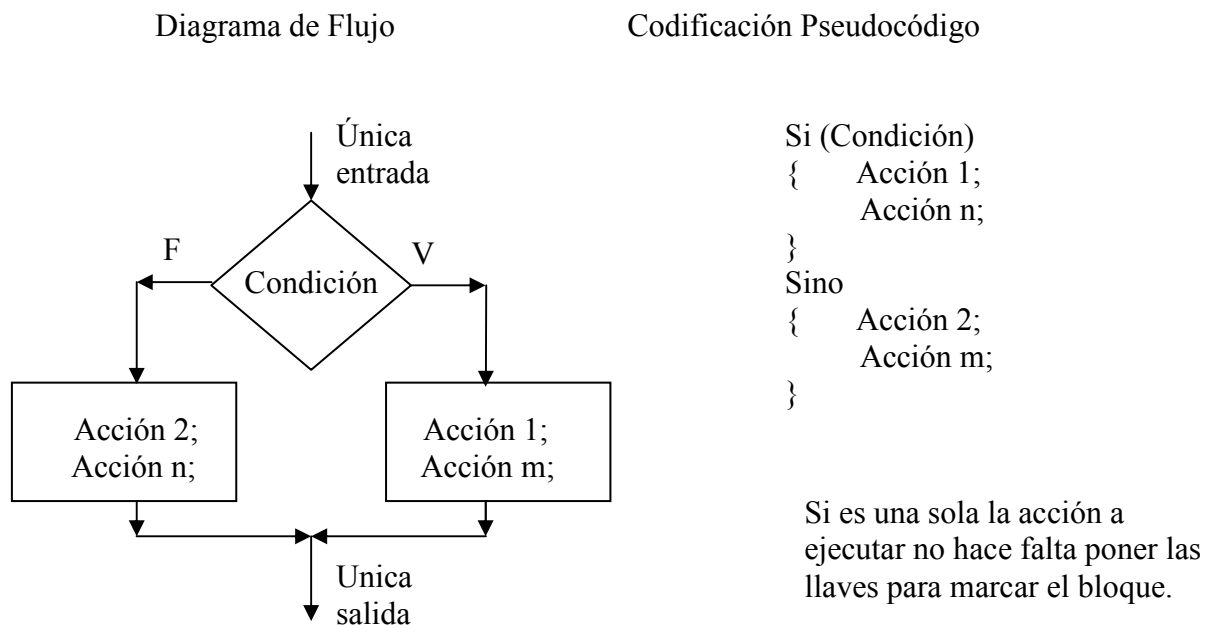
Operadores Lógicos

Como dijimos se necesitan realizar a menudo comparaciones entre distintos valores como en el algoritmo anterior (Edad ≥ 18), esto se realiza utilizando los operadores relaciones, los cuales se listan a continuación:

Operador	Descripcion	Ejemplo de aplicacion
>	Mayor Da verdadero Si el contenido de var1 es mayor que el de var2	var1>var2
>=	Mayor o igual: Da verdadero si el contenido de var1 es mayor o igual que el de var2	var1>=var2
<	Menor Da verdadero si el contenido de var2 es menor que el de var3	var2<var3
<=	Menor o igual Da verdadero si el contenido de var3 es menor o igual que el de var4	var3<=var4
==	Igual Da verdadero si el contenido de var3 es igual que el de var4	var3==var4
!=	Distinto Da verdadero si el contenido de var3 es distinto que el de var4	var3!=var4
!	NOT Niega el valor de verdad de la variable	!var3
&&	And Da verdadero si ambas proposiciones son verdaderas	(var1==var2)&&(var1==var3)
 	Or Da verdadero si alguna es verdadera	(var1==var2) (var1==var3)

Estructura selectiva doble

En otras ocasiones para poder resolver un problema debemos evaluar expresiones lógicas que tienen un valor de verdad que puede ser verdadero o falso y realizar procesos diferentes en cada caso. Si se desea ejecutar una o mas acciones si una expresión es verdadera, y otro conjunto de acciones distintas si es falsa, se debe utilizar esta estructura ya que cumple esta función. La forma en que se grafica y se escribe en los dos lenguajes algorítmicos es la siguiente:



Trabajo Práctico numero 2

Actividad 1:

Diseñar un programa en diagrama de flujo y pseudocódigo, por cada fila de la tabla del trabajo práctico anterior

Actividad 2:

Diseñar un programa en diagrama de flujo y pseudocódigo, que determine el mayor de dos números distintos (Usar estructura selectiva simple).

Actividad 3:

Diseñar un programa en diagrama de flujo y pseudocódigo que determine el menor de dos números y si son iguales indicarlo (Usar estructura selectiva simple)

Actividad 4:

Diseñar un programa en diagrama de flujo y pseudocódigo que determine el mayor de tres números (Usar estructura selectiva doble)

Actividad 5:

Diseñar un programa en diagrama de flujo y pseudocódigo que determine el mayor y el menor de cuatro números (Usar estructura selectiva doble)