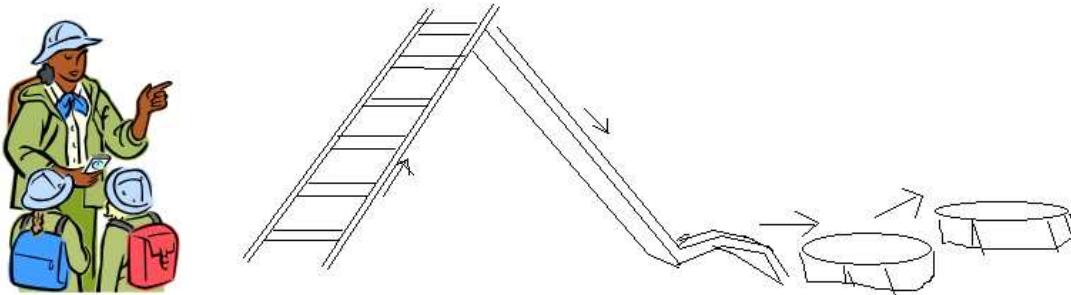


Algoritmo: Procedimiento con un numero finito de pasos, cada uno de estos preciso y suficientemente simple, y cuya ejecución (la del procedimiento) siempre termina.

Características de un algoritmo: (5)

Precisión, Simplicidad, Finitud, Carácter mecánico y procedimiento general



Preciso: Sin dudas, sobre el paso que sigue.

Simple: reglas sencillas.

Finito: ordenes o reglas limitadas, pero no el número de operaciones.

Mecanizado: Automáticamente se va realizando sin pensar mucho.

General: El procedimiento es englobado en un rango de respuestas, nunca a una sola.

Tipos de algoritmos:

Gráficos. Como el **diagrama de flujo** que gráficamente vemos los pasos.

No Gráficos. Como el **seudocódigo** el cual muestra jerárquicamente que hacer primero y que después.

Etapas de los algoritmos:

- a) **Análisis del problema:** que tipos de datos de entrada y salida requerimos, con observación de sus restricciones propias.
- b) **Construcción:** Describir los pasos para resolver un problema.
- c) **Verificación** con prueba de escritorio.

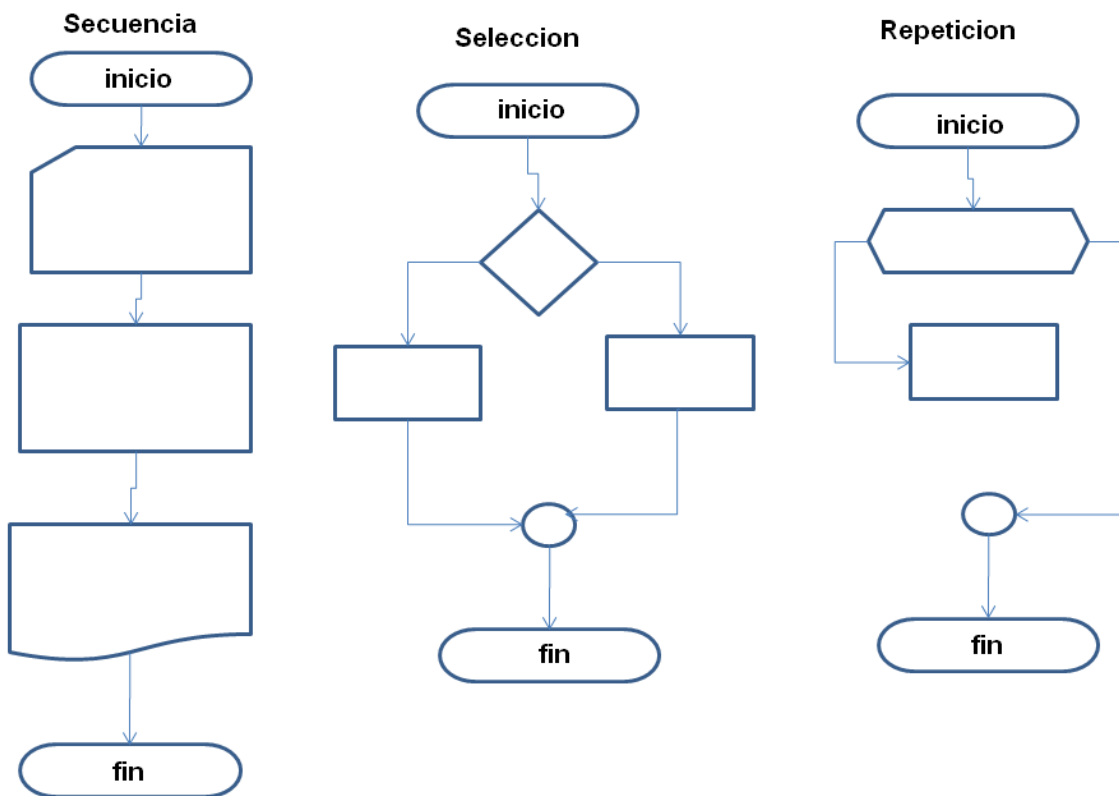
PROCESOS BASICOS DE PROGRAMACION

Los tres procesos básicos de programación estructurada son: **Secuencia, Selección y Repetición**; Estos procesos pueden representarse gráficamente con los Diagramas de Flujo.

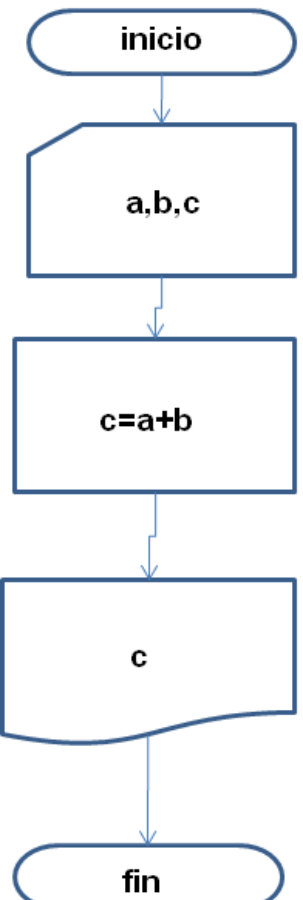
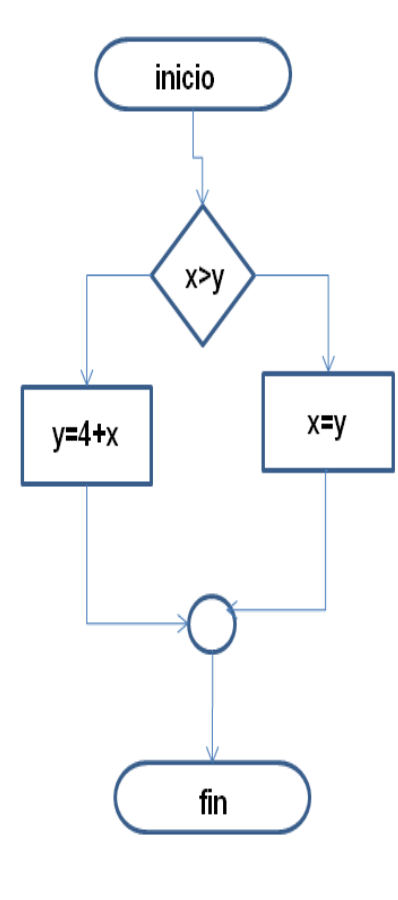
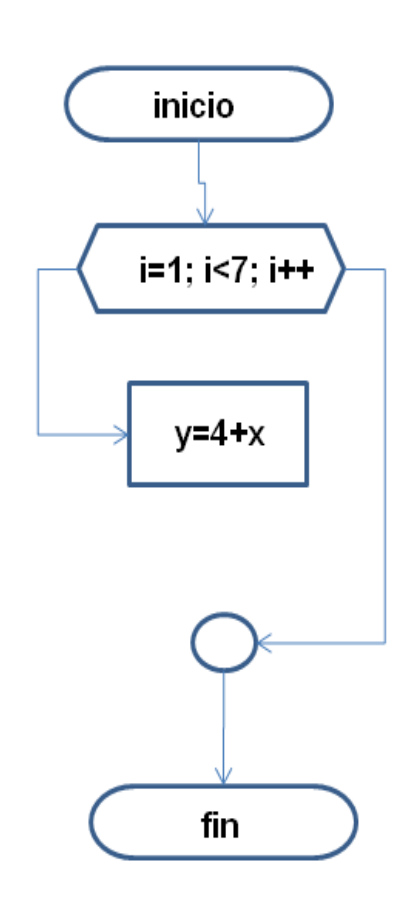
Un **diagrama de flujo** debe contener un inicio y un fin; dentro contiene cajas cuadradas de órdenes que se colocan una tras otra, indicando el flujo con flechas para saber que debe hacerse primero y que después.

Cada Caja cuadrada puede sustituirse por:

- Una entrada de datos (caja recortada en la parte izquierda superior o por un rombo)
- Una salida de impresión (caja deformada en la parte inferior como una onda)
- Un conector (circulo)
- Proceso básico de selección
- Proceso básico de Repetición
- Proceso básico de secuencia
- Otro



Veamos estos procesos en ejemplos.

Secuencia	Selección	Repetición
<p>a) Diagrama de flujo</p>  <pre> graph TD inicio([inicio]) --> abc[/a,b,c/] abc --> ca+b[c=a+b] ca+b --> c[/c/] c --> fin([fin]) </pre>	<p>a) Diagrama de flujo</p>  <pre> graph TD inicio([inicio]) --> xy{x>y} xy --> y4x[y=4+x] xy --> xy[x=y] y4x --> join(()) xy --> join join --> fin([fin]) </pre>	<p>b) Diagrama de flujo</p>  <pre> graph TD inicio([inicio]) --> loop{i=1; i<7; i++} loop --> y4x[y=4+x] y4x --> loop loop --> join(()) join --> fin([fin]) </pre>
<p>b) Lenguaje C</p> <pre> #include <stdio.h> int a, b; main() { a=4; b=5; c=a+b; printf("%d,c); } </pre>	<p>b) Lenguaje C</p> <pre> #include <stdio.h> int x,y; main() { x=4; y=5; if (x>y) x=y; else y=4+x; } </pre>	<p>b) Lenguaje C</p> <pre> #include <stdio.h> int i; main() { for (i=1;i<7;i++) y=4+x; } </pre>

Ejemplos de utilización de los procesos básicos de programación:

Realizar alguna fórmula cualquiera se encuentra en un proceso de “**secuencia**”

$x=mx+y$; $y=a+b$; $v=ri$; $v=4\pi R$ etc.

Pero si esa fórmula o ecuación paramétricas, tiene algunas restricciones entonces estamos entrando a la parte de proceso de “**selección**”. Porque buscamos “solo obtener algunos resultados”. Porque se requiere verificar solo para ciertos valores y para otros no. Porque no se harán ciertos procesos si los valores están en un rango determinado; etc.

El proceso de **repetición** entra directamente si cualquiera de lo antes mencionado nos los piden obtener “para varios valores con la misma ecuación”, nos pueden pedir repetir (ó iterar) varios procesos; etc.

NOTA IMPORTANTE: EL PROBLEMA A REALIZAR EN ALGORITMO YA DEBE CONOCERSE COMO SE RESUELVE A MANO Y CON CALCULADORA; YA DEBE SABERSE CUALES SON LAS RESTRICCIONES.

Problema ejemplo1:

Realiza un algoritmo y un programa que saque el promedio de tres calificaciones.

1. Análisis del problema.

¿Datos de entrada? ¿Datos de salida? ¿Restricciones?

¿Proceso secuencia, selección y/ò Repetición?

Me imagino que ya hice mi algoritmo y lo programe: Me pide datos; cal1, cal2, cal3;

Luego calculo el promedio que es la suma de los tres valores y lo divido entre tres; este resultado es mi promedio el cual mando a imprimir.

Con respecto a los **valores de entrada**, las calificaciones pueden ser 10, 8, 7, 5 que son enteros pero también me pueden poner 7.5, 8.3, etc. Por otra parte podrían ponerme 7.55, quizá pero es exagerado tener una calificación 7.558888. Por lo tanto, la generalización para mi será trabajar en números **reales** (para trabajar con números decimales) o sea mis tipos de datos con “float”.

Con respecto a los **valores de salida**: Para esto debo pensar ¿cómo operan mis datos de entrada con qué y que obtendré? Conozco que en la operación de suma las característica de los datos continúan igual, si entran enteros y sumo me salen enteros, si sumo reales obtengo reales, si divido enteros NO me dan enteros me dan reales y en este caso se trabajaran con reales, los datos de salida.

Por lo tanto, sumo tres valores ENTEROS y obtengo un valor ENTERO, luego este lo divido entre un numero ENTERO(3), obteniendo un valor REAL. Conclusion: datos de salida: **Reales**.

¿Restricciones? No nos piden, ¿pondremos algunas para nosotros?, no nos piden sacarlo para varias personas si fuera el caso repetiríamos varias veces.

2. Construcción del algoritmo. (corre el programa)

<p>Grafico: Diagrama de Flujo</p> <pre> graph TD inicio([inicio]) --> input[/cal1, cal2, cal3/] input --> process[$prom = \frac{cal1+cal2+cal3}{3}$] process --> output[/prom/] output --> fin([fin]) </pre>	<p>No Grafico : Seudocódigo</p> <ol style="list-style-type: none"> 1. Leer tres Calificaciones del usuario. 2. Sumarlas y dividir las entre tres. 3. Imprimir resultado <pre> //Lenguaje C #include <stdio.h> float cal1, cal2, cal3, prom; main() { cal1=5; cal2=7; cal3=8; prom=(cal1+cal2+cal3)/3; printf("promedio es %f",prom); getchar(); } </pre>
---	---

3. Verificación de algoritmo. PRUEBA DE ESCRITORIO. Una forma: "Dictale a" SUPONDREMOS tres cosas: Dibujar el registro de memoria (pizarron), tener un "ente limitado de capacidad" y una calculadora simple. (+ pantalla de salida)

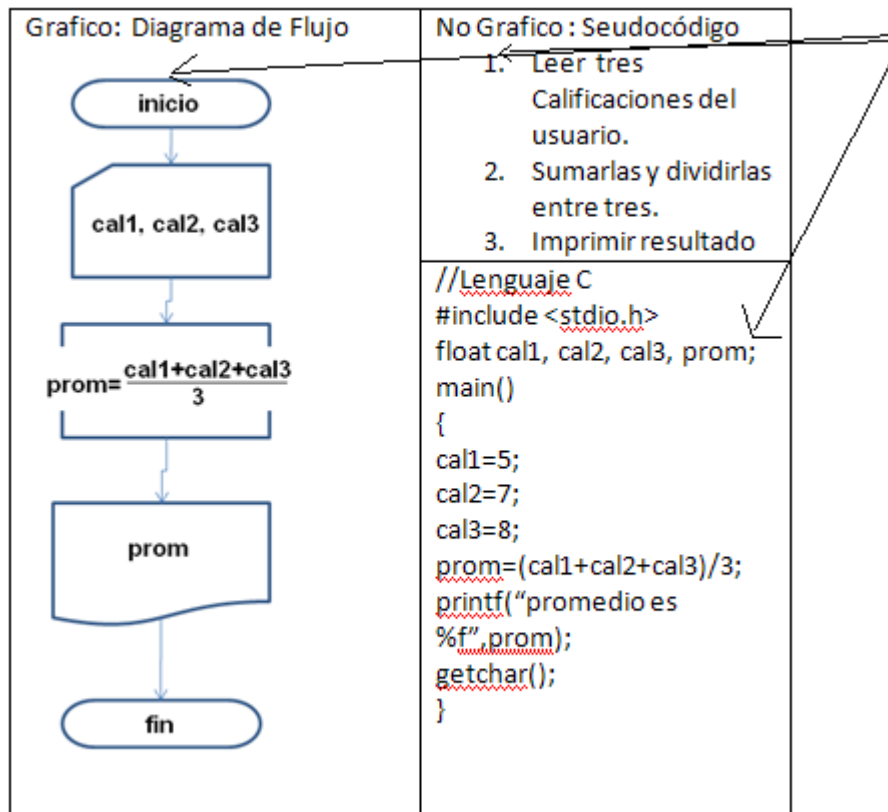


CAL1	CAL2	CAL3	
	PROM		

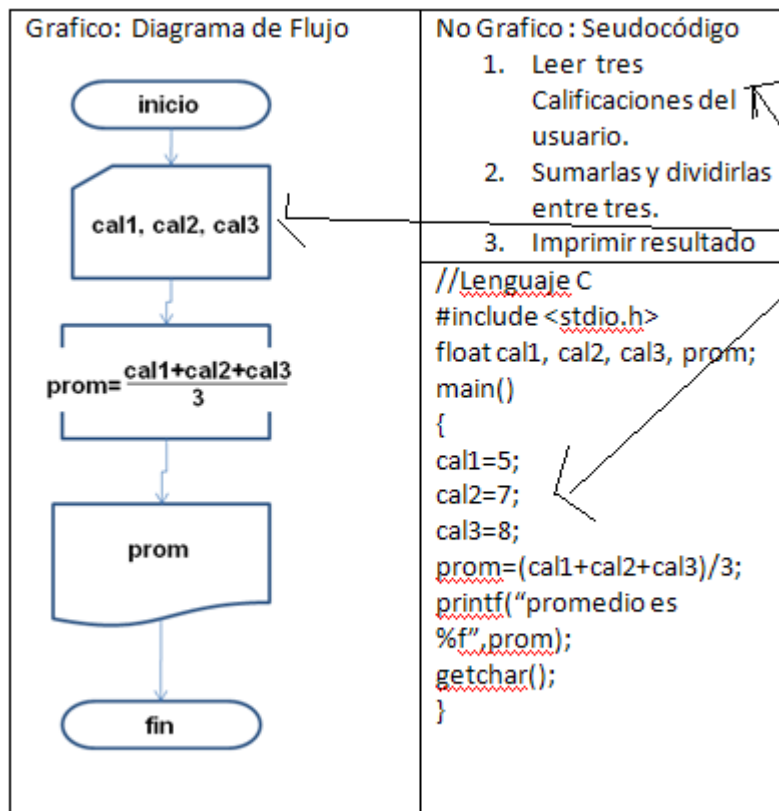


Para este caso pediremos a Robot que vaya realizando tu algoritmo.

1. Bautizar cuatro registros de memoria. Etiquétalos con cal1, cal2, cal3 y prom



2. Iniciamos en el diagrama de flujo.
3. Pedimos datos al usuario o directamente escribimos los datos.



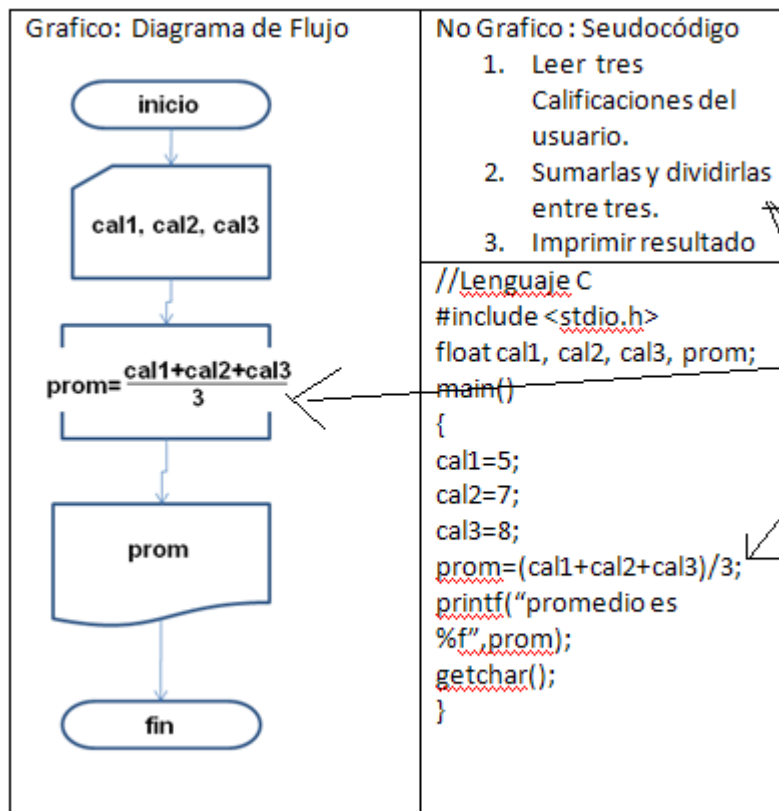
Entonces le decimos a robot que escriba los datos. Robot lo hace.



CAL1	CAL2	CAL3	
5	7	8	
	PROM		



4. Ahora checando nuestro algoritmo, le pedimos a robot que con su calculadora sume los datos que están de cal1, cal2 y cal3 y después divida entre 3, guardándolo en el registro prom.

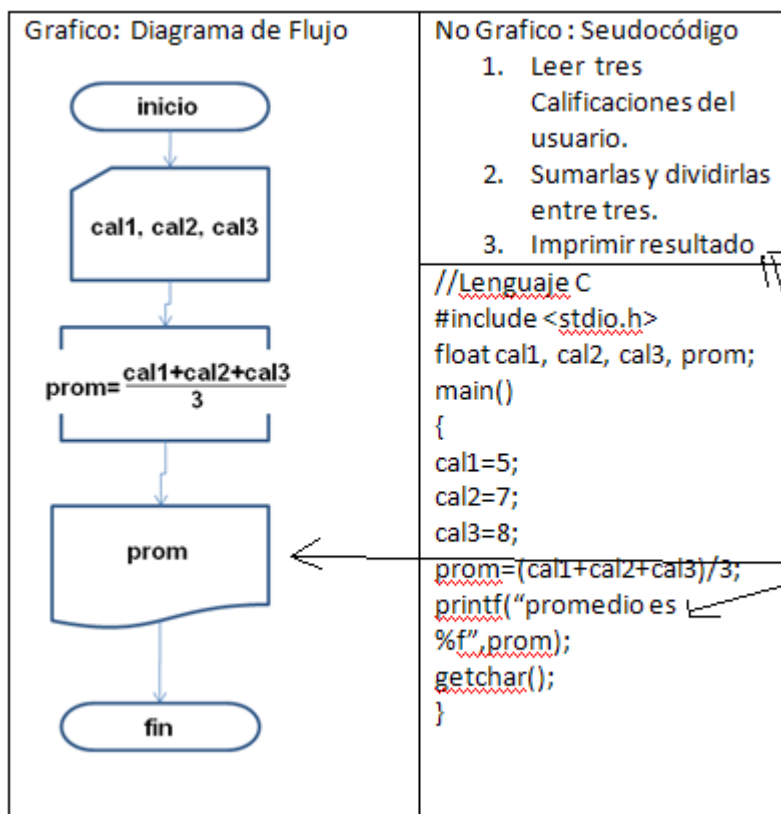




CAL1	CAL2	CAL3	
5	7	8	
	PROM		
	6.67		



5. Ahora le decimos que imprima en pantalla el valor que se encuentra en el casillero PROM.



6. Identifica y realiza los procesos básicos en algoritmo y programado en lenguaje C:
- Suma dos números enteros
 - Resta dos números enteros
 - Compara cual de los dos números es mayor

- d) Compara tres números uno de ellos es 5, verifica que el resto es diferente sino imprime "hay alguno igual"
- e) Imprime números del 1 al 7
- f) Imprime números del 5 al cero.