

**ALGORITMOS
ACTIVIDAD No 2**

**AGUIRRE JUAN MANUEL
CÁRDENAS MOYA JOSÉ GABRIEL**

**UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA UNAD
CEAD ACACIAS
TERCER SEMESTRE
INGENIERÍA DE SISTEMAS
SEPTIEMBRE 15 DE 2007
ACACIAS META**

**ALGORITMOS
ACTIVIDAD No 2**

**AGUIRRE JUAN MANUEL
CÁRDENAS MOYA JOSÉ GABRIEL**

**TUTOR
ING. DIANA MARCELA CARDONA ROMÁN**

**UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA UNAD
CEAD ACACIAS
TERCER SEMESTRE
INGENIERÍA DE SISTEMAS
SEPTIEMBRE 15 DE 2007
ACACIAS META**

INTRODUCCIÓN

Mediante el establecimiento de pasos para el análisis y comprensión de problemas generales algorítmicos se brindan orientaciones a fin de interiorizar la importancia de la selección de estrategias óptimas y viables para su resolución.

Llegado a este punto se debe tener muy clara la fundamentación de la ingeniería de sistemas, con el fin de argumentar aspectos de interés propios de la misma.

OBJETIVOS

OBJETIVOS GENERALES

Obtener un detallado conocimiento sobre lo que es un Algoritmo. Sus partes, como nos ayuda este para la resolución de un problema, aprender a desarrollar un pseudocódigo e identificar sus partes

OBJETIVOS ESPECÍFICOS

- Analizar un problema
- Conocer que es un algoritmo
- Identificar las partes de un algoritmo
- Identificar las variables y constantes de un problema
- Organizar diagramas de flujo
- Identificar los procesos y toma de decisiones de un diagrama de flujo
- Desarrollar un pseudocódigo
- Identificar las estructuras de selección

DESARROLLO ACTIVIDAD 2

Realizar la lectura de la segunda unidad del modulo de algoritmos y de acuerdo a ello resuelva y entregue en un trabajo escrito en Word, con portada, introducción, objetivos, desarrollo, conclusiones, bibliografía, anexos (fichas dadas en la actividad anterior):

1. Realice la lectura No 3 Técnicas de Programación del modulo y elabore un ensayo de Mínimo 2 hojas y Máximo 5 hojas.

Un elemento básico en el funcionamiento del computador es el software. Este es constituido por una serie de instrucciones y datos, que permiten el aprovechamiento de los diversos recursos con que cuenta el computador, para resolver gran cantidad de problemas. Un computador en, si no es mas que un conglomerado de componentes electrónicos; el software le da vida al computador, haciendo que sus componentes funcionen de forma sincronizada.

El software esta constituido por programas; los cuales son un grupo de instrucciones o comandos destinados a cumplir una tarea en particular, un computador es incapaz de hacer cosas por su propia iniciativa, hace únicamente lo que se le ordene. Es necesario especificarle en detalle, paso a paso, de que manera debe ejecutar un conjunto de órdenes. Este proceso de "enseñar" al computador comúnmente se conoce como programación; y su producto es el programa, en el que interviene, el diseñador y programador.

Los lenguajes de programación es el medio por el cual el programador se comunica con el computador. Al igual que los lenguajes que utilizan los seres humanos comúnmente para comunicarse entre ellos, están constituidos por un conjunto de palabras (vocabulario), elaboradas usando un alfabeto particular. Las palabras son combinadas, usando ciertas reglas, para formar proposiciones, bloques de instrucciones y programas en forma similar a las frases y oraciones.

En principio, el único lenguaje que el computador es capaz de procesar, es uno que tiene un alfabeto de solamente dos letras: 0 y 1(binarios). Estas se combinan de diferentes formas, generalmente en grupo de ocho, para formar las palabras del lenguaje. Este lenguaje de ceros y unos recibe el nombre de lenguaje de maquina.

Resulta obvio que hacer un programa usando el lenguaje de maquina puede convertirse en una labor inmensa, hasta para el programador experto. Debido a esto, se han desarrollado varios lenguajes de programación, que simplifican el proceso de programación. Estos lenguajes cuentan con alfabeto más grande y un léxico mas delicado.

Los programas que son realizados usando un lenguaje de programación diferente a el lenguaje de maquina, deben ser traducidos, en ultima instancia, al lenguaje de maquina, por ser Este el único lenguaje que es capaz de procesar el computador. Existen unos programas especiales que hacen esta labor de traducción, y que reciben el nombre de traductores. Cada lenguaje de programación posee su propio traductor.

En cada lenguaje de programación, siempre existe un compromiso entre su poder de expresión y su dificultad de traducción. El poder de expresión de un lenguaje se mide por las

facilidades que ofrece para expresar alguna orden; mientras el lenguaje se parezca cada vez mas al usado comúnmente por los seres humanos, el poder de expresión del lenguaje de programación será mayor. Sin embargo, mientras mayor sea el poder de expresión de un lenguaje, mayor será la dificultad para traducirlo a l lenguaje e maquina.

En la actualidad la mayoría de lenguajes son de alto nivel. Los e uso mas difundido están Basic, pascal, fortran, cobol, lisp, c, sin embargo, a veces es necesario hacer ciertos programas especiales que requieren uso del lenguaje de maquina de forma directa. Para esto, el lenguaje de bajo nivel mas usado es assembler o lenguaje ensamblador.

En la elaboración de un programa, al igual que en cual tarea de la vida cotidiana, requiere de un proceso de investigación, planeación, análisis, diseño, desarrollo y ejecución. Sobre todo si el problema a resolver es complejo. Uno de los productos mas importantes de la fase de planificación, es el algoritmo. Un algoritmo describe de una manera detallada, los pasos que e van a seguir para resolver el problema de manera correcta. Disponiendo de un algoritmo, se facilita la elaboración de un programa, en cualquier lenguaje; el algoritmo para resolver un problema no depende del lenguaje de programación que se utilice.





























2. Investigue cuales son las características del diagrama de flujo

El diagrama de flujo tiene las siguientes características y ventajas:

- Es una representación gráfica de las secuencias de un proceso, presenta información clara, ordenada y concisa.
- Permite visualizar las frecuencias y relaciones entre las etapas indicadas.
- Se pueden detectar problemas, desconexiones, pasos de escaso valor añadido etc.
- Compara y contrasta el flujo actual del proceso contra el flujo ideal, para identificar oportunidades de mejora.
- Identifica los lugares y posiciones donde los datos adicionales pueden ser recopilados e investigados.
- Ayuda a entender el proceso completo.
- Permite comprender de forma rápida y amena los procesos

3. Averiguar el significado de cada símbolo de un diagrama de flujo (Ver Imagen) , Investigar otros símbolos

SOLUCIÓN

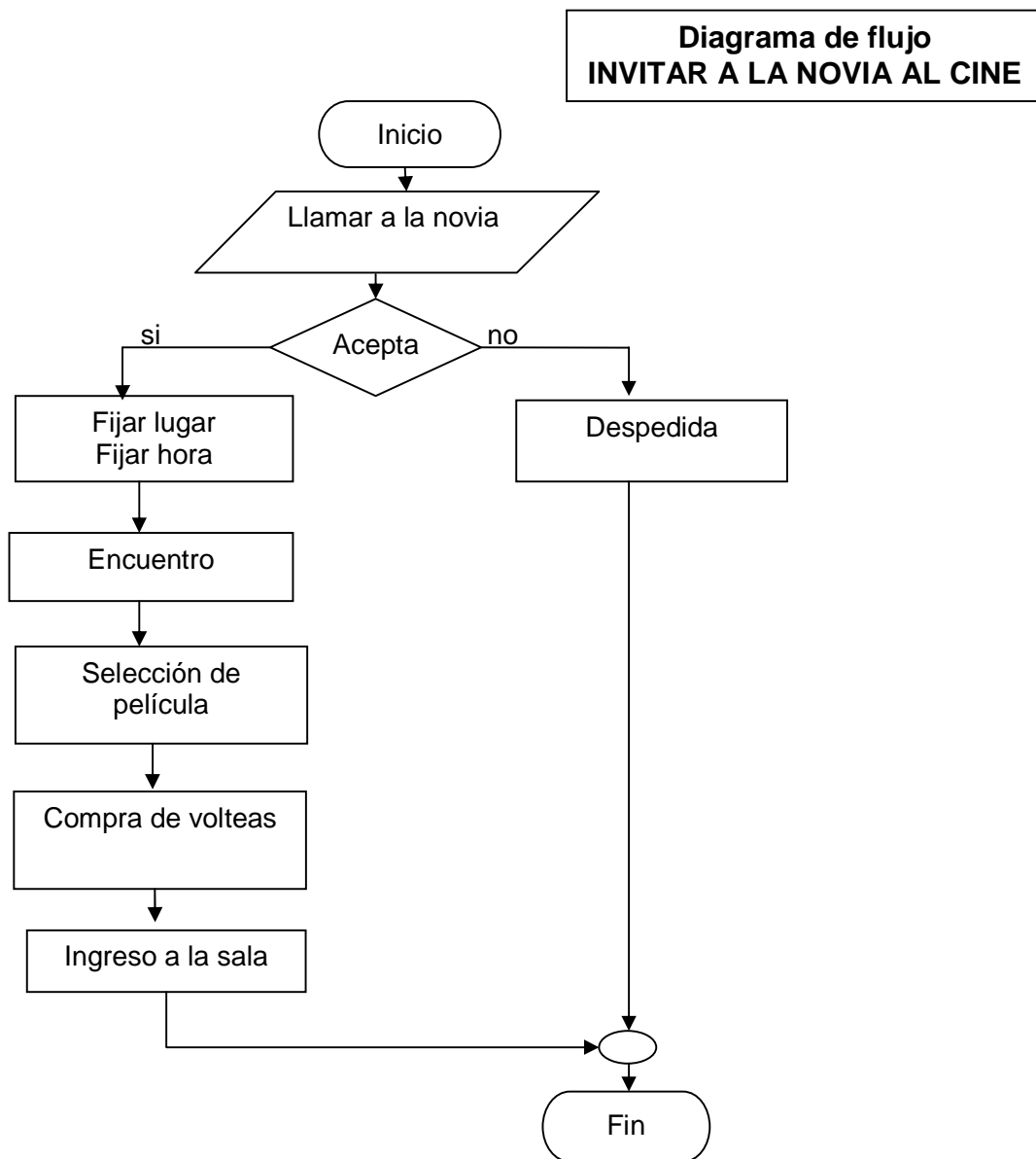
	Proceso		Cinta perforada
	Proceso alternativo		Y
	Decisión		O
	Datos		Intercalar
	Proceso predefinido		Ordenar
	Almacenamiento interno		Extracto
	Documento (Imprimir)		Combinar
	Multidocumento		Datos almacenados
	Inicio – Fin.		Retraso
	Preparación		Almacenamiento de acceso secuencial
	Entrada manual		Disco magnético
	Operación manual		Almacenamiento de acceso directo
	Conector		Pantalla
	Conector fuera de página		
	Tarjeta		

4. Realizar el análisis, diagrama de flujo y prueba de escritorio, para los siguientes planteamientos

1.- Determinar en una lista los pasos para ir al cine con el novio o la novia.

Análisis:

- Problema: invitar a la novia a cine.
- Especificaciones de entrada: llamar a la novia, hacer invitación. Especificaciones de salida: fijar lugar, fijar hora, encuentro en el lugar fijado, compra de boletas, ingreso a la sala de cine.

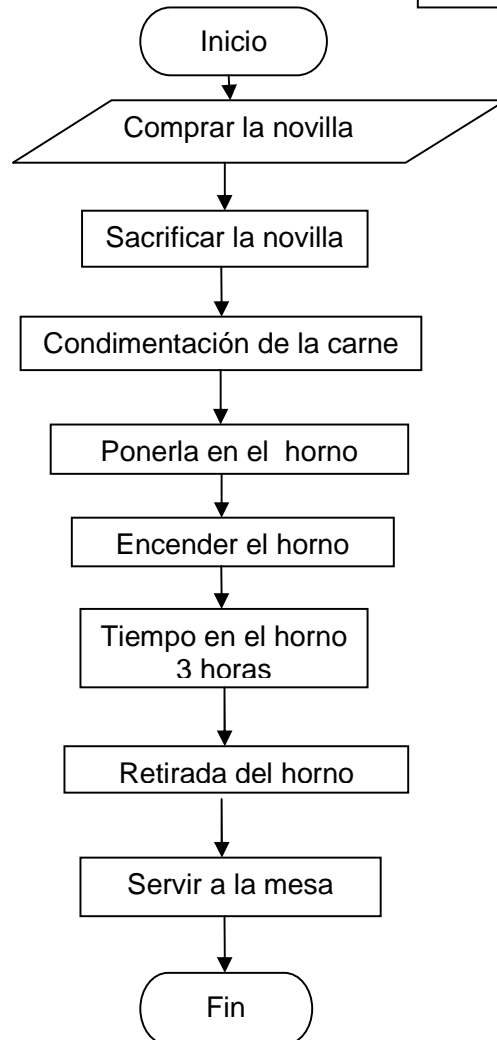


2.- Realizar un algoritmo donde se indique los pasos para realizar un plato típico de la región.

Análisis:

- Problema: elaboración de plato típico “mamona”
- Especificaciones de entrada: comprar la novilla.
- Especificaciones de salida: sacrificio de la novilla, condimentos, yuca, papa, sal, trozos de leña.

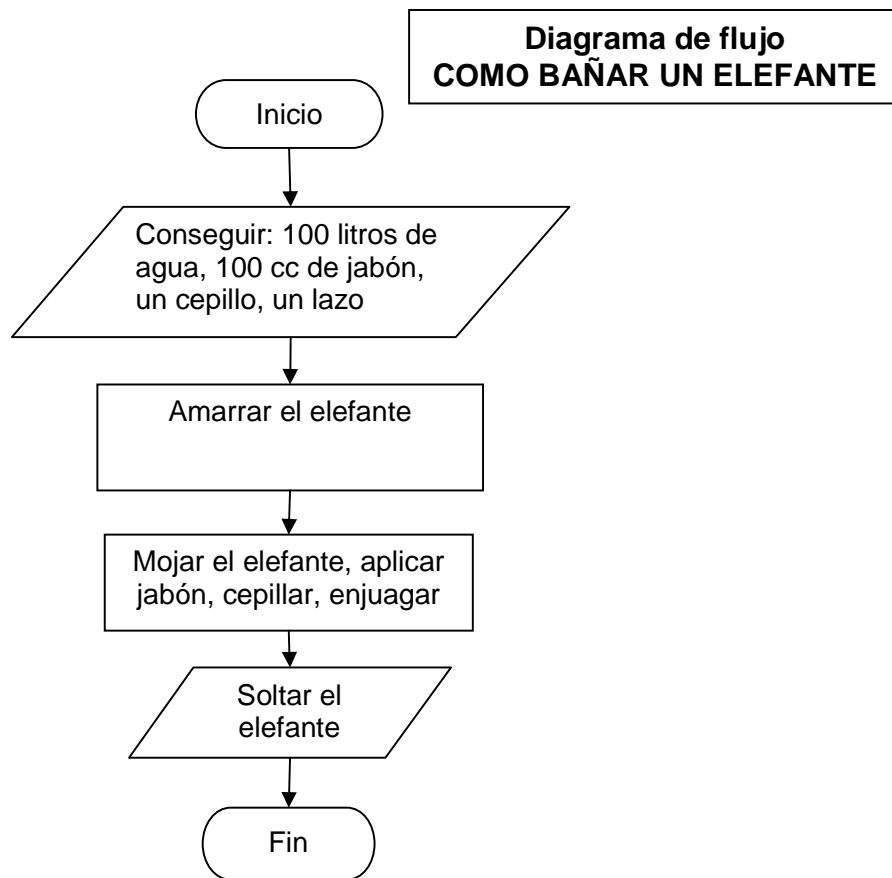
**Diagrama de flujo
REALIZAR “LA MAMONA”**



3.- Realizar un diagrama que permita esquematizar (dibujo) los pasos para bañar un elefante.

Problema: como bañar un elefante

- Especificaciones de entrada: un elefante, un lazo, 100 litros de agua, 100 cc de jabón líquido, un cepillo, un balde
- Especificaciones de salida: mojar el elefante, aplicar jabón, cepillar, enjuagar.



4.- Determinar cual es el cuadrado, de un determinado número, primero haga la lista de pasos a utilizar, luego el algoritmo en pseudocódigo y finalmente el diagrama de flujo.

Pasos:

Inicio

Ingrese un número: a

Lea numero: a

Cuadrado = a^2 ;

Escriba el resultado

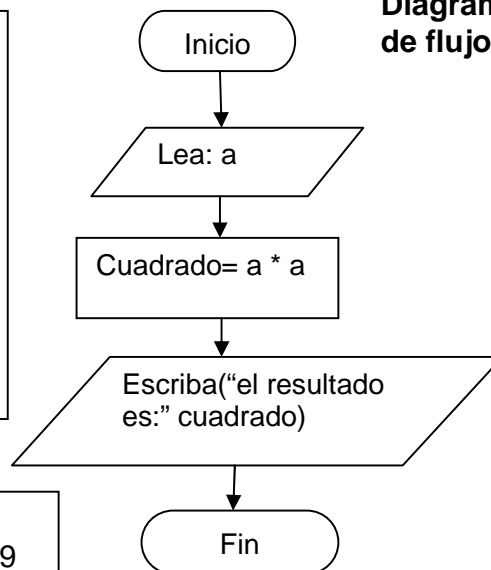
Fin

Pseudocódigo

1. Algoritmo el cuadrado de un número
2. Var
3. a: entero;
4. b
5. Inicio
6. Escriba("ingrese el numero");
7. Lea(a);
8. Cuadrado= a * a;
9. Escriba("el resultado es:",cuadrado);
10. fin

Prueba de escritorio

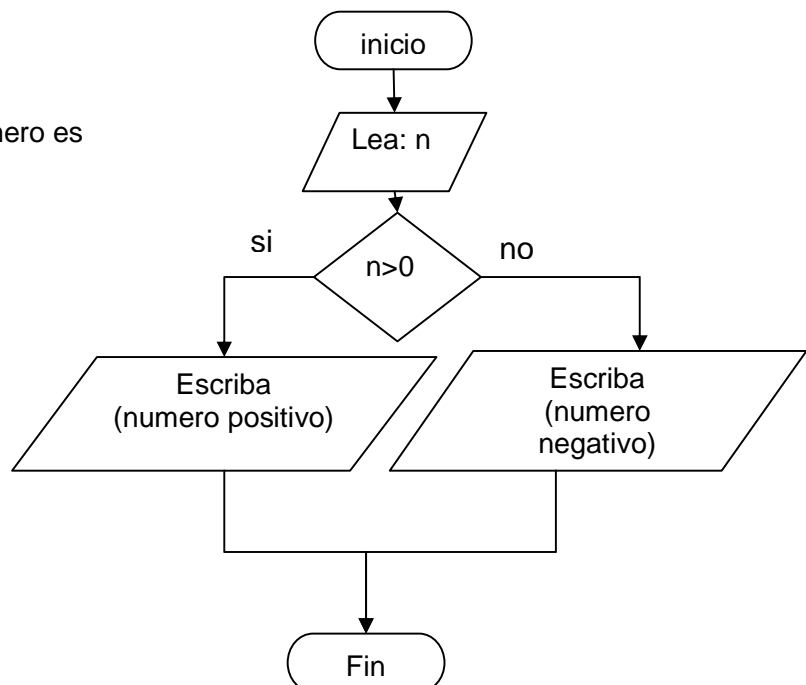
Cuadrado= a * a = 3 * 3 = 9
a "es el numero" 9

Diagrama de flujo

5.- Elaborar un diagrama de flujo que permita determinar si un número es Positivo o Negativo.

Pseudocódigo

1. algoritmo si un numero es positivo o negativo
2. var
3. n: entero;
4. inicio
5. escriba("numero");
6. lea (n);
7. si n>0
8. escriba ("el numero es:",positivo);
9. sin_no
10. escriba ("el numero es:",negativo);
11. fin_si
12. fin

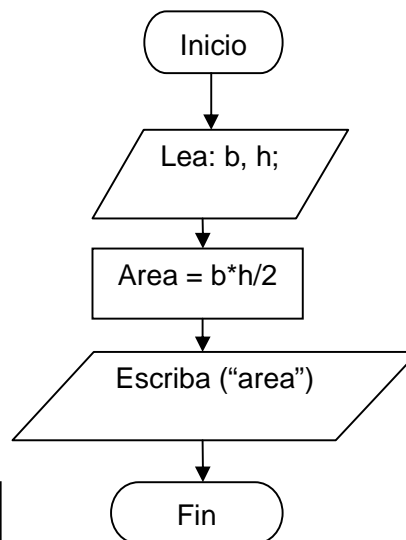


6.- Elabore en pseudocódigo y en diagrama de flujo un algoritmo que halle el área de un triángulo.

Pseudocódigo

1. algoritmo area de un triángulo
2. var
3. b,h: entero;
4. inicio
5. escriba("un numero");
6. lea(b);
7. escriba("un numero");
8. lea(h);
9. $area = b * h / 2$
10. escriba("el valor es:",area);
11. fin

Diagrama de flujo



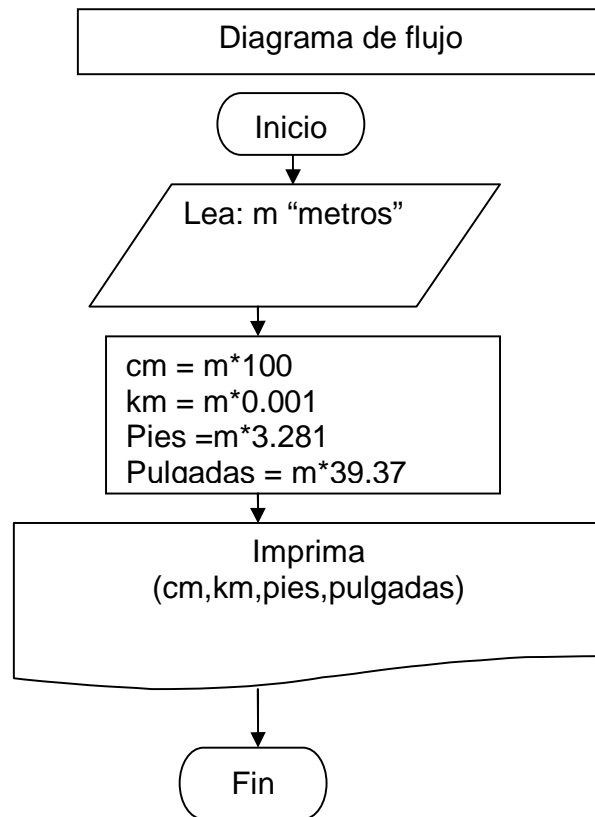
Prueba de escritorio

b	h	area
10	8	$(10 * 8) / 2 = 40$

7.- Ejercicio de conversión, desarrollar un algoritmo en pseudocódigo y luego un diagrama, que permita ingresar una cantidad en metros y la convierta a Centímetros, Kilómetros, Pies, pulgadas (utilice las sentencias de selección).

PSEUDOCODIGO

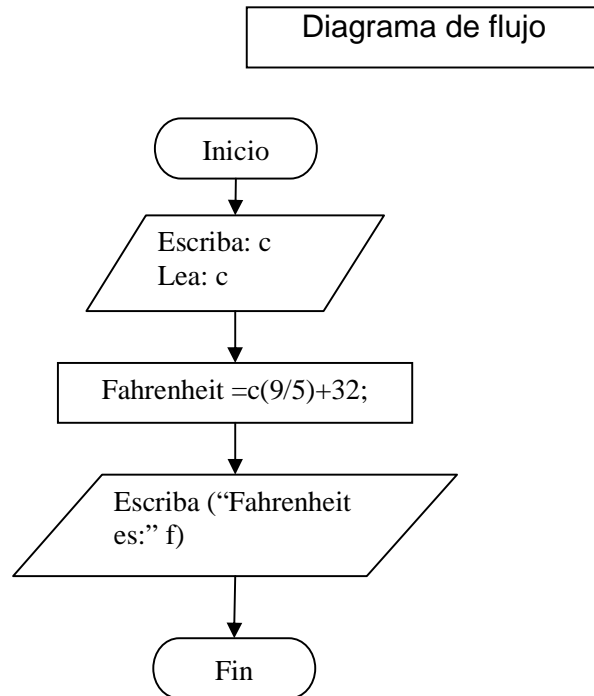
1. algoritmo de conversión de metros a: centímetros, kilometros, pies, pulgadas;
2. var
3. m,km,cm,pies,pulgadas:real
4. inicio
5. escriba("por favor ingrese un numero");
6. lea(m);
7. $cm = m * 100$;
8. escriba("el valor es:",cm);
9. $km = m * 0.001$;
10. escriba("el valor es:",km);
11. $pies = m * 3.281$;
12. escriba("el vl原因 es:",pies);
13. $pulgadas = m * 39.37$;
14. escriba("el valor es:",pulgadas);
15. imprimir(c,km,pies,pulgadas)
16. fin



8.- Leer una determinada temperatura en grados centígrados y convertirla a Fahrenheit

PSEUDOCODIGO

1. leer una temperatura en grado centigrados y convertirla en grados fahreheit.
2. var
3. c = ("grados centigrados"); real
4. inicio
5. ingrese("el valor en grados");
6. lea(c);
7. Fahrenheit = $c(9/5)+32$;
8. escriba("grados Fahrenheit es:",f);
9. fin

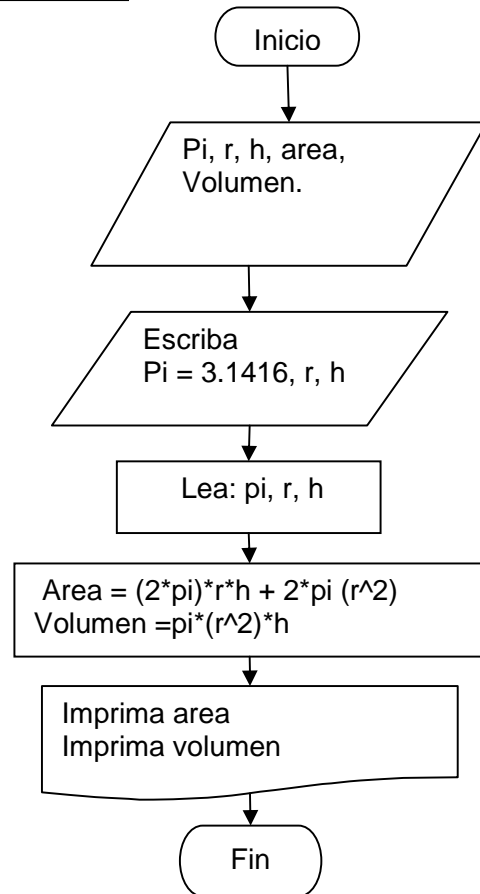


9.- Elabore un algoritmo en pseudocódigo y un diagrama de flujo que teniendo como datos de entrada el radio y la altura de un cilindro calcule el área total y el volumen del cilindro.

Diagrama de flujo

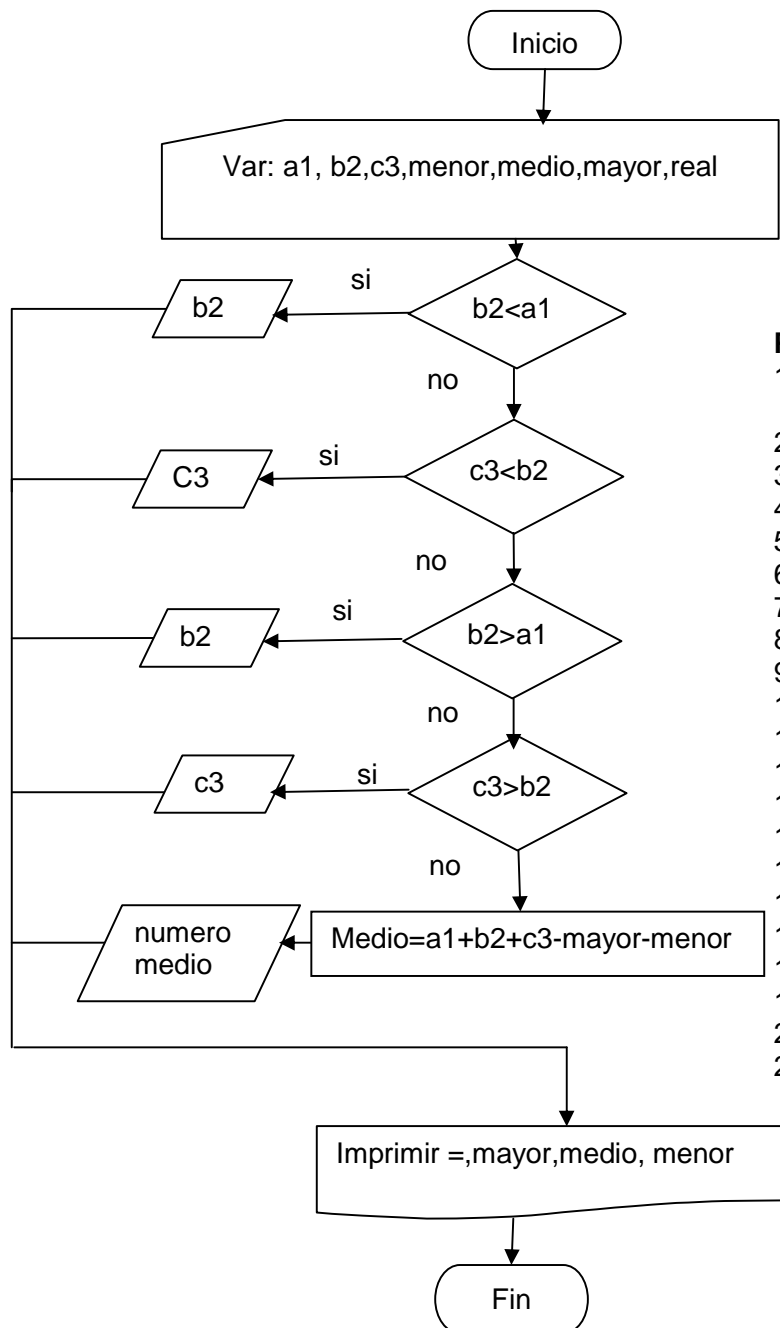
Pseudocódigo

1. area y volumen e un cilindro
2. var
3. pi,r,h,area,volumen: flotante y enetro
4. pi=3.1416,r,h;
5. inicio
6. escriba("el radio");
7. lea(r);
8. escriba("la altura");
9. lea(h);
10. $area = (2 \cdot \pi) \cdot r \cdot h + (2 \cdot \pi) \cdot r^2$;
11. escribir("el resultado es:",area);
12. $volumen = \pi \cdot r^2 \cdot h$;
13. escribir("el resultado es:",volumen);
14. fin



10.- Elaborar un diagrama de flujo, que permita ingresar 3 valores y los imprima en forma descendente.

Diagrama de flujo



Pseudocódigo

1. algoritmo que imprima 3 valores descendente
2. var
3. a1, b2, c3, menor, medio, mayor: real
4. inicio
5. escriba(a1);
6. lea("numero", a1);
7. escriba(b2);
8. lea("numero", b2);
9. escriba(c);
10. lea("numero", c3);
11. si(b2 < a1)
12. escriba("numero menor es:", b2);
13. si(c3 < b2)
14. escriba("numero menor es:", c3);
15. si(b2 > a1)
16. escriba("numero mayor es:", b2);
17. si(c3 > b2)
18. escriba("numero mayor es:", c3);
19. medio = a1 + b2 + c3 - mayor - menor;
20. imprimir(menor, medio, mayor)
21. fin

5. Analice el siguiente algoritmo en pseudocódigo y discuta con sus compañeros de grupo, porque en este ejercicio se emplearon 3 condicionales?, Existen otras formas de resolverlo?

```
1. Algoritmo mayor_v1
2. var
3. a,b:entero
4. inicio
5. Escriba("Por Favor entre un número");
6. Lea(a);
7. Escriba("por favor entre el segundo número");
8. Lea (b);
9. Si (a=b)
10. escriba ("los Números son Iguales");
11. fin_si
12. Si (a>b)
13. escriba ("El mayor es :", a);
14. fin_si
15. Si (a<b)
16. escriba("el mayor es :",b);
17. fin_si
18. fin
```

RTA

Se emplean 3 condicionales; por que necesitamos saber cual es el mayor y menor de los 2 números enteros, pero que pasa, si al introducir los números son iguales, posiblemente imprimiría error; entonces empleamos las tres condiciones, para que el algoritmo no se ambiguo.

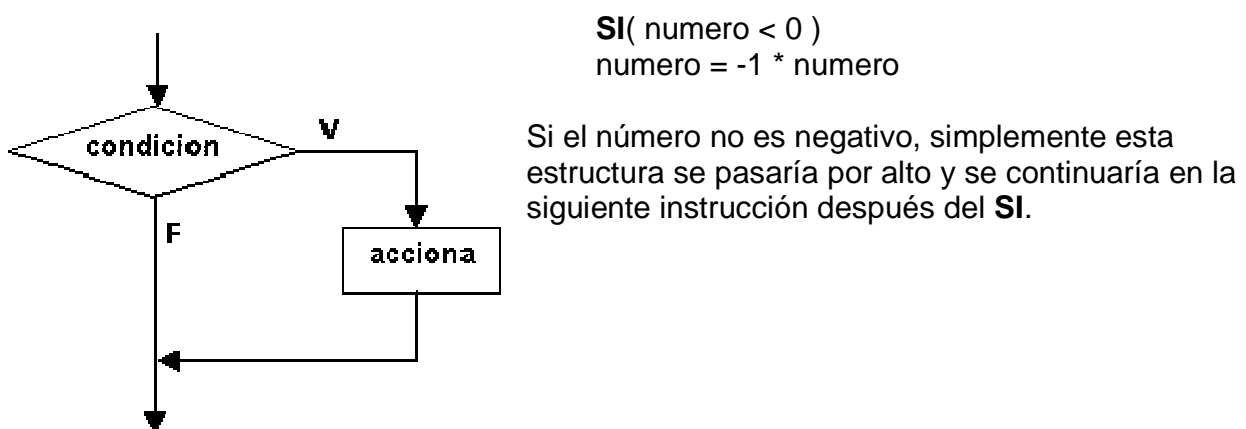
Forma 2

```
1. Algoritmo mayor_v1
2. var
3. a,b:entero
4. inicio
5. Escriba("Por Favor entre un número");
6. Lea(a);
7. Escriba("por favor entre el segundo número");
8. Lea (b);
9. Si (a=b)
10. escriba ("los Números son Iguales");
11. sino_si(a>b)
12. escriba ("El mayor es :", a);
13. sino_si(a<b)
14. escriba("el mayor es :",b);
15. fin_si
16. fin
```

6. Defina estructura de selección simple, doble y en cascada y dé un ejemplo en pseudocódigo.

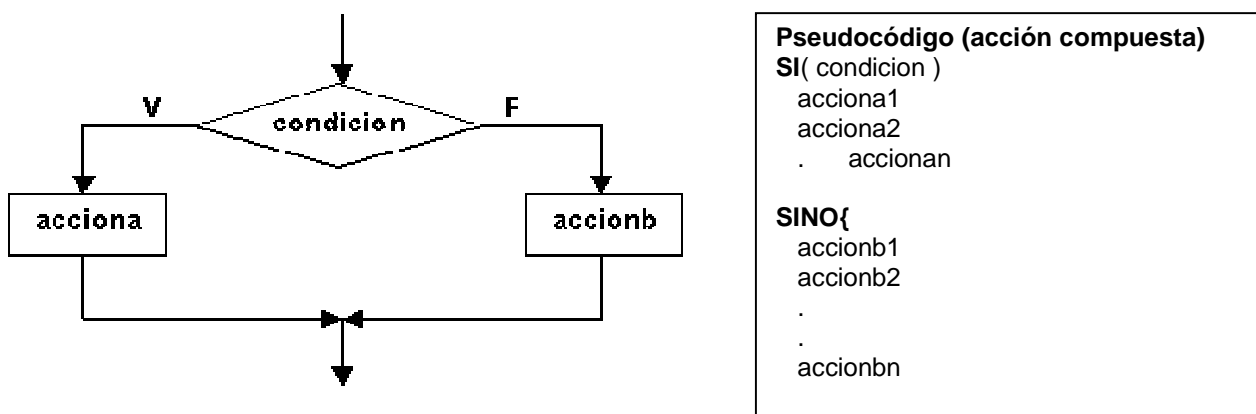
- Selección simple La estructura de selección simple permite ejecutar una acción o un grupo de acciones sólo si se cumple una determinada condición. Así, si la condición es verdadera se ejecuta la acción **acciona**; en caso contrario, no se hace nada

Por ejemplo, si se desea cambiar el signo de un número únicamente en caso que sea negativo, podemos escribir:



➤ **Estructura de Selección Doble SI - SINO**

La estructura de selección doble permite seleccionar una ruta de dos rutas posibles en base a la verdad o falsedad de una condición. Así, en la Figura 3.2, si la condición es verdadera, se ejecuta la acción A; en caso contrario, se ejecuta la acción B.



Por ejemplo, la siguiente estructura de selección doble determina si una persona es mayor o menor de edad:

```

SI( edad >= 18 )
    estado = "Mayor de edad"
SINO
    estado = "Menor de edad"

```

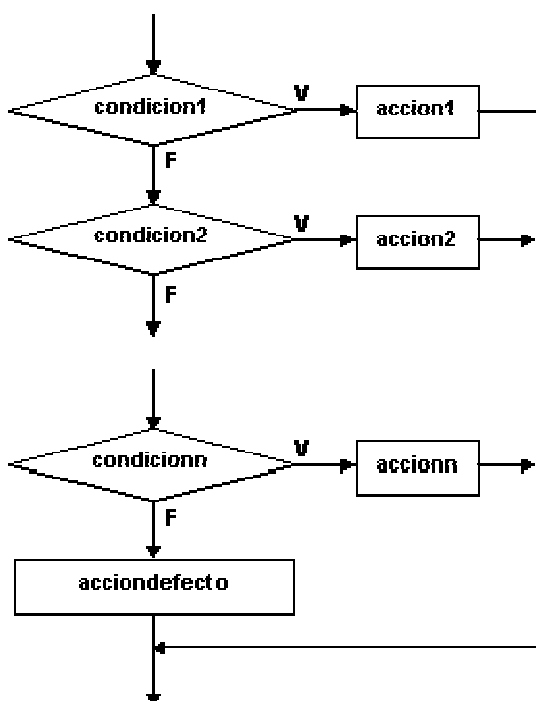
Esto imprime "Mayor de edad" si la persona tiene 18 años ó más e imprime "Menor de edad" si la persona tiene menos de 18 años. En cualquiera de los casos, después de efectuar la impresión, se ejecutará la primera instrucción que sigue a la estructura **SI...SINO**.

➤ Estructura de Selección Doble en Cascada SI-SINO-SI

La estructura de selección doble en cascada esta formada por varias estructuras de selección doble SI-SINO puestas una a continuación de otra de forma que a un SI-SINO le sigue otro SI-SINO.

En la estructura de selección doble en cascada, las condiciones se evalúan en orden descendente, pasando de una condición a otra si la condición anterior resulta falsa. En el momento que se encuentra una condición verdadera, se efectúa la acción correspondiente a dicha condición y se corta el resto de la estructura. Si todas las condiciones resultan falsas, se efectúa la acción correspondiente al último SINO, que se considera como la acción por defecto.

➤ Diagrama de flujo y el pseudocódigo de la estructura de selección doble en cascada.



Pseudocódigo (Forma 1)

```

SI(      condicion1      )
    accion1
SINO
    SI(      condicion2      )
        accion2
    SINO
        SI(      condicion3      )
            accion3
        .
        .
        .
    SINO
        acciondefecto

```

Pseudocódigo (Forma 2)

```

SI( condicion1 )
    accion1
SINO SI( condicion2 )
    accion2
SINO SI( condicion3 )
    accion3
.
.
.
SINO
    acciondefecto

```

Ejemplos de selección simple:

Ejemplo 1:

Por ejemplo, si se desea cambiar el signo de un número únicamente en caso que sea negativo, podemos escribir:

1. cambio de signo a un numero si es negativo
2. var
3. numero:entero,real
4. inicio
5. introduzca("el numero");
6. lea(n);
7. si($n < 0$)
8. numero= $-1 * n$;
9. fin

Ejemplo 2:

Realizar un pseudocódigo, para saber si pase un determinado curso.

Pseudocodigo

1. Algoritmo para conocer el estado académico
2. var
3. nota: entero, real;
4. inicio
5. ingrese("por favor la nota");
6. lea(nota);
7. si($\text{nota} \geq 3.0$)
8. entonces("paso el semestre");
9. fin

Ejemplos de selección doble:

EJEMPLO 1

Hallar el salario de los empleados de ventas de la empresa comercial andina: teniendo los siguientes datos, sueldo básico 500.000, según las ventas de cada empleado se le da una comisión, así: menor de 2.000.000 5%, mayor 10%.

PSEUDOCODIGO

1. algoritmo para calcular el salario de un empleado
2. var
3. s,ventas,sueldo basico,0.05,0.1:entero,real
4. inicio
5. introduzca("valorventas");
6. lea(valor)
7. si($\text{ventas} < 2000000$)
8. $s = (\text{ventas} * 0.05) + \text{sueldobasico}$;
9. sino($\text{ventas} * 0.1$)+sueldobasico;
- 10.imprima ("salarios de empleados")
- 11.fin

Ejemplos de selección doble en cascada: SI-SINO-SI.

Diseñe un algoritmo que califique el puntaje obtenido en el lanzamiento de tres dados en base a la cantidad seis obtenidos, de acuerdo a lo siguiente: tres seis, excelente; dos seis, muy bien; un seis, regular; ningún seis, pésimo.

PSEUDOCODIGO

calificacion del puntaje según el lanzamiento

VAR

ENTERO: dado1, dado2, dado3

CADENA: calificacion

inicio

LEER(dado1,dado2,dado3)

SI(dado1+dado2+dado3==18)

calificacion="Excelente"

SINO

SI(dado1+dado2==12||dado1+dado3==12||dado2+dado3==12)

calificacion="Muybien"

SINO

SI(dado1==6||dado2==6||dado3==6);

calificacion="Regular"

SINO

calificacion="Pésimo"

IMPRIMIR(calificacion)

FIN

CONCLUSIONES

- Estudiada la segunda unidad, aprendimos analizar un problema, conocer sus variables, sus constantes y estructurarlo en diagrama de flujo, y realizar un pseudocódigo.
- Cuando necesitamos resolver un problema, lo primero en lo que debemos pensar es en analizarlo y entenderlo de forma clara y suficiente. En esto radica el éxito de la solución obtenida (optimización de recursos para llegar a la solución).

BIBLIOGRAFIA

FUENTES

Criado Pérez, Antonio M. Y Frutos Rayego, Fabián: Introducción a los Fundamentos físicos de la informática, Madrid: Paraninfo, 1999
Gabor Iserincs Enciclopedia de la micro computación

DIRRECCIONES ELECTRONICAS

[http://es.wikipedia.org/wiki/Software libre](http://es.wikipedia.org/wiki/Software_libre)
<http://www.um.es/atika/softla/que-es.php>
<http://www.conozcasuhardware.com>