Unidad dos: Estructura general del programa

1. Concepto de programa
   - 1.1 Codificación y edición del problema
   - 1.2 Partes constitutivas de un programa
   - 1.3 Instrucciones y tipos de programas
     - 1.3.1 Tipos de Instrucciones
     - 1.3.2 Instrucciones de asignación
     - 1.3.3 Instrucción de lectura de datos
     - 1.3.4 Instrucción de escritura de resultados
     - 1.3.5 Instrucción de bifurcación

2. Elementos básicos de un programa
   - 2.1.1 Bucles
   - 2.1.2 Bucles Anidados
   - 2.1.3 Contadores
   - 2.1.4 Acumulador
   - 2.1.5 Estructuras de decisión o selección
   - 2.1.6 Interruptores
3. Programación
   - 3.1 Tipos de programación
   - 3.2 Programación estructurada.
   - 3.3 Estructura secuencial o secuencia de instrucción
   - 3.4 Operaciones o instrucciones de asignación

4. Pasos para la asignación de problemas
   - 4.1.1 Ejemplo

5. Estructura selección de instrucción
   - 5.1 La estructura Si entonces SI/ No
     - 5.1.1 estructura selectiva anidada

1. Concepto de programa.

Un programa de computadora es un conjunto de instrucciones, ordenes dadas a la maquina, que producirán la ejecución de una determinada tarea. También podemos decir que un programa es un medio para conseguir un fin. En conclusión programa es el proceso para solucionar un problema. Programación de computadoras es la ciencia que permite a una persona programar un computador para que resuelva tareas de manera rápida. Un Programa de computador se puede definir como una secuencia de instrucciones que indica las acciones o tareas que han de ejecutarse para dar solución a un problema determinado. La computadora resuelve problemas de acuerdo como se le haya programado de manera rápida.

1.1 Codificación y edición del problema

La codificación consiste en traducir el diagrama de flujo a instrucciones de un lenguaje de computadora. Al conjunto de instrucciones escrita en un lenguaje se llama programa. La edición consiste en introducir el programa al computador, por medio de un editor de texto, generalmente incluido en lenguaje. El algoritmo es un método para resolver un problema mediante una serie de pasos definidos, precisos y finitos. Preciso: implica el orden de realización de cada uno de los pasos. Definido: si se sigue dos veces, se obtiene el mismo resultado. Finito: Tiene un número determinado de pasos, implica que tiene un fin.

El programador diseña un programa, para resolver un problema particular. Diseñar es un proceso creativo y consta de los siguientes pasos o etapas:

Pasos	Etapa	Descripción
1	Análisis del problema	Conducen al diseño detallado por medio un código escrito en forma de un algoritmo
2	Diseño de algoritmo	
3	Codificación	Se implementa el algoritmo en un código escrito en un lenguaje de programación. Refleja las ideas desarrolladas en las etapas de análisis y diseño
4	Compilación y ejecución	Traduce el programa fuente a programa en código de máquina y lo ejecuta.
5	Verificación	Busca errores en las etapas anteriores y los elimina.
6	Depuración	
7	Documentación	Son comentarios, etiquetas de texto, que facilitan la comprensión del programa

Imagen1.etapas del diseño

Para llegar a tener una secuencia de instrucciones que den solución a un problema es necesario ejecutar varias etapas.

- Etapa de análisis: En esta etapa el programador debe entender claramente el problema. Saber que es lo que se quiere resolver. (analizar)
- Etapa de Solución general: Escribir la serie de pasos que sean necesarios para dar solución al problema. Estos pasos se pueden desarrollar a través de un Diagrama de flujo (Utilizando símbolos) ó a través de un seudo lenguaje (Utilizando Lenguaje común). A lo anterior es lo que se conoce con el nombre de Algoritmo.
- Etapa de prueba: Consiste en chequear el algoritmo paso a paso para estar seguro si la solución da solución verdaderamente el problema. (Prueba de escritorio).
- Etapa de implementación específica: Consiste en traducir el algoritmo a un lenguaje de programación. (Codificar).

- Etapa de prueba: Consiste en ejecutar el programa en un computador y revisar los datos arrojados para ver si son correctos y hacer los ajustes necesarios. (Implementar).
- Etapa de uso: Consiste en instalar el programa de manera definitiva para el uso por parte del usuario.

1.2. Partes Constitutivas de un programa.

El programador debe establecer el conjunto de especificaciones que debe contener el programa: Entrada salida y algoritmos de resolución que incluirán las técnicas para obtener las salidas a partir de las entradas. Se debe establecer de donde provienen las entradas al programa es decir los dispositivos de entrada teclado, discos, medios magnéticos. La entrada de datos =operación de lectura de datos o acción de leer. Las salidas de datos se deben presentar en dispositivos de salida como impresora, monitor, dispositivos de almacenamiento. Operación de salida de datos = escritura o acción de escribir

Entrada ---------- programa ---------- salida

1.3 Instrucciones y tipos de programas

El proceso de diseño de algoritmos y posteriormente la codificación del programa, consiste en definir las acciones o instrucciones que resolverá el problema. Las instrucciones son todas las acciones que van ser ejecutadas por el computador par resolver un problema.

1.31 Tipos de Instrucciones

Las instrucciones disponibles en un lenguaje de programación dependen del tipo de lenguaje. La clasificación más usual es:

a) Instrucción de inicio/fin
b) Instrucción de asignación
c) Instrucción de lectura
d) Instrucción de escritura
e) Instrucción de bifurcación
f)

Tipos de instrucción español	pseudocódigo en ingles	pseudocódigo en
Comienzo de proceso	bejín	Inicio
Fin de proceso	end	fin
Entrada	read	leer
Salida	write	escribir
Asignación	A ← 5	B ← 7

Imagen2. Instrucciones básicas

1.3.2 Instrucciones de asignación

La operación de asignación es el modo de darle valores a una variable se representa con el operador, la operación de asignación se conoce como instrucción o sentencia de asignación en lenguaje de programación.

Ejemplos
a) A ← 5
La acción de asignar es destructiva ya que el valor que tuviera la variable antes se perdería.
A ← 21
A ← 15
A ← 5
El valor que quedara es 5 los otros han desaparecido. Las acciones de asignación se clasifican según sea el tipo de expresiones en aritméticas, lógicas y de caracteres.

b) Cual es el valor del variable AUX al ejecutarse la instrucción 5?

1. A ← 10
2. B ← 20
3. AUX ← A
4. A ← B
5. B ← AUX

En la instrucción 1, A toma el valor 10
En la instrucción 2, A toma el valor 20
En la instrucción 3, AUX toma el valor anterior de A, es decir 10
En la instrucción 4, A toma el valor anterior de B, es decir 20
En la instrucción 5, B toma el valor anterior de AUX, es decir 10

Asignación lógica

La expresión que se evalúa es de tipo lógico. Ejemplo

M ← 8>5 Su valor lógico es verdadero.

Asignación de caracteres

La expresión que se evalúa es de tipo carácter

X ← 12 de Octubre de 1942

1.3.3 Instrucciones de lectura de datos

Esta instrucción lee datos de algún dispositivo de entrada

Leer NUMERO, HORAS, TASA

¿Cuál será el significado de las instrucciones siguientes?

Leer del terminal los valores NÚMERO, HORAS, TASA.

Archivándolos en la memoria si los tres números que se teclean en respuesta a la instrucción son:

12325, 32, 1200, significaría que se han asignado a las variables esos valores y equivaldría a la ejecución de las instrucciones.

NUMERO ← 12325
ORAS ← 32
TASA ← 1200

1.3.4 Instrucciones de escritura de resultados

Estas instrucciones se escriben en un dispositivo de salida.

Se asignaron previamente valores a las variables A B C. de manera que:

A ← 100
B ← 200
C ← 300

Se imprimirán o visualizaran en la pantalla los valores 100, 200, 300.

1.3.5. Instrucciones de bifurcación

El desarrollo lineal de un programa se interrumpe cuando se ejecuta una bifurcación pueden ser según el punto del programa a donde se bifurca hacia delante o hacia atrás. La bifurcación puede ser condicional o incondicional:

Incondicional: la bifurcación se da sin necesidad del cumplimiento de ninguna condición.

Condicional: la bifurcación depende del cumplimiento de una determinada condición.

2. Elementos básicos de un programa

- Palabras reservadas (inicio, fin, si, entonces... etc.)
- Identificadores (nombres de variables esencialmente)
- Caracteres especiales (coma apostrofe, etc.)
- Constantes
- Variables
- Expresiones
- Instrucciones
- Bucles
- Contadores
- Acumuladores
- Interruptores
- Estructuras: selectiva, secuenciales, repetitivas

2.1.1 Bucles

Un bucle o lazo (loop) es un segmento de un algoritmo o programa, cuyas instrucciones se repiten un número determinado de veces mientras se cumple una determinada condición, se debe establecer un mecanismo para determinar las tareas repetitivas. Este mecanismo es una condición que puede ser verdadera o falsa y se comprueba una vez a cada paso o interacción del bucle (total de instrucciones que se repiten en el bucle).

Un bucle consta de tres partes:

- decisión
- cuerpo del bucle
- salida del bucle

El bucle a continuación es infinito ya que las instrucciones (10 (2) (3) se ejecutaran indefinidamente, pues no existe salida del bucle al no cumplirse una determinada condición.

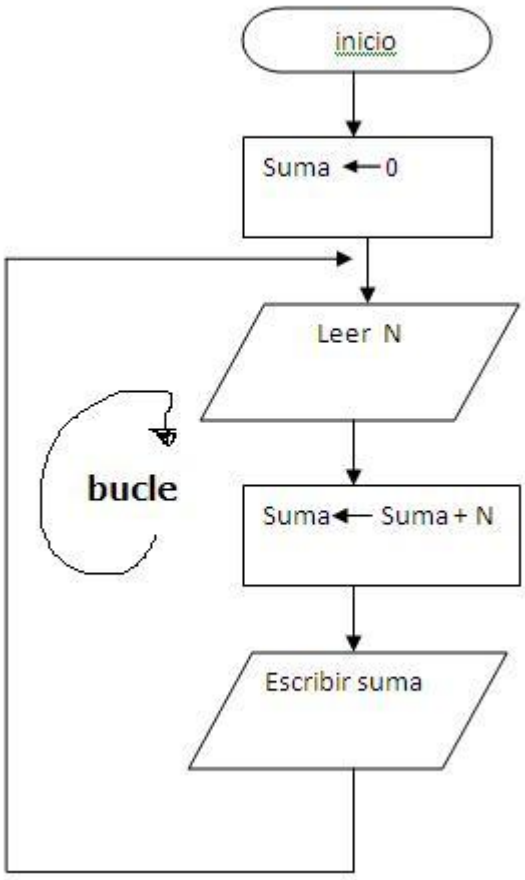


Imagen3. Bucle infinito

2.1.2 Bucles anidados

En un algoritmo puede haber varios bucles. Estos pueden ser anidados o independientes. Bucles anidados: cuando están dispuestos de tal modo que unos son interiores a otros.

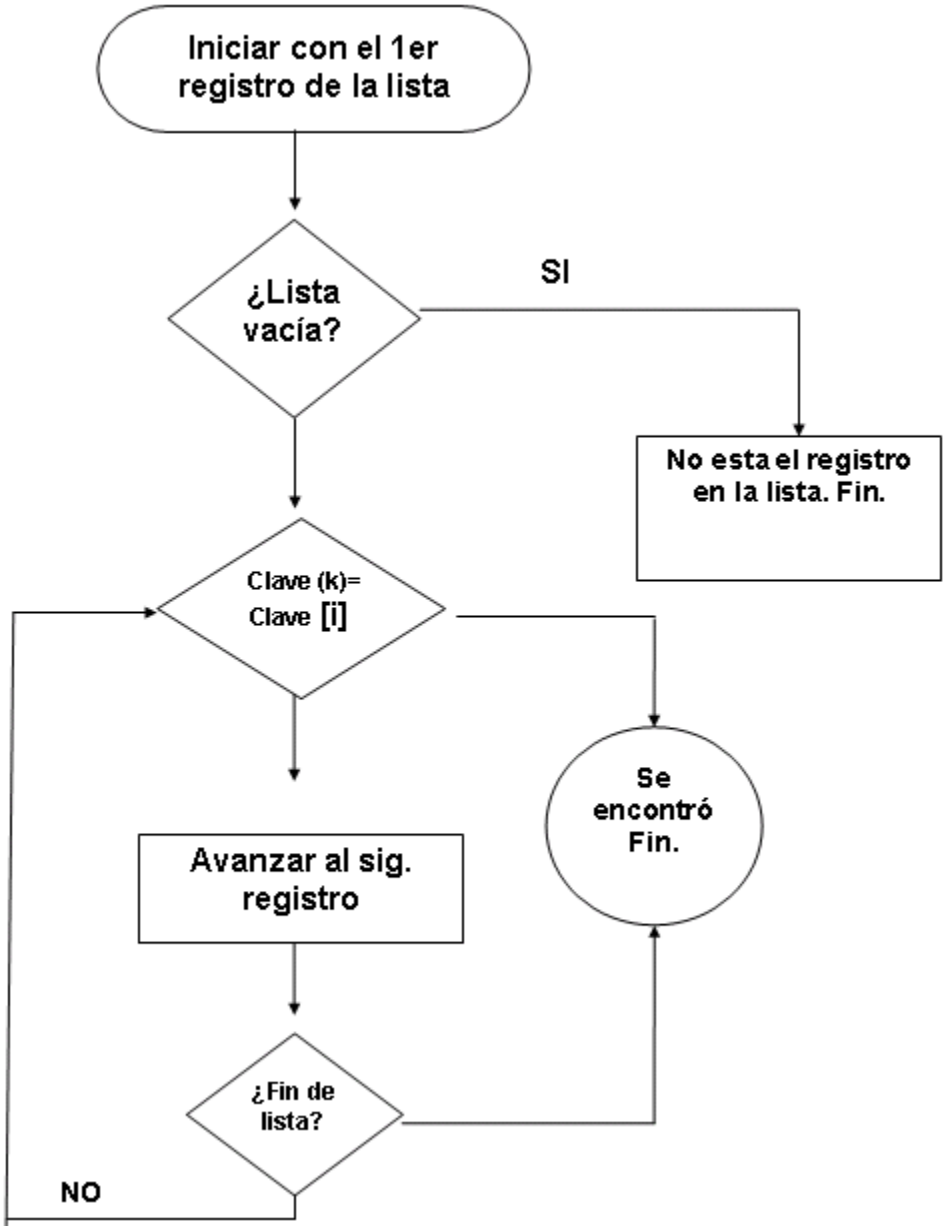


imagen4.bucle anidado

2.1.3 Contadores

Los procesos repetitivos son la base del uso de la computadora. En estos procesos se necesitan contar normalmente con los procesos o acciones internas del bucle, como pueden ser los elementos de un fichero, el número de tareas a realizar por el bucle etc. Una forma de controlar un bucle es mediante un contador. Un contador es una variable cuyo valor se incremento o decrementa, en una interacción. Los procesos repetitivos son la base del uso de las computadoras. En estos procesos se necesitan normalmente contar los sucesos o acciones internas del bucle, como pueden ser los elementos de un fichero de interacciones a realizar por el bucle.

El contador se muestra en el ejemplo siguiente con la variable CONT. Este es un diagrama de flujo para un algoritmo que se desea repetir 50 veces.

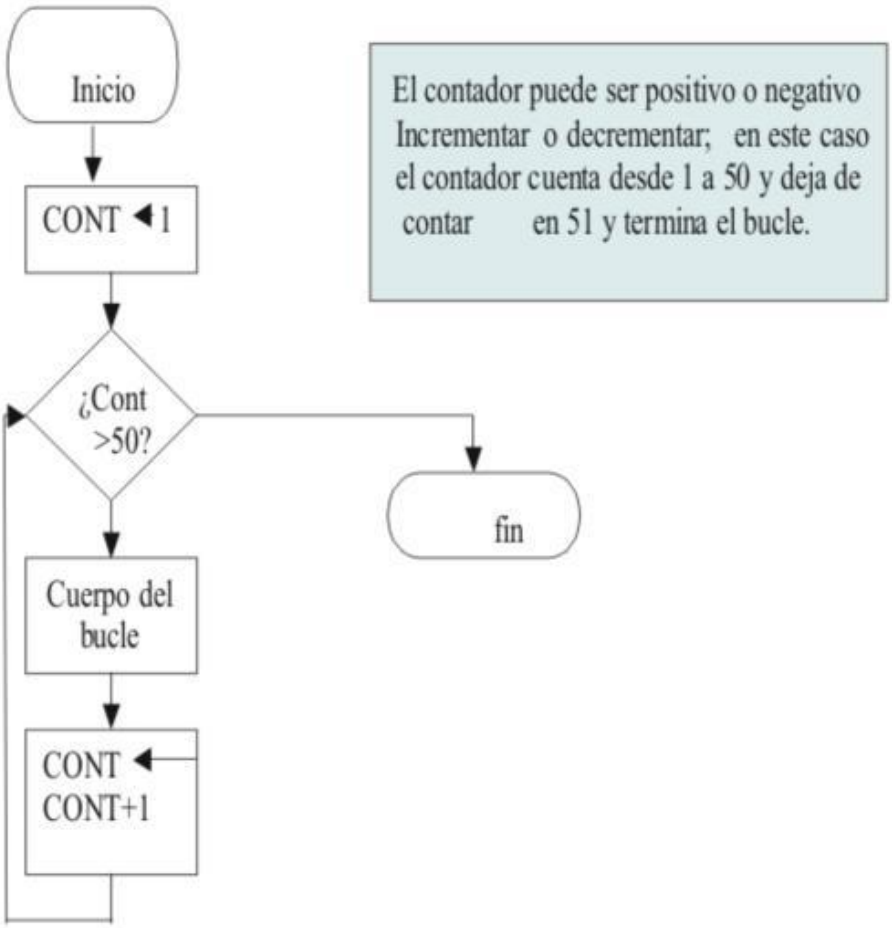


Imagen5.contador

2.1.4 Acumulador

Un acumulado o totalizador es una variable cuya misión es almacenar cantidades variables resultantes de sumas sucesivas. Realiza la misma función que un contador, con la diferencia de que el incremento o decremento de cada suma es variable, en lugar de constante como en el caso del contador. Se representa con la instrucción S S+N, donde N es una variable y no una constante. También podemos decir que un contador es una variable (casi siempre de tipo entero) cuyo valor se incrementa o decrementa en cada repetición de un bucle. Es habitual llamar a esta variable “cont” (de contador) o “i” (de índice).

Representación: <Nombre del acumulador> <nombre del acumulador> + <valor variable>

Ejemplo:
Calcular la suma de los cuadrados de los primeros 100 enteros y escribir el resultado. Se desea resolver el problema usando estructura Desde, Mientras y luego Repetir.

1. Usando la estructura Desde:

```
Inicio
        Suma ← 0
        Desde I = 1 hasta 100 hacer
                Suma ← suma + I * I
        Fin_desde
        Escribir (suma)
Fin
```

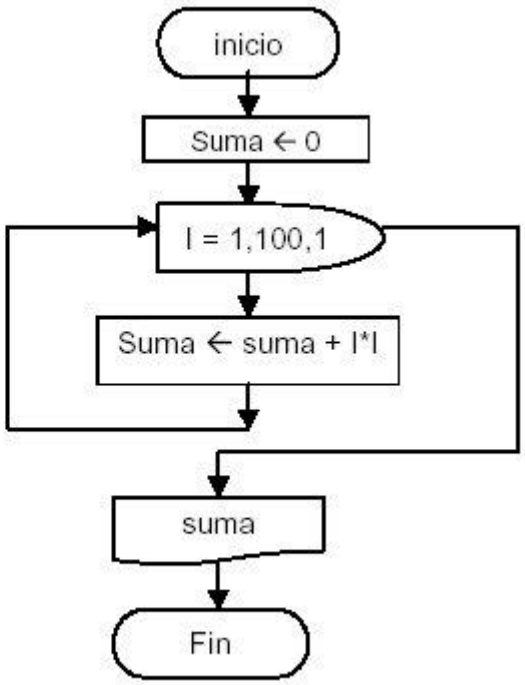
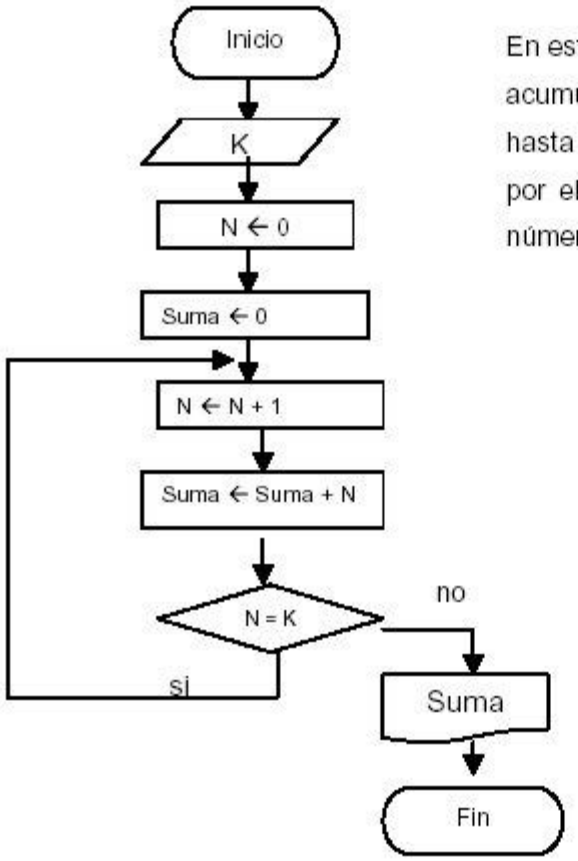


Imagen6. Acumulador

Ejemplo 2.
Elaborar un flujograma para encontrar la suma de los K primeros números enteros



En este caso, utilizamos un contador (N) y un acumulador (suma). N cuenta cada número hasta llegar a K que es el valor máximo dado por el problema. Suma lleva la suma de los números enteros que se van generando

.

Imagen7.ejmplo2acumulador

2.1.5. Estructuras de decisión o selección

Cuando se quiere especificar dos o más caminos alternativos en un algoritmo se deben utilizar estructuras de decisión o solución. Una instrucción de decisión o selección evalúa una condición y en función del resultado de esa condición se bifurca en un determinado punto.

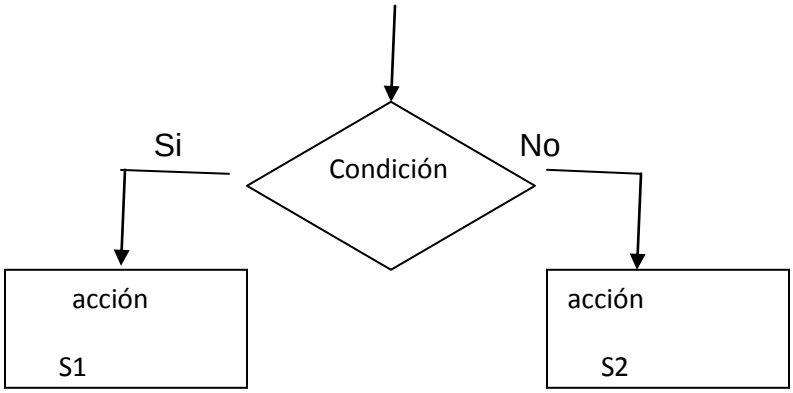


Imagen8. Estructuras de decisión o selección

2.1.6 Interruptores

Un interruptor o conmutador (switsh) a veces se les denomina indicador, o bandera (flag) es una variable que puede tomar diversos valores a lo largo de la ejecución del programa y que permite comunicar información de una parte a otra del mismo. Los interruptores pueden tomar dos valores diferentes 1 y 0 (De ahí su nombre interruptor prendido apagado)

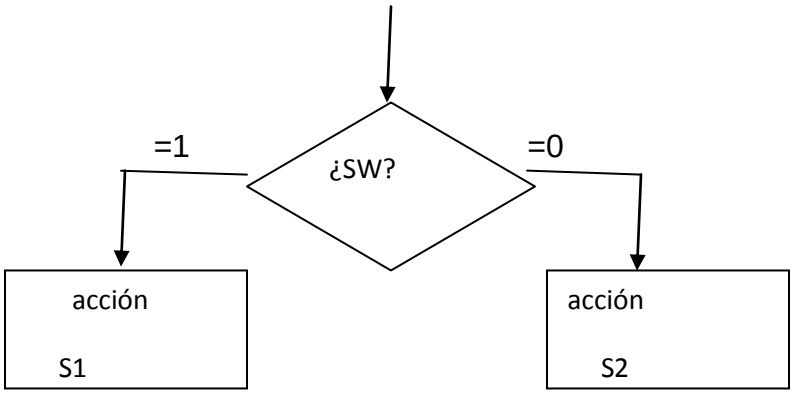


Imagen9. Interruptores

3. Programación

Desde que se desarrollaron las máquinas programables se han desarrollado lenguajes con los cuales las personas puedan dar órdenes a éstas. En su orden los lenguajes de programación se pueden clasificar así:

- Lenguaje de máquina: Las primeras computadoras se programaban en código de máquina. Se puede decir que los programas eran diseñados en código binario. Eran difíciles de leer, difíciles de entender y por su puesto difíciles de corregir. Los programas se caracterizaban por ser pequeños.

- Lenguajes de Bajo Nivel: Para dar solución a lo difícil que era programar en código máquina, se desarrolló un lenguaje conocido como lenguaje ensamblador. Este lenguaje era encargado de tomar algunas palabras comunes a una persona y traducirlas al código máquina. Lo anterior facilitaría un poco la escritura de programas.

- Lenguajes de alto nivel: Como las personas resuelven problemas y se comunican en lenguajes naturales (español, ingles, francés, etc.), se desarrollaron lenguajes de programación que estuvieran mas cerca de ésta manera de resolver problemas.
Como se hace necesario traducir el programa a lenguaje de máquina, en los lenguajes de alto nivel esa operación la realiza algo que se conoce con el nombre de Compilador.

3.1 Tipos de programación

Dependiendo del lenguaje de programación que se elija, se puede hablar del tipo de programación que se va a realizar.

-Secuencial: Se considera programación secuencial a los programas que se diseñan con instrucciones que van unas detrás de otras. Las líneas se ejecutan una a una en secuencia.

-Estructurada: Se considera programación estructurada a la programación que se hace por módulos. Cada módulo realiza alguna tarea específica y cuando se necesite esa tarea simplemente se hace el llamado a ese módulo independiente de que se tengan que ejecutar los demás.

-Orientada a Objetos: Se considera programación orientada a objetos aquellos lenguajes que permiten la utilización de objetos dentro del diseño del programa y el usuario puede pegar a cada objeto código de programa.

- Lógica o de lenguaje natural: son aquellos programas que se diseñan con interfaces tal que la persona o usuario puede ordenar a la máquina tareas en un lenguaje natural. Pueden interactuar como una persona pero nunca llegan a producir conocimiento. Estos lenguajes se desarrollaron con base en las estructuras de sus antecesores. Recorren o navegan las bases de datos obedeciendo a reglas.

-Inteligencia Artificial: Los programas de inteligencia artificial Son programas que se acercan a la inteligencia humana. Estos programas son capaces de desarrollar conocimiento. Este tipo de lenguajes trabajan similar a la mente humana.

3.2 Programación Estructurada

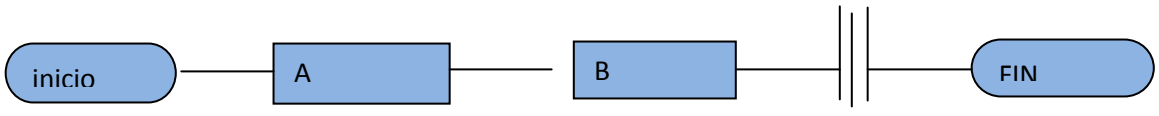
Programación Estructurada es una técnica en la cual la estructura de un programa, esto es, la interpelación de sus partes realiza tan claramente como es posible mediante el uso de tres estructuras lógicas de control:

a. Secuencia: Sucesión simple de dos o mas operaciones.
b. Selección: bifurcación condicional de una o mas operaciones.
c. Interacción: Repetición de una operación mientras se cumple una condición.

3.3 Estructura secuencial o secuencia de instrucción

Es aquella en que una acción sigue la secuencia. La acción B se ejecuta después de la A y ninguna acción puede ejecutarse entre ellas. La acción C, sigue a ala acción B, y así sucesivamente.

La estructura secuencial o secuencia de instrucción



Imgen10. Estructurada selectiva

A,B representan las instrucciones que pueden ser de lectura, asignación o impresión. Las líneas verticales; significan que pueden existir mas instrucciones. Las estructuras selectivas se utilizan para tomar decisiones lógicas; de ahí que se suelan denominar también estructuras de decisión o alternativas. En las estructuras selectivas se evalúa una condición y en función del resultado de la misma se escoge una opción u otra. Las condiciones se especifican usando expresiones lógicas. La representación de una estructura selectiva se hace con palabras en pseudocódigo (if, then, else o bien en español si, entonces, sino), con una figura geométrica en forma de rombo o bien con un triángulo en el interior de una caja rectangular. Las estructuras lógicas selectivas se encuentran en la solución algorítmica de casi todo tipo de problemas. Se utilizan cuando en el desarrollo de una solución de un problema se debe tomar una decisión para establecer un proceso o señalar un camino alternativo a seguir.

CLASES
Simples
Compuestas
Múltiples
ESTRUCTURAS SELECTIVAS
Anidadas

Imagen11. Estructura

3.4 Operaciones o instrucciones de asignación

Se utilizan para asignar valores o expresiones a una variable. La asignación es una operación destructiva, esto significa que si la variable tenía asignado un valor anteriormente, este se destruye, conservando ahora el nuevo valor.

Normas para la creación de operaciones o instrucciones de asignación.

- El primer término debe ser siempre una variable.
- El valor del segundo termino (2) es asignado al primer termino o variable (1)
- El símbolo utilizado para indicar asignación es el =(igual a).

4. Pasos para la solución de problemas.

Los pasos para la solución de un problema de computador son:
1. Análisis de la solución
2. Definición de variables
3. Diseñar diagrama de Flujo
4. Prueba de Escritorio.
5. Escribir Algoritmo.

1. Leer el enunciado (texto) hasta entender que esta pidiendo el problema
2. Escribir la definición de variables, este es muy importante, ya que ayuda a buscar la solución al problema planteado.

Para la definición de variables de sebe tener en cuenta lo siguiente:

- Asignar el nombre a cada una de las variables que se necesitan para resolver el problema.
- Al lado de cada una de las variables, hacer una descripción del uso(s) que se le va a dar a dicha variable. Ejemplo COD= sirve para leer el código de un estudiante
- Cuando se esta en proceso de definir una variable, se debe tener presente, par que no se olvide escribir ninguna, es conveniente organizarlas en el siguiente orden. Entrada, proceso, salida. (E,P,S)
- Las variables de entrada son aquellas que sirven para leer o asignar un valor inicial con el cual se puede comenzar una acción o proceso.
- Las variables de proceso, son aquellas que se utilizan para calcular y guardar un valor después de afectada la operación que se necesite.
- Las variables de entrada pueden ser también de salida.
- Las variables de proceso pueden también ser de salida
- Las variables de entrada se pueden también utilizar como de proceso.
- En la definición de variables se deben indicar todos los usos que se le van a dar a las variables.

3. Construir el Diagrama de Flujo: que es la representación grafica de una serie de pasos ordenados y lógicos, que llevan a la solución de un problema, a la realización de una actividad o tarea o a la obtención de una respuesta.

4 Practicar una prueba de escritorio. Esta prueba es manual por eso se llama de escritorio, porque normalmente se usa papel y lápiz para realizarla. El objetivo de esta prueba es determinar si la solución dada en ese momento, a través de un diagrama, es la solicitada en el enunciado del ejercicio. Para ello se debe hacer lo siguiente:

- Sacar en forma horizontal y en orden, todas las variables utilizadas
- Recorrer paso a paso el diagrama e ir indicando todo lo registrado debajo de las variables correspondientes. Si la prueba indica que los resultados no son los esperados, se debe volver a desarrollar el diagrama.

5. Escribir el algoritmo correspondiente. Ejemplo. Escribir un algoritmo que calcule e imprima el valor de C.

```
Incio
  Lea A,B
  Calcula C= A+B
 Escriba C
Fin-algoritmo
```

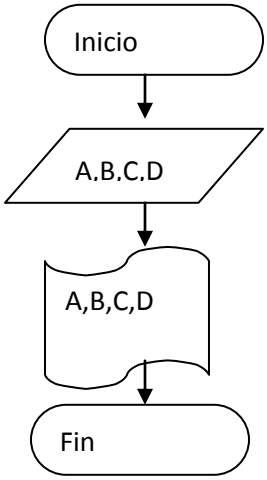
4.1 Ejemplo

1. Leer 4 números y luego imprimirlos

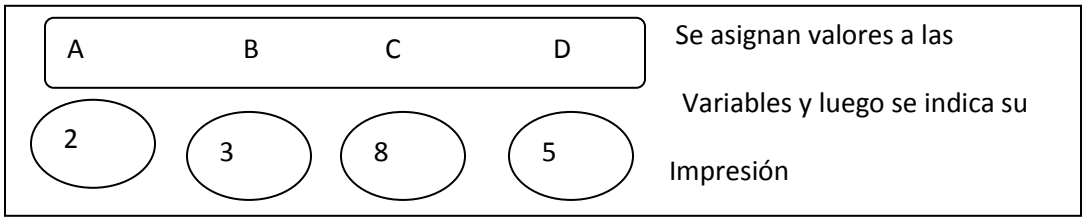
a. Análisis de la solución: en este caso es necesario definir 4 variables, cada una para leer un número.

b. Definición de las variables: A, se utiliza para leer el primer numero y posteriormente visualizarlo : B, se utiliza para leer el segundo numero y posteriormente visualizarlo : C, se utiliza para leer el tercer número y posteriormente visualizarlo: D, se utiliza para leer el cuarto número y posteriormente visualizarlo.

c. Diagrama de flujo



d. Prueba de escritorio:



En la pantalla los valores aparecen

2	3	4	8	5

imagen12. Diagrama y prueba de escritorio

Cuando se compara lo encontrado en la prueba de escritorio, con el enunciado se cumplen las 2 cosa. Que se leen los datos. Se imprimen o muestra los valores guardados en las variables, por lo tanto se puede continuar.

e. Escribir el algoritmo: si el diagrama funciona correctamente solo, queda interpretarlo y transformarlo.

```
Inicio
 lea A;B;C;D
 imprima A;B;C;D
Fin_algoritmo
```

2.. Desarrolle un algoritmo que le permita calcular el área (A) de un segmento de círculo.
Análisis: Para calcular el área de un segmento de círculo lo primero que hay que hacer es leer el valor del radio del vinculo y leer el valor de X que es la distancia del centro al segmento. Una vez leído dichos valores se calcula aplicando la fórmula respectiva y por último se escribe el valor del área.

Leer (X y r)
Calcular (Area)
Escribir

Seudo lenguaje
Inicio
Leer el valor de X y r
Calcular el valor de área
Escribir el valor de Area (A)
Fin

Diagrama de flujo

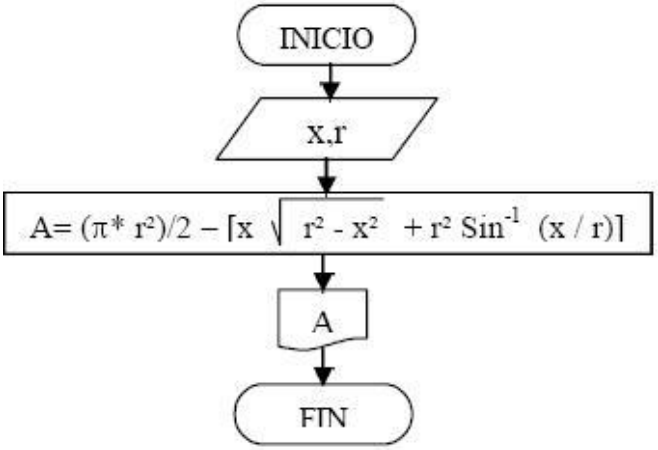


imagen 13.diagrama de flujo

5. Estructura de Selección o de Instrucción

Las operaciones o instrucciones de decisión trabajan con las estructuras de selección o instrucciones y sirven para la solución de casi todo tipo de problemas. Se utilizan cuando en el desarrollo de la solución de un problema se debe tomar decisión, con el fin de establecer que camino seguir. El símbolo para indicar selección o pregunta es el rombo. Mínimo debe tener 2 salidas

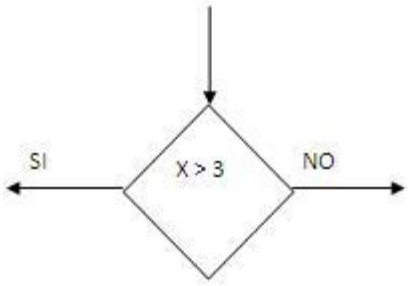


imagen14.seleccion

En las estructuras de selección de instrucciones cuando la condición se cumple tomara el camino SI o Verdadero; si la condición no se cumple, se ira por el camino NO o Falso (si no) en cualquiera de los casos después de ejecutar las instrucciones que se encuentren en el respectivo camino, saldrá de la estructura e ira a ejecutar las instrucciones que se encuentran por fuera y a continuación de ella, hasta encontrar el fin del proceso.

Las operaciones o instrucciones de decisión se trabajan en las estructuras de de selección de instrucción llamadas comúnmente.

Estructura selectiva simple o compuesta.

- La estructura selectiva lógica “si entonces” permite que el flujo siga por un camino específico si se cumple una condición o un conjunto de condiciones.
- Si al evaluar la condición (o condiciones) el resultado es verdadero, entonces se ejecuta (n) cierta (s) operación (es), luego continua con la secuencia normal del proceso.
- Se utiliza cuando alguna operación está condicionada para que se lleve a cabo, pero no tiene una opción alterna

formato:

si condición entonces
operación (es)
{ Fin del condicional }

Donde:

si.- identifica la estructura selectiva
condición.- expresa la condición o conjunto de condiciones a evaluar
entonces.- indica el curso de acción a seguir si se cumple la condición
Operación.- expresa la operación o conjunto de operaciones.
fin del condicional.- indica el fin de la estructura de selección (si)

Funcionamiento:

Al llegar al si se evalúa la condición (es):
a). Si se cumple, se ejecuta (n) la (s) operación (es) del entonces y luego salta hasta el siguiente paso después del fin del condicional.
b). Si no se cumple, salta hasta después del fin del condicional, es decir no hace nada.

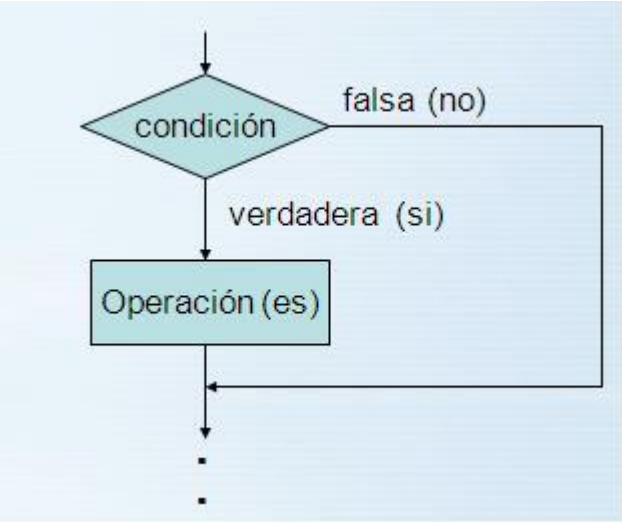


imagen15 estructura si.

Ejemplo.

Leer la calificación o nata de un estudiante; si la nota es mayor que 3, escribir o imprimir un mensaje que diga Aprobado, si no terminar.

a. Análisis de la solución

El ejercicio pide que leamos una nota; que después se pregunte si es mayor que 3 y dependiendo de eso, si resulta ser > 3 imprimir el mensaje que nos indique el enunciado.

b. Definición de Variables

Solo se necesita una variable para leer la nota.

c. Diagrama de flujo

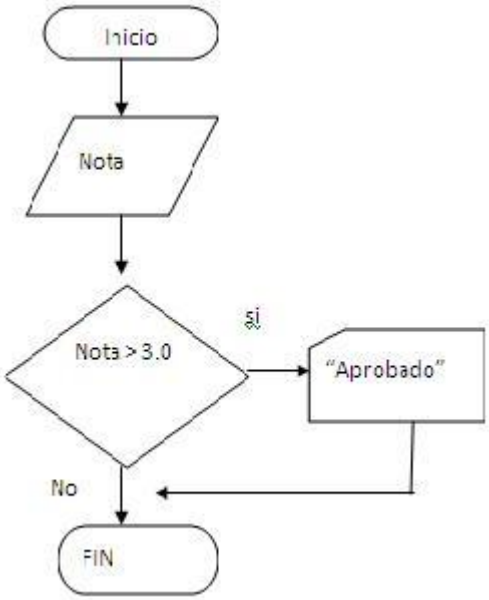


Imagen16. Ejemplo estructura si.

d. Prueba de escritorio

Para este ejemplo se harán 2 pruebas de escritorio con notas aleatorias, de tal manera que se vaya primero por un camino y luego por otro.

Nota	3.5	"aprobado"	Cuando la condición se cumple imprime el Aprobado
	2.8	cuando la condición no se cumple va a buscar el fin de la estructura.	

Imagen17. Prueba de escritorio

e. Algoritmo

```
Inicio
  Lea Nota
  Si Nota > 3 entonces  Imprima " Aprobado"
 Fin si
Fin algoritmo
```

5.1 La estructura Si entonces SI/No (estructura selectiva doble o compuesta. If – Then – Else)

Permite que el flujo del diagrama se bifurque por 2 caminos diferentes. Si al evaluar la condición el resultado el resultado es Si o verdadero, entonces se ejecutaran las instrucciones que se encuentran por dicho camino; si la condición no se cumple se ejecutaran las instrucciones que se encuentran por No (Si-No) o falso. Si al evaluar la condición (o condiciones) el resultado es verdadero, entonces sigue por un camino específico y se ejecuta (n) cierta (as) operación (es). Por otra parte, si el resultado es falso entonces se sigue por otro camino y se ejecuta (n) otra (s) operación (es). En ambos casos, luego de ejecutarse la (s) operación (es) indicada (s), se continúa con la secuencia normal del proceso. Por la naturaleza de éstas, se debe ejecutar una o la otra, pero no ambas a la vez, es decir, son mutuamente excluyentes.

Formato:

```
si condición
entonces
        hacer operación 1
   sino
        hacer operación 2
{ Fin del condicional }
```

Donde:

si.- identifica la estructura selectiva
condición.- expresa la condición o conjunto de condiciones a evaluar
entonces.- indica el curso de acción a seguir si se cumple la condición.

Operación 1.- expresa la operación o conjunto de operaciones.
Operación 2.- expresa la operación o conjunto de operaciones.
fin del condicional.- indica el fin de la estructura de selección (si)

Funcionamiento:

Al llegar al si se evalúa la condición (es):

a). Opción verdadera (entonces). si se cumple, se ejecuta (n) la (s) operación (es) del entonces y luego salta hasta el siguiente paso después del fin del condicional.

b). Opción falsa (sino). De lo contrario, salta hacia el sino, ejecuta la (s) operación (es), y después salta hasta el siguiente paso después del fin del condicional.

Ejercicio

Después de leer 3 notas y calcular el promedio, determinar o imprimir si un estudiante aprueba o reprueba una asignatura. Sabiendo que aprobara si el promedio es > = 3.0 si no reprobara.

a.. Análisis de la solución. Se deben definir tres variables cada una para leer una nota y luego definir la variable que guardara el promedio de dichas notas; luego se sabe si el promedio es > = 3.0 o inferior.

b. Definición de variables

Nota1, Nota2 Nota 3 = Sirven para leer cada una de las notas de la asignatura.

Prom = sirve para calcular y guardar le promedio de las notas leídas.

c. Diagrama de Flujo.

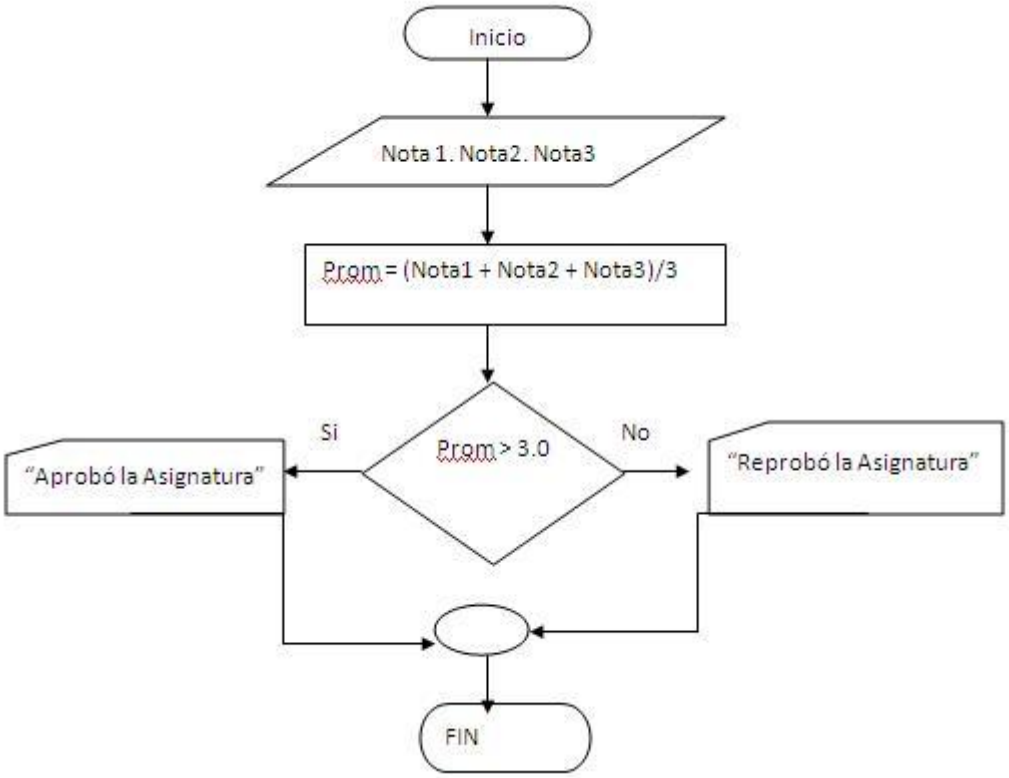


Imagen18.diagrana. si/no

d. prueba de escritorio

Nota 1	Nota 2	Nota3	Prom	
3.5	4.5	2.7	3.43	"Aprobó la asignatura"
2.7	3.2	2.3	2.73	"Reprobó la asignatura"

Imagen 19. Prueba escritorio si.no

e.. Algoritmo

```
Inicio
  Lea Nota1, Nota2 Nota3
   Prom = (Nota1 + Nota2 + Nota3 ) /3
   Si Prom > = 3.0 entonces
     Imprima " Aprobó la asignatura"
    Si no
     Imprima " Reprobó la asignatura"
Fin si
Fin algoritmo
```

5.1.1 Estructura selectiva anidada

Existen numerosos casos en el desarrollo de la solución de problemas en el que luego de tomar una decisión y marcar el camino correspondiente a seguir, es necesario tomar otra decisión.. Se señala, luego de evaluar las condiciones, la rama correspondiente a seguir, y nuevamente podemos tener que tomar otra decisión. El proceso puede repetirse numerosas veces. En este caso, para resolver el problema, estamos aplicando estructuras selectivas en cascada o anidadas. Algunos ejemplos son:

```
n       si condición1 entonces
              n1      si condición2
                            entonces
                                 hacer operación21
                            sino
                                 hacer operación22
              n2 fin del condicional del paso n1
        n + 1 fin del condicional del paso n
```

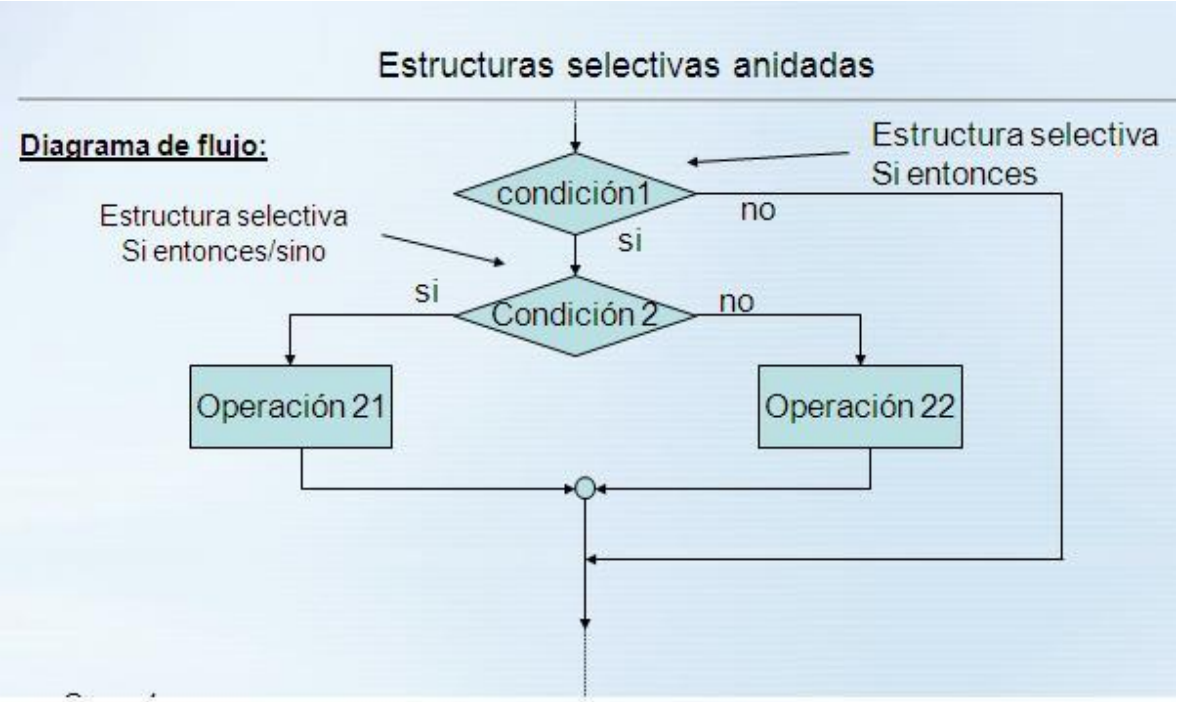


Imagen20. Estructura selectiva anidada

Ejemplo: desarrollar un algoritmo y diagrama de flujo el cual, dado un valor entero, determinar si es cero, positivo o negativo.

Algoritmo Dado un valor entero determinar si es cero, positivo o negativo

```
inicio
Declaración de variables: num: entero
Leer num
si num ==0
   entonces
     imprimir “cero”
            sino
                 si num > 0
                  entonces
                       imprimir “positivo”
                  sino
```

imprimir “negativo”
fin del condicional paso 4.1
Fin del condicional del paso 4

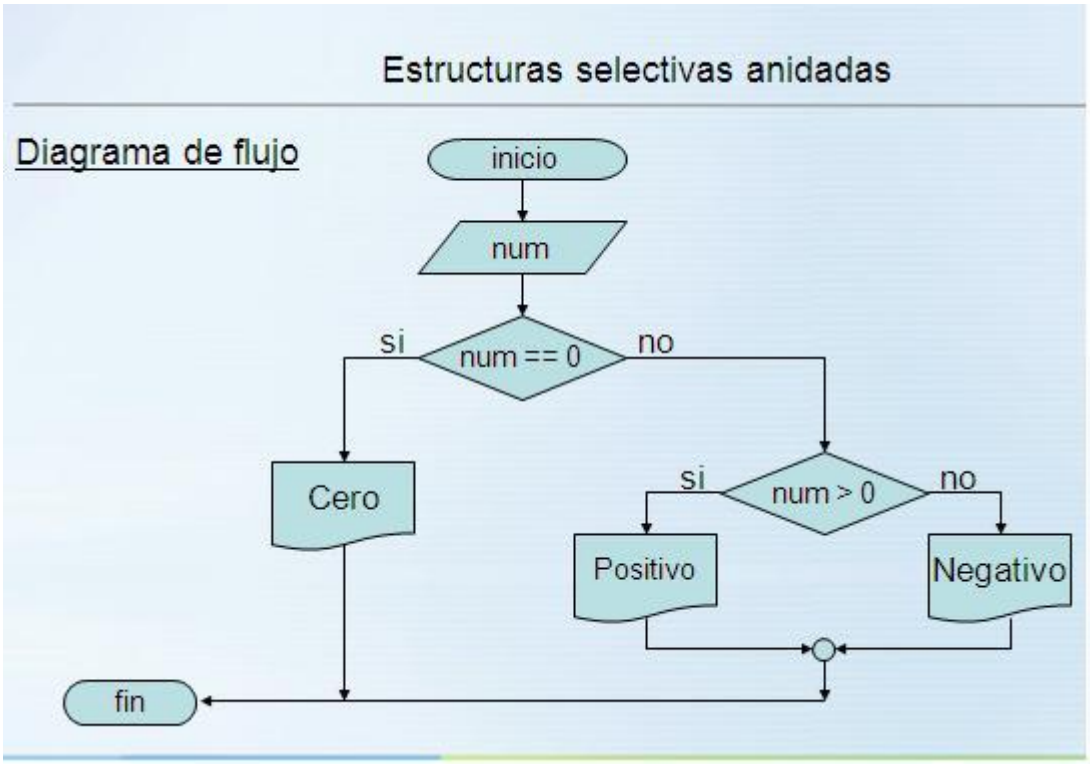


Imagen21. Ejemplo diagrama estructura selectiva anidada