

Instrucciones de bifurcación

1. Qué es una bifurcación o toma de decisión

Los algoritmos y programas secuenciales se caracterizan por ejecutarse instrucción tras instrucción desde la primera a la última en el mismo orden en que son indicadas.

Pero no todos los problemas pueden ser resueltos de esta manera. A veces, la solución final o las operaciones que hay que realizar dependen de una serie de condicionantes.

En estos casos, el ordenador debe examinar cuales son las distintas posibilidades y elegir entre ellas la más adecuada.

Tanto el punto de decisión como las alternativas deben estar claramente expresadas en el algoritmo.

Además, la decisión se toma en función del resultado de evaluar una expresión lógica.

Operadores lógicos o de condición.

OPERADOR	DESCRIPCIÓN
==	Es igual
!=	Es distinto
<, <=, >, >=	Menor, menor o igual, mayor, mayor o igual
&&	Operador and (y)
	Operador or (o)
!	Operador not (no)

2. La sentencia si

Esta instrucción está presente en cualquier lenguaje de programación. Podemos destacar las siguientes variantes:



AVILÉS

2.1 Bifurcación simple

Formato:

Si CONDICIÓN

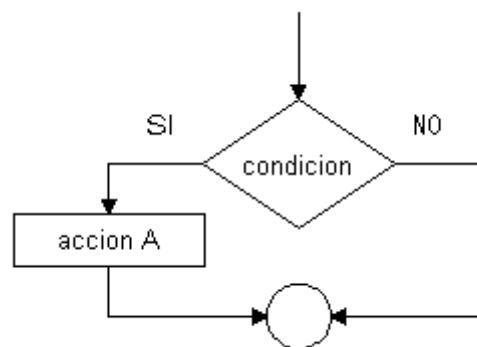
entonces I1 I2 ... In

finsi

Se ejecuta de la siguiente forma:

- primero se evalúa la expresión lógica
- si resulta verdadera, se ejecuta las sentencias
- en caso contrario, no se hace nada en particular
- tanto si resultó verdadera como falsa, el programa continúa por la siguiente instrucción

En un organigrama, se representa así:



2.2 Bifurcación doble

Formato:

Si CONDICIÓN

entonces I1 I2. ... In

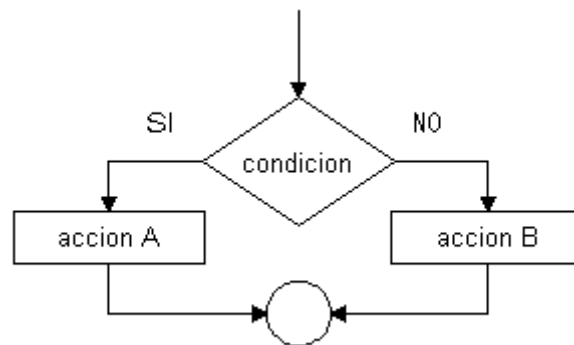
sino J1 J2. ... Jk

finsi

Se ejecuta de la siguiente forma:

- primero se evalúa la expresión lógica
- si resulta verdadera, se ejecuta la **sentencias I1,I2...**
- en caso contrario, se ejecuta la **sentencias J1,J2...**
- tanto si resultó verdadera como falsa, el programa continúa por la siguiente instrucción

En un organigrama, su representación es la siguiente:



2.3 Bloque de sentencias si (si anidados)

Si dos alternativas no son suficientes para indicar una toma de decisión, será necesario recurrir a un bloque de sentencias if, que tiene el siguiente aspecto:

si (expresión lógica 1)

entonces sentencia1

sino

si (expresión lógica 2)

entonces sentencia2

sino

si (expresión lógica n)

entonces sentencian

sino sentencia-n

finsi

.....

finsi

finsi

La ejecución de este bloque consiste en ir evaluando de arriba abajo las expresiones lógicas hasta encontrar una que sea verdadera, en cuyo caso se ejecutará su sentencia asociada.



3. Bifurcaciones en pseudocódigo

Alternativa doble	Alternativa simple
Si CONDICIÓN	Si CONDICIÓN
entonces I1 I2. ... In	entonces I1 I2 ... In
sino J1 J2. ... Jn	finsi
finsi	

En esta instrucción, CONDICIÓN es la expresión lógica. Tras evaluarla, si da verdadero se ejecutan las instrucciones I1, I2, ... In, en caso contrario, J1, J2, ... Jn en la primera, y en la segunda, no se haría nada en particular.

4. La sentencia switch

El bloque de sentencias “si” resuelve múltiples bifurcaciones escalonadas, pero existe en la mayoría de los lenguajes de alto nivel otra instrucción que produce un código más claro y compacto, aunque no puede usarse en todos los casos. Es la sentencia switch:

Formato:

switch (expresión)

caso Literal-1

sentencia-1

caso Literal-2

sentencia-2

.....

default

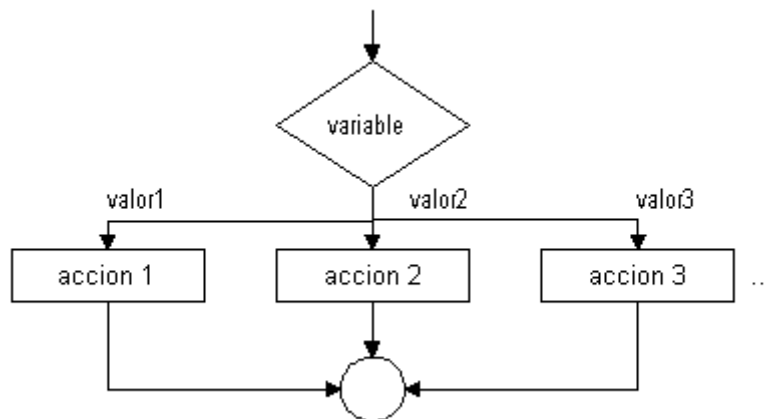
sentencia-n

finswitch

La ejecución de esta instrucción es como sigue:

- se evalúa la **expresión**
- si coincide con algún **Literal-i**, se ejecutará la **sentencia-i** asociada
- **default** indica que, de no coincidir con ningún caso, se ejecutará la **sentencia-n**. Es opcional.

En un organigrama se indica como:



switch (opcion)

```

caso 1:
    Visualizar "Sumar"
case 2:
    Visualizar "Restar"
case 3:
    Visualizar "Multiplicar"
case 4:
    Visualizar "Dividir"
default:
    Visualizar "No operación"
  
```

finswitch;

5. Relación de ejercicios

1. Hacer el pseudocódigo de un programa que lee un número por teclado y visualizar si es mayor que 100.
2. Hacer el pseudocódigo de un programa que lee un número por teclado y compruebe si dicho número es nulo o no.
3. A partir de la nota numérica entera de un alumno, construye el programa que indique su calificación en texto: Suspenso o Aprobado.
4. Realizar el pseudocódigo de un programa que permita calcular el cuadrado de un número que se introduce por teclado, si este es positivo. Visualizar en pantalla mensajes aclaratorios de los datos que se están pidiendo o mostrando.
5. Una tienda ofrece un descuento del 15% sobre el total de la compra durante el mes de octubre (mes 10). Dado un mes y un importe, calcular cuál es la cantidad que se debe cobrar al cliente.
6. Diseñar un algoritmo que pida por teclado tres números; si el primero es negativo, debe imprimir el producto de los tres y si no lo es, imprimirá la suma.

7. Realizar el pseudocódigo del programa que, dados dos números enteros, indique si el primero es divisible por el segundo (operador resto()).
8. Haz el programa que calcule las raíces de una ecuación de segundo grado, sólo si son reales, a partir de los coeficientes a, b y c.

```
leer a
leer b
leer c

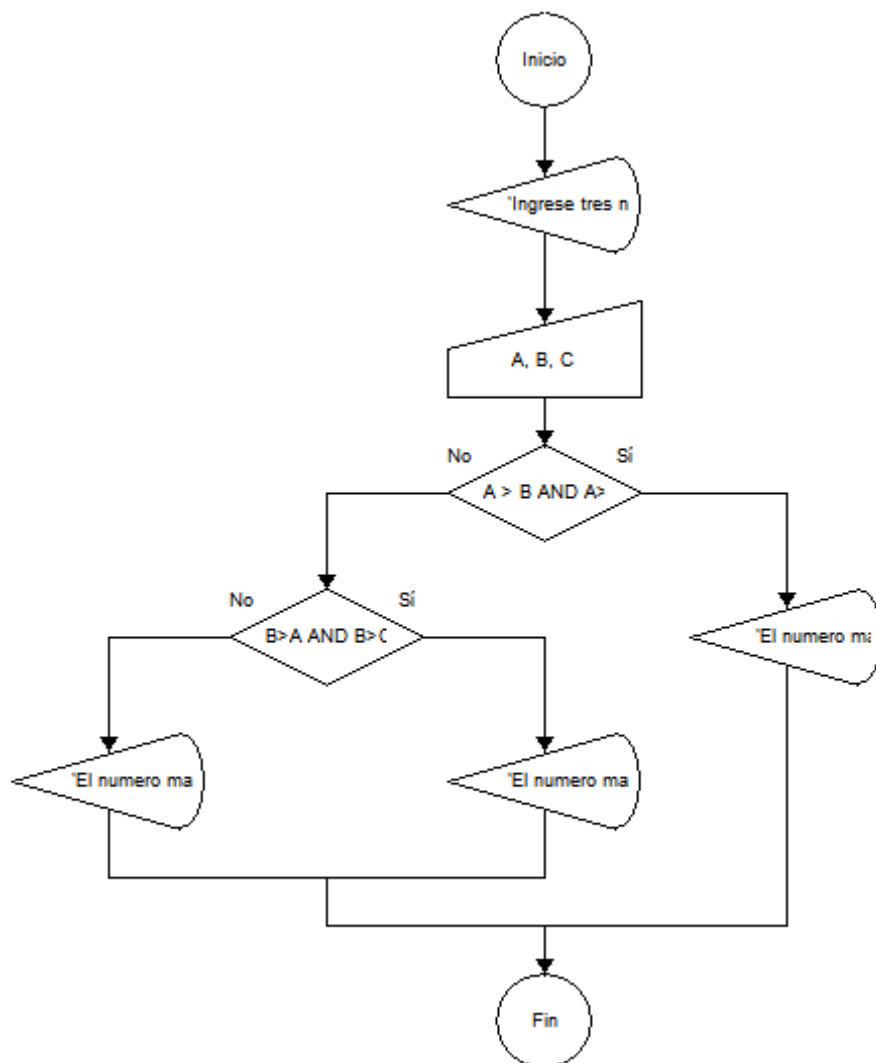
discriminante =  $b^2 - 4ac$ 

si discriminante < 0
    'La ecuación no tiene soluciones reales'

si discriminante = 0  $\rightarrow x = -b / 2a$ 

si discriminante > 0
     $x1 = (-b - \sqrt{\text{discriminante}}) / 2a$ 
     $x2 = (-b + \sqrt{\text{discriminante}}) / 2a$ 
```

9. Crea un programa-calculadora con un menú de opciones que dé a elegir entre suma, resta, multiplicación, división o resto entre dos números enteros.
10. Programa que muestra por pantalla 3 números de mayor a menor. Los números habrán sido introducidos previamente por teclado de manera aleatoria. Previamente concluir el ordinograma y realizar el pseudocódigo que visualiza el mayor de esos tres números.



11. A partir de la nota numérica entera de un alumno, construye el programa que indique su calificación en texto: Suspenso, Aprobado, Bien, Notable, Sobresaliente.
12. Crear el pseudocódigo para un programa que pida un número entero distinto de cero y nos muestre en pantalla un mensaje indicándonos si el número es par o impar.
13. En un garaje se ha establecido una promoción de las llantas de lujo, dicha promoción consiste en lo siguiente:
 - Si se compran menos de cinco llantas el precio es de 30000 euros cada una, de 25000 euros si se compran de 5 a 10 y de 20000 si se compran más de 10.
 - Obtener la cantidad de dinero que una persona tiene que pagar por cada una de las llantas que compra y la que tiene que pagar por el total de la compra.
14. Una frutería ofrece las manzanas con descuento según la siguiente tabla:

NUM. DE KILOS COMPRADOS	% DESCUENTO
0 – 2	0%
2.01 – 5	10%

5.01 – 10	15%
10.01 en adelante	20%

Determinar cuánto pagara una persona que compre manzanas es esa frutería.

15. Una compañía dedicada al alquiler de automóviles cobra una cantidad fija de 300.000 euros para los primeros 300 Km. de recorrido.

Para más de 300 Km. y hasta 1000 Km., cobra una cantidad adicional de 15000 por cada kilómetro en exceso sobre 300.

Para más de 1000 Km. cobra una cantidad adicional de 10000 por cada kilómetro en exceso sobre 1000.

Los precios ya incluyen el 16% del IVA. Diseñe un algoritmo que determine el precio a pagar por el alquiler de un vehículo y la cantidad incluida del impuesto.

16. Realice un Pseudocódigo que calcule el salario neto, de unos obreros cuyo trabajo se paga en horas. El cálculo se realiza de la siguiente forma:

- Las primeras 35 horas a una tarifa fija de 50 euros.
- Las horas extras se pagan a 1.5 más de la tarifa fija.
- Los impuestos a deducir de los trabajadores varían, según el sueldo mensual si el sueldo es menos de 20.000 euros, el sueldo es libre de impuesto y si es al contrario se cobrará un 20% de impuesto.



AVILÉS

DEPARTAMENTO DE INFORMÁTICA Y COMUNICACIONES

Desarrollo de Aplicaciones Web

Módulo Programación

Unidad 1