

## SELECTOR MÚLTIPLE

Esta estructura permite remplazar una serie de decisiones lógicas anidadas, cuando estas se hacen sobre una misma variable, y los valores que puede tomar dicha variable son finitos, fijos y conocidos, Su sintaxis es:

En caso de (variable) haga

Valor 1 : instrucciones en el caso de que la variable contenga el valor 1

Valor 2 : instrucciones en el caso de que la variable contenga el valor 2

Valor 3 : instrucciones en el caso de que la variable contenga el valor 3

.  
.   
.

Valor N : instrucciones en el caso de que la variable contenga el valor N

E.O.C.: Instrucciones en el caso de que la variable contenga un valor diferente a los establecidos en las opciones anteriores

Fin Caso

Su funcionamiento es el siguiente:

Cuando el control llega a la cabecera del selector, se evalúa el valor de la variable y el control es transferido al grupo de instrucciones que se encuentren al frente del valor que tenga la variable en ese momento, en caso de no coincidir con alguno de los valores, el control es transferido a las instrucciones que se encuentren al frente del E.O.C. (En Otro Caso, en los lenguajes por lo regular es el ELSE).

Ejemplo:

Se tiene el registro de un estudiante así: Nombre, estrato (1, 2, 3, 4, 5, o 6), número de créditos matriculados y valor unitario del crédito (cada materia tiene asignado un número de créditos que por lo regular, no siempre, corresponde con la intensidad horaria semanal).

Se debe crear un algoritmo que imprima el nombre del estudiante, el valor bruto de la matrícula, el valor del descuento, el valor del recargo y el valor neto a pagar sabiendo que se debe regir por la siguiente tabla:

ESTRATO	DESCUENTO	RECARGO
1	40%	0%
2	30%	0%
3	10%	0%
4	0%	10%
5	0%	20%
6	0%	40%

ANÁLISIS:

D.E. { nom, estr, ncm, vuc

D.S. { nom, Vrbto, VrDesc, VrRec, VrNto

$$\text{D. P.} \left\{ \begin{array}{l} \text{VrVrBto} = \text{ncm} * \text{vuc} \\ \text{VrVrDesc} = \text{Depende del estrato} \\ \text{VrVrRec} = \text{Depende del estrato} \\ \text{VrVrNto} = \text{VrBto} - \text{VrDesc} + \text{VrR|ec} \end{array} \right.$$

INICIO

IMPRIMA "Digite el nombre del estudiante:"

LEA nom

IMPRIMA "Digite el estrato:"

LEA estr

IMPRIMA "Digite el número de créditos:"

LEA ncm

IMPRIMA "Digite el valor del crédito:"

LEA vuc

$\text{VrBto} = \text{ncm} * \text{vuc}$

En caso de (estr) haga

1:  $\text{VrRec} = 0$

$\text{VrDesc} = \text{VrBt} * 40 / 100$

2:  $VrRec = 0$

$VrDesc = VrBt * 30 / 100$

3:  $VrRec = 0$

$VrDesc = VrBt * 10 / 100$

4:  $VrRec = VrBto * 10 / 100$

$VrDesc = 0$

5:  $VrRec = VrBto * 20 / 100$

$VrDesc = 0$

6:  $VrRec = VrBto * 40 / 100$

$VrDesc = 0$

E.O.C.: Imprima “ el valor digitado para el estrato está errado”

Fin caso

$VrNto = VrBto - VrDesc + VrRec$

Imprima nom, Vrbto, VrDesc, VrRec, VrNto

Termine

## CICLOS

La mayoría de las veces los algoritmos deben trabajar para varios registros, a los cuales se les debe aplicar el mismo procedimiento (solo cambian los valores sobre los cuales se trabaja), para esto los lenguajes de programación proveen unas estructuras que permiten implementar la repetición de los procesos, a estas estructuras se les conoce con el nombre de ciclos los cuales son:

### CICLO PARA (for)

Se dice que es un ciclo predeterminado debido a que para poderse utilizar se debe conocer de antemano la cantidad de veces que se va a repetir el grupo de instrucciones. Su sintaxis es:

```
PARA I = LI, LF, K
_____
_____
_____
_____
FIN PARA
```

} Instrucciones que se han de repetir

Donde:

I → Es el índice del ciclo, es un campo variable numérico entero, que toma como primer valor LI y que se incrementa en K cada vez que se ejecutan las instrucciones incluidas dentro del ciclo.

LI → Se le denomina valor inicial, es un campo numérico entero, no necesariamente variable. Es el primer valor que toma el índice.

LF → Se le denomina límite final, es un campo numérico entero, no necesariamente variable, cuando el índice supera este valor se da por terminado el proceso repetitivo y el control es transferido a la instrucción que se encuentre inmediatamente después del ciclo.

K → Incremento, es el valor que se le suma al índice cada vez que se ejecuta el grupo de instrucciones que hay dentro del ciclo.

**NOTA:** El incremento puede ser un valor negativo, en este caso el LI debe ser mayor que el LF ya que el índice se irá decrementando.

### Ejemplo 1:

Escribir un algoritmo que muestre en la pantalla los múltiplos de 7 comprendidos entre 1 y 70 inclusive

**ANÁLISIS:** Como podemos observar, se necesita una variable que comience en 7 y que cada vez, después de imprimirla se le sume 7, debe detenerse cuando llegue a 70.

Lógicamente podemos implementar un ciclo PARA cuyo valor inicial sea el 7, el valor final sea el 70 y el incremento sea de 7 así:

Inicio

Para J = 7 , 70, 7

Imprima J

Fin para

Termine

## DATO IDENTIFICADOR

Es la cantidad de registros que se van a procesar, este dato se le puede pedir al usuario que lo ingrese antes del ciclo (mediante una instrucción LEA), y lo utilizamos como LF en la cabecera del ciclo; veamos un ejemplo:

### Ejemplo 2:

Se tiene una cantidad conocida de registros, cada registro contiene el número del vuelo en una aerolínea, la fecha, el destino (suponga que solo puede ser Bogotá, Cali o Barranquilla) y el número de pasajeros transportados. Se debe crear un algoritmo que muestre por cada vuelo el número y el valor total de los pasajes sabiendo que el valor de cada pasajes es: a Bogotá \$600.000, a Cali \$750.000 y a Barranquilla \$ 980.000.

ANÁLISIS:

D.E. { NrVue, Fecha, Dest, NrPas

D.S. { NrVue, Total

INICIO

IMPRIMA "Digite la cantidad de registros:"

LEA N

PARA K = 1, N, 1

IMPRIMA "Digite el número del vuelo:"

LEA NrVue

IMPRIMA "Digite la fecha:"

LEA Fecha

IMPRIMA "Digite el Destino:"

LEA Dest

IMPRIMA "Digite el número de pasajeros transportados:"

LEA NrPas

EN CASO de ( Dest) haga

"Bogotá":  $\text{Total} = \text{NrPas} * 600.000$

"Cali":  $\text{Total} = \text{NrPas} * 750.000$

"Barranquilla":  $\text{Total} = \text{NrPas} * 980.000$

E.O.C.: Imprima "El Destino digitado no es válido"

Fin caso

Imprima NrVue, Total

FIN PARA

Termine