**ESTRUCTURAS DE CONTROL SELECTIVAS O ESTRUCTURAS DE DECISIÓN:**

Son las que permiten la selección de acciones alternativas

**EXPRESIÓN LÓGICA:** es una expresión que al ser evaluada da como resultado un valor lógico (verdadero o falso).

Podrían definirse dos tipos de expresiones lógicas, las cuales tienen la siguiente forma:

**Tipo A** (también denominadas expresiones relacionales): Operando1 operador de relación Operando2

**Donde:**

• Operando1 y 2 pueden ser ambos expresiones aritméticas o ambos expresiones carácter.

• Expresiones carácter pueden ser constantes o variables de tipo carácter o cadena.

• Operador de relación: > (mayor), < (menor), = (igual), >= (mayor o igual), <= (menor o Igual), <> (diferente).

Los operadores de relación tienen igual prioridad entre ellos, pero la más baja prioridad en relación con los operadores aritméticos.

Su orden de evaluación es el siguiente:

**Prioridad Operador**

**1. ( )**

**2. \*, /, div, mod**

**3 . +,-**

**4 . >,<,=,>=,<=,<>**

**Nota:**

• Los valores numéricos sólo pueden compararse con otros valores numéricos y los caracteres con otros caracteres.

• Los valores carácter no pueden compararse con las cadenas de caracteres.

• Las cadenas de caracteres sólo pueden compararse si tienen exactamente la misma longitud.

• En las comparaciones entre expresiones carácter el orden, según la codificación ASCII, es el siguiente: ‘0’<’1’<’2’<…..<’9’, ‘a’<’b’<’c’<…..<’z’, ‘A’<’B’<’C’<…. <’Z’

**Ejemplos de expresiones lógicas (también llamadas relacionales)**

a) 6.7315 > 6.7342 (false) b) 15 < 20 (cierto) c) 8 = 8.0 (cierto) d) ‘A’ < ’B’ (cierto)

e) ‘5’ = 5 produce error, f) ‘2’ < ‘235’ produce error

f) Dado N1 = 3, N2 = 6 y CAR = ‘m’, el resultado de la evaluación de las siguientes expresiones es: CAR = ‘m’ (verdadero) (2+3\*N1) <> 6 + (N2 – 1) (falso) (1+ (2div2) mod N2) <= 4\*n1 (verdadero) (1+1 mod N2) <= 4\*N1 (verdadero)

**Tipo B** Las expresiones lógicas pueden combinarse para formar expresiones más complejas: Operando1 operador lógico Operando2

**Donde:**

• Operando1 y 2 son expresiones lógicas que pueden estar constituidas por: constantes lógicas, variables lógicas, expresiones de relación, expresiones lógicas entre paréntesis.

• Operando1 es opcional, dependiendo del operador podría no existir.

• Operador lógico: not (no lógico), and (Y lógico), or (O lógico), xor Mayor prioridad 🡪Menor prioridad

El orden de prioridad de los operadores lógicos en relación con el resto de operadores es el siguiente:

**Operando1 Operando2 not Op1 Op1 and Op2 Op1 or Op2 Op1 xor Op2**

**V V F V V F**

**V F F F V V**

**F V V F V V**

**F F V F F F**

**Ejemplos de expresiones lógicas:**

a) not (‘A’=’Z’) (verdadero), b) not (2<3) (falso), c) not p or q and r

d) ((X\*2 > Y-3) or (X > Y-1)) and (Y < 5)

e) Valor := (resp=’T’) or (resp=’t’) verdadero si resp toma el valor de ‘T’ o ‘t’ de lo contrario es falso

**ESTRUCTURA DE DECISIÓN SIMPLE:** se usan para representar estructuras en las que si la evaluación de la expresión lógica resulta ser verdadera se ejecuta la sentencia1 o la serie de sentencias comprendidas entre begin y end de la instrucción IF-THEN, según sea el caso.

Mientras que si el resultado de su evaluación es falso se continúa como si la instrucción del IFTHEN no hubiese existido.

IF expresión lógica THEN sentencia1 🡨 Para el caso de sentencia simple. Es decir, una sola sentencia

IF expresión lógica THEN 🡨 Para el caso de sentencias compuestas. Es decir, varias sentencias.

BEGIN

Sentencia 1;

sentencia 2;

:

Sentencia N

END;

**ESTRUCTURA DE DECISIÓN DOBLE:** se usan para representar estructuras en las que si la evaluación de la expresión lógica resulta ser verdadera se ejecuta la sentencia1. Mientras que si el resultado de su evaluación es falso se ejecuta la sentencia2. Recuerde que en el caso de sentencias compuestas, en lugar de la sentencia1 o la sentencia2 se ejecutarán las series de sentencias comprendidas entre sus respectivos begin end.

IF expresión lógica THEN 🡨Para el caso de sentencia simple. Es decir, una sola sentencia

sentencia1

ELSE

Sentencia2;

IF expresión lógica THEN 🡨Para el caso de sentencias compuestas. Es decir, varias sentencias.

BEGIN

sentencia1;

sentencia2

END

ELSE

BEGIN

sentencia3;

sentencia4

END;

**ESTRUCTURAS DE DECISIÓN MÚLTIPLE**: existen dos tipos de sentencias, las IF anidadas y las CASE.

**SENTENCIA IF ANIDADA:** se usan para representar estructuras en las que si la evaluación de la expresión lógica resulta ser verdadera se ejecuta la sentencia1. Mientras que si el resultado de su evaluación es falso se evalúa otra expresión lógica, si el resultado de esta última es verdadero se ejecuta la sentencia2, si es falso se evalúa otra expresión y así sucesivamente. Recuerde que en el caso de sentencias compuestas, en lugar de la sentencia1, sentencia2 o sentencia N, se ejecutarán la serie de sentencias comprendidas entre sus respectivos begin - end.

IF expresión lógica1 THEN

sentencia1

ELSE IF expresión lógica2 THEN

sentencia2

:

ELSE IF expression lógica N THEN

Sentencia N

ELSE

SentenciaN+1;

**SENTENCIA CASE:** se usa para elegir entre diferentes alternativas. Puede utilizarse de cualquiera de las dos maneras descritas a continuación, según sea el caso

CASE selector OF

lista constantes1: sentencia1;

lista constantes2: sentencia2;

:

lista de constantesN: sententiaN

END;

CASE selector OF

lista constantes1: sentencia1;

lista constantes2: sentencia2;

:

lista de constantesN: sententiaN

ELSE

SentenciaN+1

END;

**Donde:**

• Lista de constantes es igual a uno o más posibles valores del selector separados por coma y pueden ser constantes o un valor subrango: constante1..constante2.

• Cada sentencia puede ser simple o compuesta (recordar que las compuestas deben ir entre un begin y un end).

• El selector es una expresión que debe ser de tipo ordinal (integer, char, bolean, subrango o enumerado). No son válidos los tipo real, string, longint, word.

• Todas las constantes CASE deben ser únicas y de un tipo ordinal compatible con el tipo del selector.

• Cada sentencia excepto la última, debe ir seguida de ;

• La cláusula ELSE es opcional