

## 4.1.- EXPRESIONES LOGICAS

Una expresión lógica únicamente puede tomar dos valores que son :

**.true.** o **.false.**

Este tipo de expresiones permiten crear *estructuras de control* dentro del programa FORTRAN que dependiendo de los valores obtenidos por la expresión se bifurcan y recorren caminos alternativos.

A la hora de construir una expresión lógica FORTRAN cuenta con una serie de **operadores lógicos** que se aplican siempre entre dos elementos del mismo tipo, bien numéricos o bien tipo carácter. Una vez construida la expresión esta únicamente puede tomar dos valores cierto o falso.

Los operadores en FORTRAN son los siguientes:

OPERADOR	SIGNIFICADO
<b>.EQ.</b>	<i>Igual que ( = )</i>
<b>.NE.</b>	<i>No igual que ( <sup>1</sup> )</i>
<b>.LT.</b>	<i>Menor que ( &lt; )</i>
<b>.LE.</b>	<i>Menor o igual que ( £ )</i>
<b>.GT.</b>	<i>Mayor que ( &gt; )</i>
<b>.GE.</b>	<i>Mayor o igual que ( <sup>3</sup> )</i>

### Ejemplo

Supongamos que una variable real **A** posee un valor de 5.6 y otra variable real **B** posee un valor de 7.8, la siguiente tabla muestra los valores de lógicos para una serie de expresiones lógicas:

EXPRESION	VALOR
(A .gt. 50.0)	<b>.false.</b>
(A .ge. B)	<b>.false.</b>
(A.ne.B)	<b>.true.</b>

Por otro lado, hay otro tipo de **operadores lógicos** que son utilizados únicamente entre expresiones lógicas, en concreto **.NOT.** opera sobre una sola expresión lógica y el resto sobre dos. La siguiente tabla muestra este tipo de operadores y sus significado.

OPERADOR	SIGNIFICADO
<b>.NOT.</b>	<i>Cambia el valor de la expresión lógica a su opuesto</i>
<b>.AND.</b>	<i>Cierto únicamente si ambas expresiones lógicas son ciertas</i>
<b>.OR.</b>	<i>Cierto si una de las expresiones es cierta</i>
<b>.XOR.</b>	<i>Cierto únicamente si una de las expresiones lógicas es cierto</i>
<b>.EQV.</b>	<i>Cierto si ambas expresiones tiene el mismo valor</i>
<b>.NEQV.</b>	<i>Cierto si ambas expresiones no tienen el mismo valor</i>

### Ejemplo

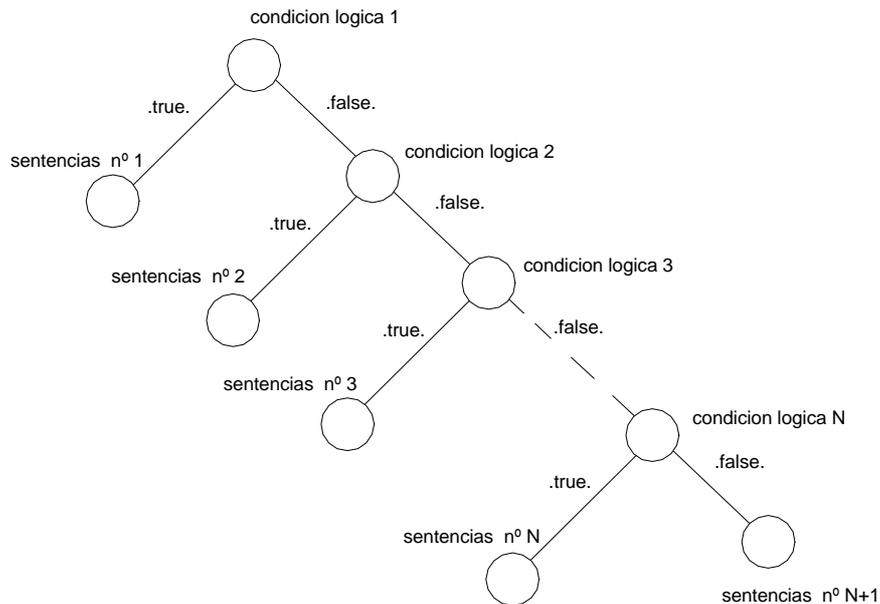
Supongamos dos variables con valores  $A=2.0$  y  $B=10.0$ , entonces:

**.not. (a .lt. b)**                    .false.  
**(a .ne. b) .and. (a .lt. b)**        .true. (ambas expresiones lógicas son ciertas)  
**(a .lt. b) .or. (a .gt. b)**            .true. (una de las expresiones lógicas es cierta)  
**(a .ne. b) .xor. (a .lt. b)**            .true. (ambas expresiones son ciertas)  
**(a .lt. b) .eqv. (a .gt. b)**            .false. (los resultados de las expresiones no son iguales)  
**(a .lt. b) .neqv. (a .gt. b)**          .true. (los resultados de las expresiones no son iguales)







**EJERCICIO 4.1**

- Realizar un programa que lea un numero por pantalla y comunique al usuario si es par o impar.

**EJERCICIO 4.2**

- Realizar un programa que lea un valor entero comprendido entre 1 y 5 y presente por pantalla la vocal a,e,i,o,u en función del numero leído. El programa deberá advertir al usuario en caso de que el valor introducido no este en el intervalo [1,5].

**EJERCICIO 4.3**

- Realizar un programa que lea una letra del abecedario y diga al usuario si se trata de una vocal o no.

**EJERCICIO 4.4**

- Programar la ecuación de 2º grado  $Ax^2 + Bx + C = 0$ , con la capacidad de dar como resultado las raices complejas.

**EJERCICIO 4.5**

- Realizar un programa que calcule el modulo y el argumento de un numero complejo  $(a+bi)$ , poniendo especial cuidado en la ubicación del argumento resultante en el cuadrante correcto.