ESTRUCTURA DE SELECCIÓN O DECISIÓN

SELECCIÓN SIMPLE

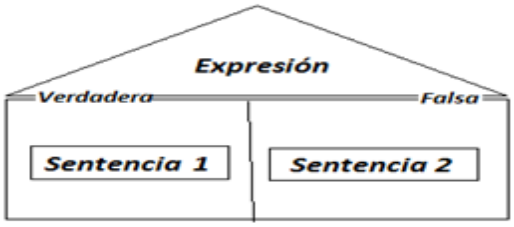
Esta estructura permite realizar la selección entre dos posibles cursos de acción, en base a la verdad o falsedad de una expresión que se escribe en forma de "expresión lógica".

Para escribir una expresión lógica es necesario utilizar “operadores de comparación”. Estos operadores permiten evaluar una condición entre dos partes, ya sean dos variables, o una variable y una constante.

Operadores de comparación

Operador	Símbolo
Mayor	>
Menor	<
Mayor o igual	>=
Menor o igual	<=
Igual	==
Distinto	!=

La representación gráfica de la estructura de selección simple es la siguiente:



Por ejemplo, dada una variable **num** numérica, se puede escribir como expresión **num** > 6 en cuyo caso será verdadera ó falsa según el resultado de la evaluación del contenido ingresado en **num**. Si es verdadera, se ejecuta la “Sentencia 1”, y si es falsa se ejecuta la “Sentencia 2”. Dentro del verdadero y/o falso puede haber una o más sentencias.

El lenguaje C admite también que se utilice una expresión aritmética, tomando el resultado de la evaluación como “falso” si es igual a cero y verdadero por distinto de cero.

Ejemplo: si se evalúa 5-6+3 esta expresión devuelve un valor distinto de cero, por lo cual, continua la secuencia por verdadero. En cambio, al evaluar 4+3-7 esta expresión devuelve como resultado un valor igual a cero, por lo que la secuencia continua por el lado falso.

Se puede utilizar, si es que lo desea, solo la parte verdadera, o sea que las acciones por 'falso' son opcionales. Las expresiones que se utilizarán al principio serán simples, es decir, una relación entre dos valores y luego se ampliarán utilizando los operadores lógicos.

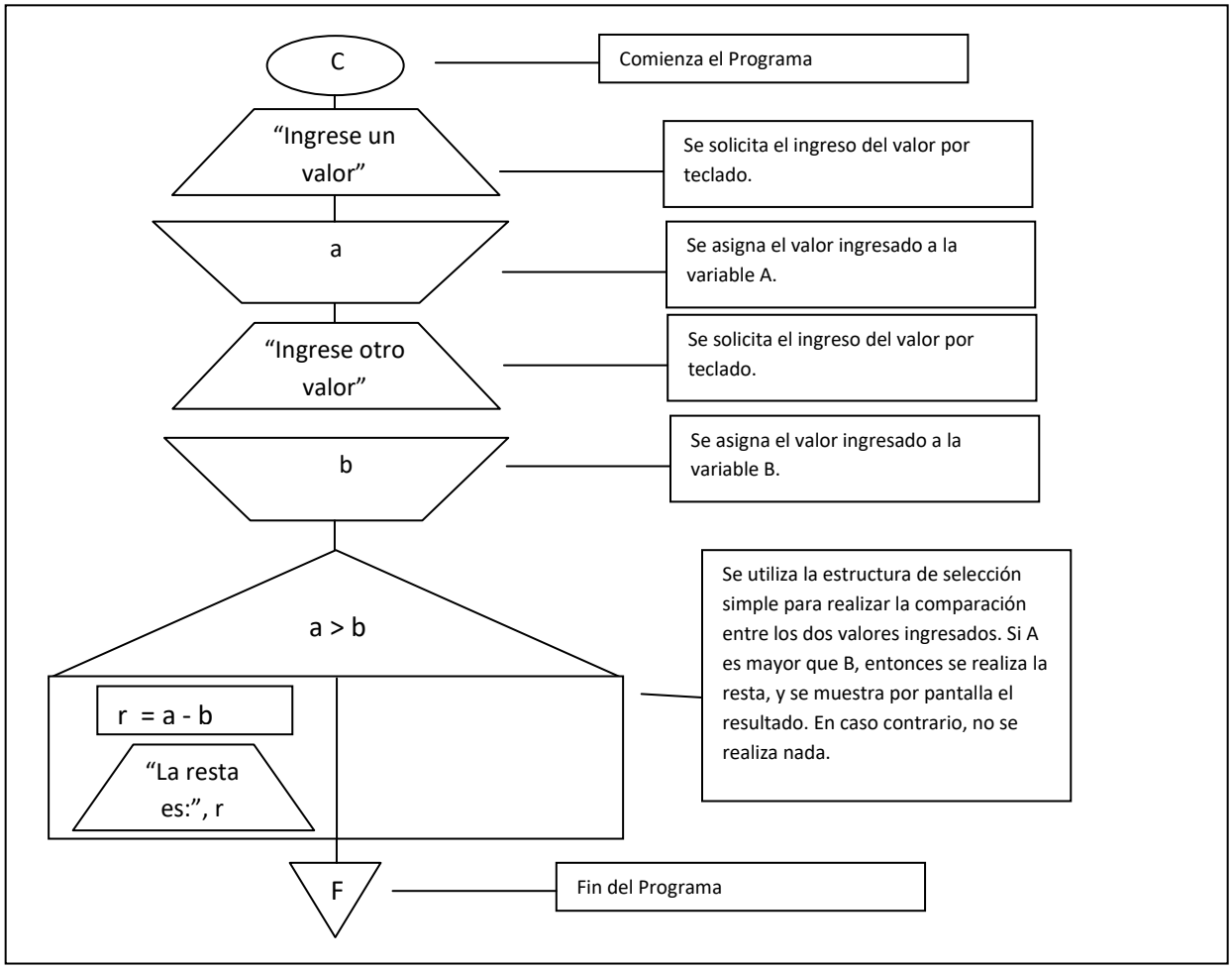
Estas estructuras se pueden enlazar en forma 'descendente', es decir, que dentro del verdadero y/o falso puede haber otras estructuras condicionales. Esto se verá en otros ejercicios más adelante. Esta estructura tiene un solo punto de entrada y uno solo de salida cualquiera fuese la alternativa ejecutada.

Ejemplo 1: Ingresar dos valores enteros en dos variables, que se llamaran **a** y **b**. Si **a** es mayor a **b** calcular e informar la diferencia entre **a** y **b**.

ESTRATEGIA

1. Ingresar los dos valores en **a** y **b**
2. Si **a** es mayor que **b** calcular la diferencia
3. Informar la diferencia calculada si se cumplió la condición del punto 2.

DIAGRAMA:


C
Comienza el Programa
"Ingrese un valor"
Se solicita el ingreso del valor por teclado.
a
Se asigna el valor ingresado a la variable A.
"Ingrese otro valor"
Se solicita el ingreso del valor por teclado.
b
Se asigna el valor ingresado a la variable B.
a > b
r = a - b
"La resta es:", r
Se utiliza la estructura de selección simple para realizar la comparación entre los dos valores ingresados. Si A es mayor que B, entonces se realiza la resta, y se muestra por pantalla el resultado. En caso contrario, no se realiza nada.
F
Fin del Programa

Ejemplo 2: Confeccionar un programa que ingrese dos valores numéricos y determine e informe al mayor de ellos si son distintos, o un mensaje que diga IGUALES en caso de serlo.

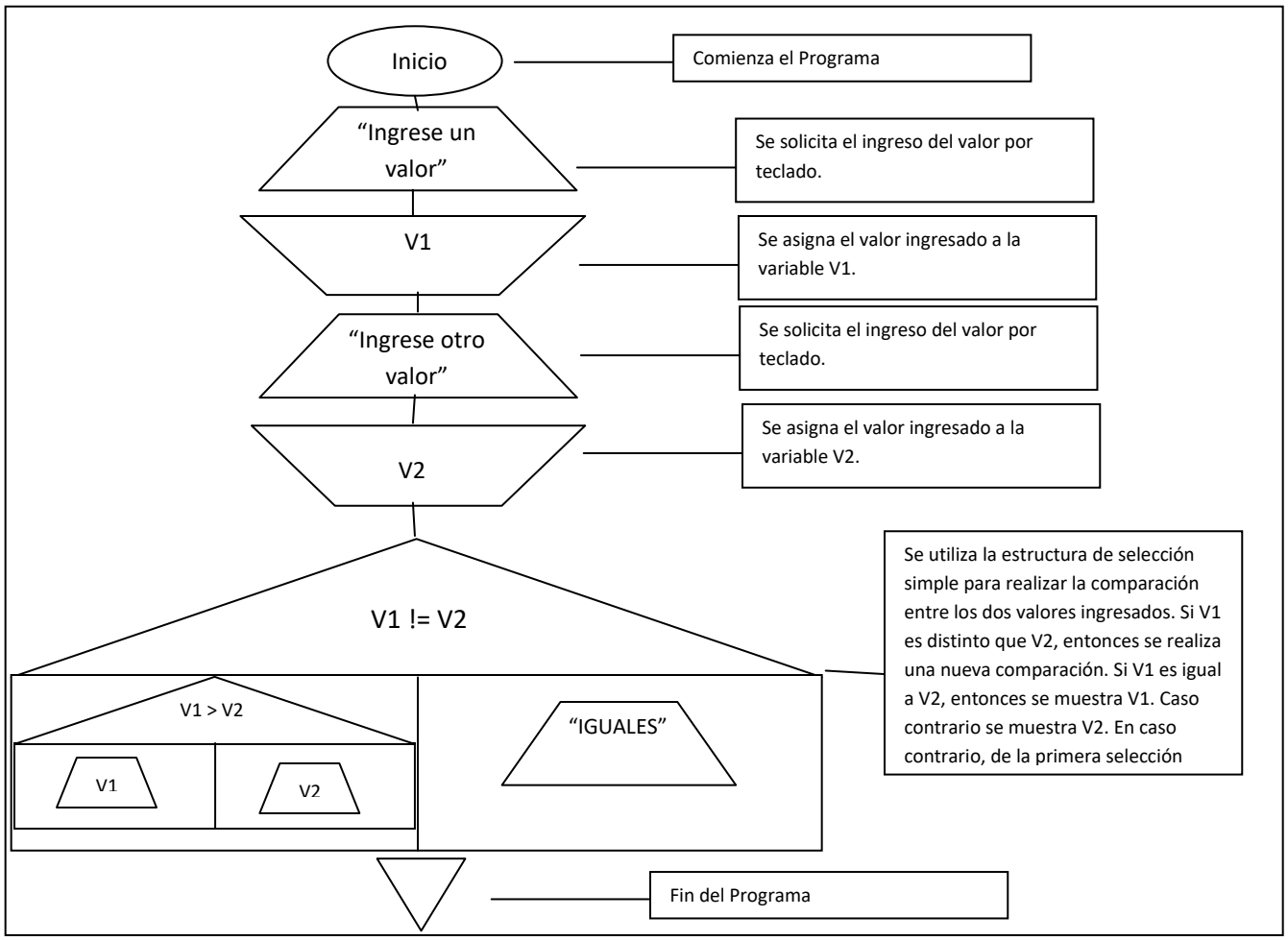
ESTRATEGIA

1] Ingresar los dos valores

2] Si no son iguales, determino el mayor

3] Informo el mayor ó el mensaje IGUALES

DIAGRAMA :



Operadores Lógicos

Los operadores lógicos permiten asociar varias expresiones lógicas ó aritméticas formando solo una, lo cual simplificará la cantidad estructuras de selección que se deben utilizar.

Los operadores lógicos son 3:

- AND (y)
- OR (ó)
- NOT (negación)

Operadores lógicos en el lenguaje C y C++

Operador	Símbolo
AND	&&
OR	||
NOT	!

Estos operadores trabajan sobre valores lógicos, por lo tanto, pueden aplicarse sobre condiciones o sobre valores numéricos de una variable recordando que el 0 es falso y cualquier valor distinto de cero es verdadero.

El operador AND (&&) evalúa la dos condiciones y retorna verdadero solo en el caso de que ambas condiciones sean verdaderas, es decir, se deben cumplir ambas al mismo tiempo.

El operador OR (||) evalúa las dos condiciones y retorna verdadero cuando al menos una de las dos es verdadera, es decir, se debe cumplir al menos una de las condiciones para que sea verdadera la expresión.

El operador NOT (!) trabaja sobre una sola condición negándola. Es decir, que si es verdadera la hace falsa y si es falsa la hace verdadera.

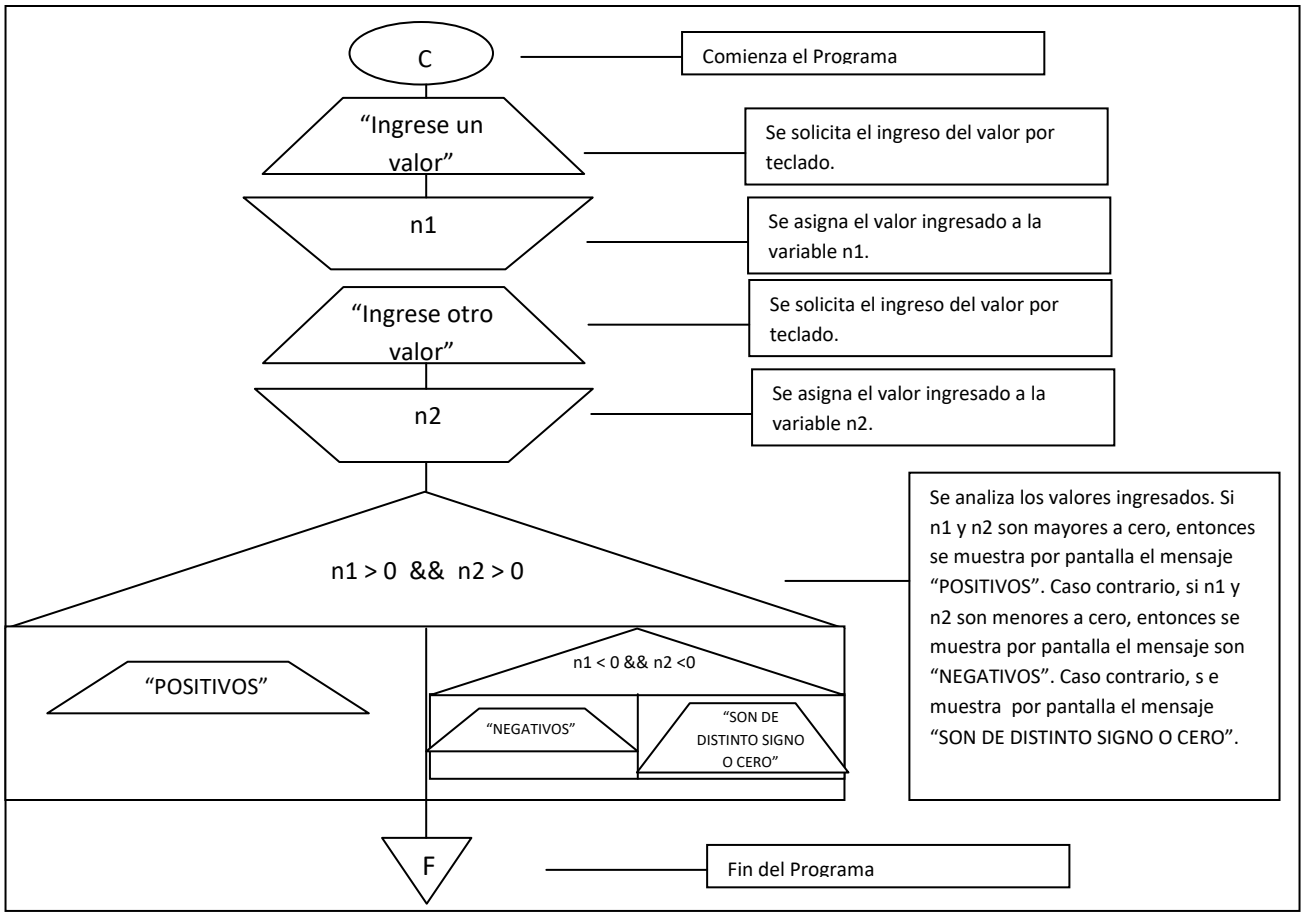
A continuación se muestra la tabla de verdad de los operadores lógicos, siendo A y B condiciones lógicas.

Tabla de verdad de los operadores lógicos

A	B	A && B	A || B	!A
F	F	F	F	V
F	V	F	V	V
V	F	F	V	F
V	V	V	V	F

Las precedencias de los operadores son: NOT (!) - AND (&&) - OR (||)

Ejemplo 3: Ingresar dos números. Indicar si ambos son positivos, ambos son negativos, o son distinto signo o cero.

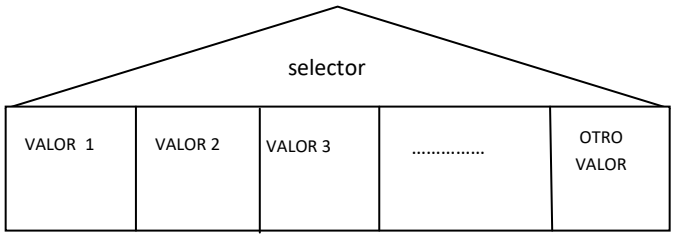


Esta es solo una de las formas de resolver este ejercicio ya que otra persona podría comenzar preguntando por ejemplo si ambos son negativos, o si son cero o de distinto signo. Recordemos que siempre hay varias formas para resolver un mismo ejercicio.

SELECCIÓN MÚLTIPLE

Piense como sería un programa que ingresando el número de un mes tiene que informar el nombre. Utilizando la sentencia if se tienen que anidar las condiciones preguntado si el mes es 1 mostrar enero, sino preguntar si el mes es dos mostrar febrero, sino preguntar si el mes es 3, y así sucesivamente, hasta abarcar todas las condiciones. Para casos de este tipo existe una "estructura condicional múltiple".

Su diagrama y codificación son los siguientes:




```
switch (selector)
{
	case valor1:
			//instrucciones a ejecutar si selector toma el valor1
			break;
	case valor2:
			//instrucciones a ejecutar si selector toma el valor1
			break;
	case valor3:
			//instrucciones a ejecutar si selector toma el valor1
			break;

	….

	default: //instrucciones a ejecutar si selector no toma ninguno de los valores anteriores
}
```

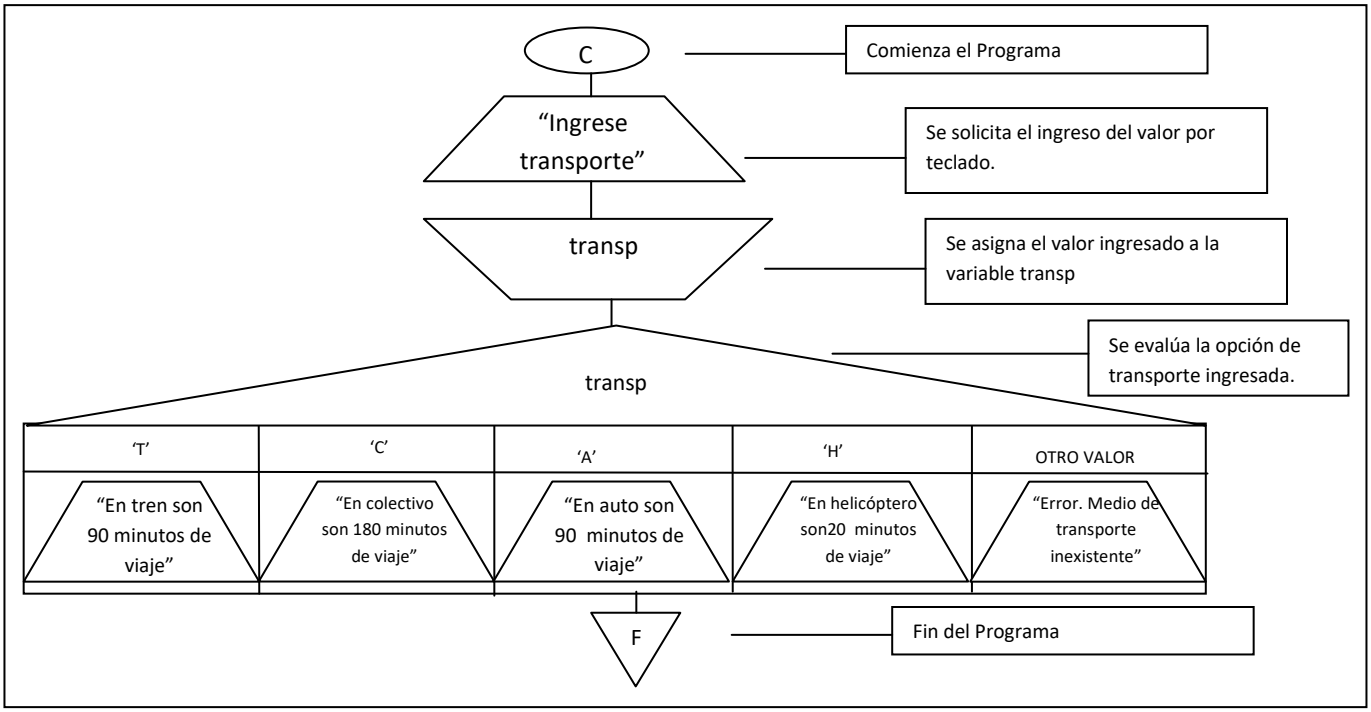
Consideraciones:

- Se evalúa el selector y si su valor coincide con alguno de los casos establecidos se pasa directamente a ejecutar el código de dicho caso
- El default es el valor por defecto, es opcional y se ejecuta si selector no toma ninguna de los valores anteriores.
- Selector puede ser una variable o una expresión pero debe ser del tipo **entera** o **carácter**.
- Los valores deben ser constantes (numéricas o caracteres). **NO pueden ser rangos**. Si se necesita evaluar rangos se debe utilizar la sentencia if.
- Cada caso termina con la instrucción break que hace que la ejecución se corte y salga de la estructura del switch. Si no se coloca el break en un caso seguirá ejecutando el código de los casos que están debajo hasta encontrar un break o llegar a la llave de fin. Es por esto que el default, o si el default no está, el último caso no necesita de la instrucción break. Esta particularidad permite que distintos casos compartan el mismo código.

Los casos no tiene necesidad de estar ordenados, pero cada valor solo puede aparecer una vez.

Ejemplo 4: Confeccionar un programa que solicite el medio de transporte para llegar a un determinado lugar, mostrando en pantalla el tiempo estimado de viaje para cada medio de transporte. El medio de transporte se ingresa con una letra siendo:

- T: tren, tiempo estimado de viaje 90 min.
- C: colectivo, tiempo estimado de viaje 180min
- A: auto, tiempo estimado de viaje, 90 min.
- H: helicóptero, tiempo estimado de viaje, 20 min.



Nótese que los tiempos en tren y auto son los mismos.

Entonces el algoritmo anterior se puede mejorar de la siguiente manera.