

## 2.4 Estructuras de control de tipo condicional.

Las sentencias de un programa son ejecutadas por el ordenador según su flujo natural, es decir, de arriba hacia abajo y de forma consecutiva a no ser que, de alguna manera, se altere este orden. Los comandos que realizan esta función se denominan comandos de control de flujo y las sentencias a las que pertenecen sentencias de control de flujo.

Las sentencias condicionales permiten realizar ciertas instrucciones del programa sólo si se cumple la condición asociada a ellas.

### – Sentencia `if` simple

La forma básica de una sentencia condicional `if` es la siguiente:

```
if      expresión_lógica
      sentencias
end
```

o bien

```
if expresión_lógica , sentencias , end
```

Las sentencias se ejecutan sólo si la expresión lógica se evalúa como cierta, en el caso de que la expresión sea evaluada como falsa, el programa continúa por la instrucción siguiente a `end` sin realizar las sentencias asociadas al condicional.

### – Bloque `if` unicondicional

Se trata de otro tipo de sentencia más compleja pero en la que sigue apareciendo una única expresión lógica. La sintaxis se indica a continuación.

```
if  expresión_lógica
    bloque 1 sentencias
else
    bloque 2 sentencias
end
```

Si la expresión lógica se evalúa como cierta se ejecuta el bloque 1 de sentencias, si se evalúa como falsa se ejecuta el bloque 2. Nunca se ejecutan los dos bloques de sentencias a la vez, ni ninguno de los dos. Se utiliza para elegir uno de dos '*caminos*' en el programa.

– Bloque `if` multicondicional

A continuación mostramos la sintaxis de otra expresión condicional en la que se pueden evaluar dos o más condiciones.

```
if  expresión_lógica_1
    bloque 1 sentencias
elseif  expresión_lógica_2
    bloque 2 sentencias
    .....
elseif expresión_lógica_n
    bloque n sentencias
else
    bloque n+1 sentencias
end
```

Su funcionamiento es simple, la primera expresión lógica que se evalúe como cierta provoca la ejecución de sus sentencias asociadas y el posterior abandono de la estructura completa. Se utiliza para elegir uno de varios '*caminos*' en el programa.

La última rama (`else`) es opcional. Si se escribe, sólo se ejecuta el bloque perteneciente a ella si todas las expresiones lógicas anteriores han resultado falsas.

Veamos unos ejemplos de estas estructuras.

En el siguiente programa se estudia si un número *n* es múltiplo de 2 y/o de 3.

```
n=input ('Introduce un entero');
if  rem(n,2)== 0
    disp('Es múltiplo de 2')
end
if  rem(n,3)== 0
    disp('Es múltiplo de 3')
end
```

Obsérvese que las dos condiciones a estudiar son independientes: el cumplimiento o no de la primera no induce a que la segunda se cumpla o no. Cuando ocurra esto se deben estudiar los casos de forma independiente usando `if` simples.

En el siguiente programa, se elige entre una de dos alternativas posibles (par o impar). Un número es par o es impar, no puede ser las dos opciones a la vez ni ninguna de ellas. Cuando ocurran estos casos se debe utilizar la estructura `if/else`.

```
n=input ('Introduce un entero positivo');  
  
if    rem(n,2)== 0  
    disp('n es par')  
else  
    disp('n es impar')  
end
```

A continuación se incorpora al programa anterior una tercera opción, número negativo, que aumenta a tres las posibilidades de elección. Cuando tengamos más de dos posibilidades excluyentes se debe usar la estructura `if/elseif/else`.

```
n=input ('Introduce un entero');  
  
if    n<0  
    disp('n es negativo')  
elseif    rem(n,2)== 0  
    disp('n es par')  
else  
    disp('n es impar')  
end
```