

## SENTENCIAS DE CONTROL

### Sentencias

- **Sentencia** = toda orden ejecutable que compone un programa
  - **Sentencia simple** = delimitada por el punto y coma de la anterior sentencia y el punto y coma final
  - **Sentencia compuesta** = delimitada por la palabra reservada BEGIN y la palabra reservada END.

Entre ambas pueden existir sentencias simples u otras sentencias compuestas.

Se tratan como una única sentencia.

### Sentencias de control selectivas

- **Sentencia selectiva**

```
IF < expresión condicional> THEN
    <sentencia>
[ELSE
    <sentencia>];
```

Cualquiera de las sentencias que se ejecutan en cualquiera de las dos alternativas puede ser a su vez una sentencia de control.

```
IF Num > 5 THEN {siendo Num de tipo INTEGER}
BEGIN
    IF Num < 10 THEN
        Writeln (num, 'está entre 5 y 10')
    ELSE
        Writeln (num, 'es mayor que 10:');
    WRITELN ('Selección finalizada');
END
ELSE
```

```
WRITELN('El número es menor que cinco');
```

- **Selección múltiple**

```
CASE <expresión ordinal> OF
  {la expresión debe evaluarse a un integer o un char}
  <caso1>: <sentencial>;
  <caso2>: <sentencia2>;
  <caso3>: <sentencia3>;
  ...
  [ELSE
    <sentencia else>;]
END;
```

donde

```
<caso>= <constante> | <rango> | <lista>
<rango>= <constante>..<constante>
<lista>= <constante>,<constante>,...
```

Ejemplo:

```
CASE Ch OF
  'A'..'Z', 'a'..'z': WriteLn('Letra');
  '0'..'9': WriteLn('Número');
  '+', '-', '*', '/': WriteLn('Operador');
  ELSE
    WriteLn('Carácter especial');
END;
```

## Sentencias de control repetitivas

- **Sentencia FOR** = Se itera cero, una o más veces.

```
FOR <variable>:=<expr_inicio> {TO|DOWNTO} <expr_fin> DO
  <sentencia>;
```

Ejemplo:

```
PROGRAM Ciclo_FOR;
VAR
    Numero : Integer;
BEGIN
    FOR Numero := 1 TO 50 DO
        WRITELN(Numero);
    END.
```

```
PROGRAM Ciclo_FORbis;
VAR
    Numero : Integer;
BEGIN
    FOR Numero := 50 DOWNTO 1 DO
        WRITELN(Numero);
    END.
```

- **Sentencia WHILE** = Se itera cero, una o muchas veces (incluso infinito)

```
WHILE <expresión condicional> DO
    <sentencia>;
```

Ejemplo:

```
PROGRAM Ciclo_WHILE;
VAR
    Numero : Integer;
BEGIN
    Numero := 1;
    WHILE Numero <= 50 DO
        BEGIN
            WRITELN (Numero);
            Numero := Numero +1;
        END;
    END.
```

- **Sentencia REPEAT** = Se itera una o muchas veces (incluso infinito)

```
REPEAT
    <lista de sentencias>
UNTIL <expresión condicional>;
```

#### Ejemplo

```
PROGRAM Ciclo_RepeatUntil;
VAR
    Numero : Integer;
BEGIN
    Numero := 1;
    REPEAT
        WRITELN (Numero);
        Numero := Numero + 1;
    UNTIL Numero > 50;
END
```