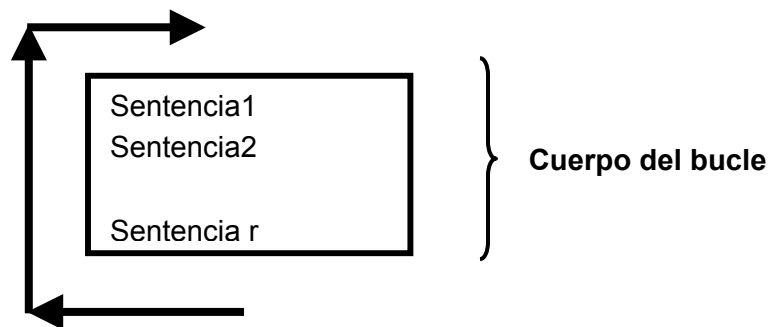


Recuerde que estos son apuntes muy simplificados que deberá completar con la bibliografía recomendada

APUNTES Nº 6

ESTRUCTURAS DE CONTROL REPETITIVAS: son aquellas que permiten que una o varias sentencias se ejecuten repetidamente.

BUCLE: es la estructura de control que permite la repetición de una serie determinada de sentencias. Es importante tener en cuenta cuántas veces se repite el bucle y cuál es el cuerpo del bucle. El cuerpo del bucle lo constituyen la serie de sentencias que pueden ser de cualquier tipo (secuenciales, de decisión o repetitivas) las cuales serán repetidas **n** veces, pudiendo ser **n** conocido o desconocido. Cuando **n** es desconocido, el número de veces que debe repetirse el cuerpo del bucle estará condicionado por una expresión lógica.



ESTRUCTURA DE CONTROL REPETITIVA WHILE: se usa cuando no se conoce el número de iteraciones. Mediante esta estructura el cuerpo del bucle se **repite mientras** se cumpla una determinada condición (en otras palabras, mientras el resultado de la evaluación de la expresión lógica sea verdadero).

La condición se evalúa siempre al principio de cada iteración. Esto implica que, si el resultado de su evaluación es falso desde el comienzo de la estructura, el bucle no se ejecuta nunca.

WHILE expresión lógica **DO**
Sentencia;

Para el caso de sentencia simple. Es decir, una sola sentencia representa el cuerpo del bucle

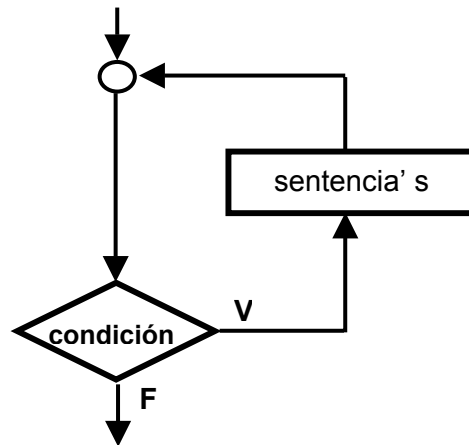
WHILE expresión lógica **DO**
BEGIN
Sentencia1;
Sentencia2;

SentenciaN
END;

Para el caso de sentencias compuestas. Es decir, varias sentencias representan el cuerpo del bucle.

recuerde que antes de un **END** no va ;

El algoritmo gráfico de una **estructura de control repetitiva WHILE** se representa de la siguiente manera:



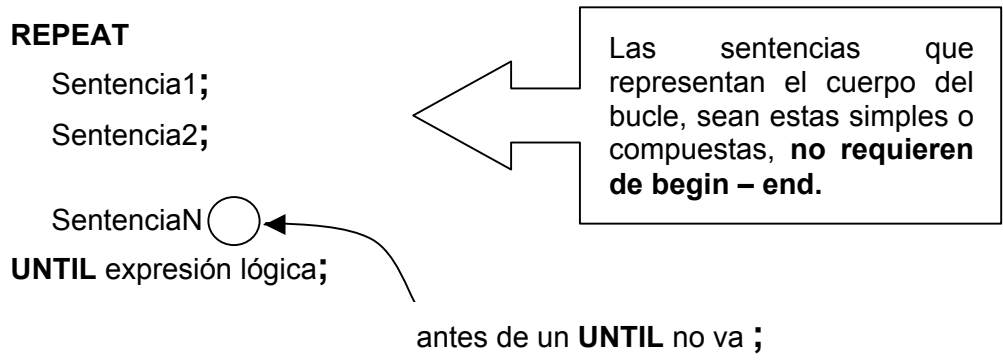
Nota: condición es igual a expresión lógica

El algoritmo escrito: **Nº. Repetir mientras (condición)**
Nº.Nº sentencia's

Otra forma válida también es: **Nº. mientras (condición) haga**
Nº.Nº sentencia's

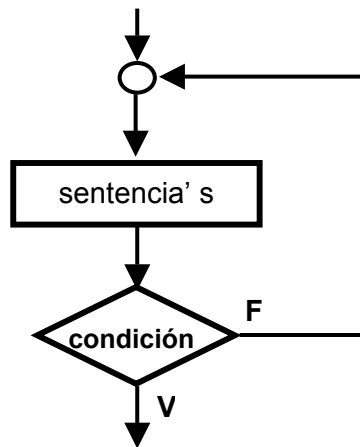
ESTRUCTURA DE CONTROL REPETITIVA REPEAT: se usa cuando no se conoce el número de iteraciones. Mediante esta estructura el cuerpo del bucle se **repite hasta que** la condición se hace verdadera.

La condición se evalúa después de que el cuerpo del bucle se ha ejecutado al menos una vez.



NOTA: los bucles **WHILE** y **REPEAT** son complementarios en su actuación.

El algoritmo gráfico de una **estructura de control repetitiva REPEAT** se representa de la siguiente manera:



Nota: condición es igual a expresión lógica

El algoritmo escrito: **Nº. Repetir**
 Nº.Nº sentencia's
 Hasta_que (sondición)

ESTRUCTURA DE CONTROL REPETITIVA FOR: se usa cuando se desea ejecutar un bucle un número determinado de veces, cuyo número se conoce por anticipado. En la ejecución del FOR participan tres elementos **V** : variable de control del bucle, **Vi** : valor inicial, **Vf** : valor final. Existen dos formas de utilizar las estructuras de control repetitivas:

a) Con **INCREMENTO DEL CONTADOR (Vi < Vf)**:

FOR V:= Vi TO Vf DO
 Sentencia;

Para el caso de sentencia simple. Es decir, una sola sentencia representa el cuerpo del bucle

FOR V:= Vi TO Vf DO
 BEGIN
 Sentencia1;
 Sentencia2;
 SentenciaN
 END;

Para el caso de sentencias compuestas. Es decir, varias sentencias representan el cuerpo del bucle.

antes del **END** no va ;

Cuando **Vi** es menor que **Vf** el valor de **V** se incrementa de uno en uno desde el valor de **Vi** hasta el valor de **Vf**. Por cada valor que **V** toma, el cuerpo del bucle se ejecuta una vez.

b) Con **DECREMENTO DEL CONTADOR** ($V_i > V_f$):

```
FOR V:= Vi DOWNTO Vf DO  
  Sentencia;
```

Para el caso de sentencia simple. Es decir, una sólo sentencia representa el cuerpo del bucle

```
FOR V:= Vi DOWNTO Vf DO  
  BEGIN  
    Sentencia1;  
    Sentencia2;  
  
    SentenciaN  
  END;
```

Para el caso de sentencias compuestas. Es decir, varias sentencias representan el cuerpo del bucle.

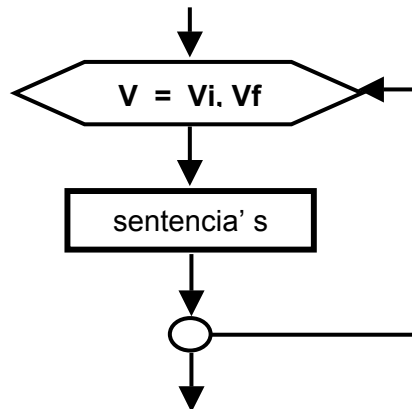
antes del **END** no va ;

Cuando V_i es mayor que V_f el valor de V se decrementa de uno en uno desde el valor de V_i hasta el valor de V_f . Por cada valor que V toma, el cuerpo del bucle se ejecuta una vez.

NOTA válida para ambas formas:

- V debe ser de tipo ordinal(entero, carácter, lógico u otro tipo ordinal) o subrango.
- V , V_i , V_f deben ser todas del mismo tipo, pero el tipo real no está permitido.
- V_i , V_f pueden ser expresiones o constantes.
- No se debe modificar el valor de V , V_i , V_f dentro del cuerpo del bucle.
- El valor de V queda indefinido cuando se termina el bucle.

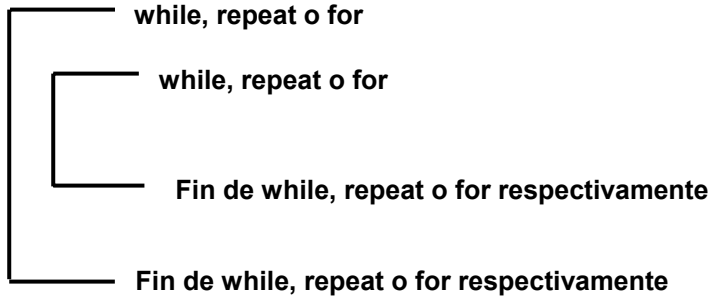
El algoritmo gráfico de una **estructura de control repetitiva FOR** se representa de la siguiente manera:



El algoritmo escrito:

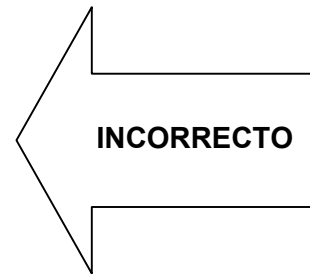
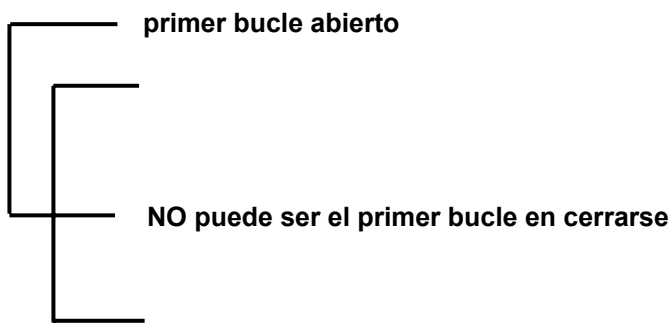
**Nº. Desde $V = V_i$ hasta V_f hacer
Nº.Nº sentencia's**

BUCLES ANIDADOS: son denominados todos aquellos bucles que estén contenidos dentro de otro bucle. Cuando se anidan bucles, se debe tener cuidado que el bucle interior esté contenido completamente dentro del bucle exterior. Todos los tipos de bucles pueden anidarse, sea entre sí o entre cada uno de éstos (**while**, **repeat** o **for**).



El primer bucle que se abra debe ser el último en cerrarse. Lo que es equivalente a decir, que el último bucle en abrirse debe ser el primero en cerrarse y así tantas veces como sea necesario.

Si los bucles se cruzan no serán válidos. Es decir, es incorrecto si los bucles anidados se construyen de la siguiente forma:



Ver ejemplos y demás explicaciones en clase

Los conceptos anteriores son tomados básicamente de:

- Programación en Turbo/ Borland. Pascal 7. Luis Joyanes Aguilar
- Programación con Lenguaje Turbo Pascal. F.J. Sanchis Llorca.-