

Estructuras Lógicas de Control

Índice

1. Secuencia.....	2
2. Condicional.....	2
2.1. Condicional Simple (IF).....	2
2.2. Condicional Doble (IF – ELSE).....	3
2.3. Condicional Múltiple (IF – ELSEIF – ELSE).....	3
3. FOR.....	4
4. WHILE.....	5
5. DO – WHILE.....	5
6. RUPTURA DE CICLOS DE REPETICIÓN BREAK y CONTINUE.....	6
7. SWITCH.....	7

1. Secuencia

Es aquella en la que una acción (instrucción) sigue a la otra en el orden en el que están escritas.

Su representación y el diagrama de flujo se muestran en las siguientes figuras:



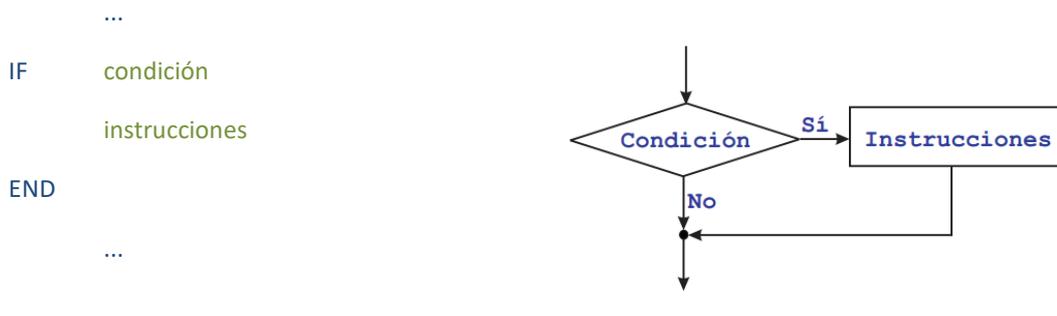
2. Condicional

Permiten que se ejecuten conjuntos distintos de instrucciones, en función de que se verifique o no determinada condición.

2.1. Condicional Simple (IF)

Este es el tipo más sencillo de estructura condicional. Sirve para implementar acciones condicionales del tipo siguiente:

- Si se verifica una determinada condición, se ejecuta una serie de instrucciones y luego se sigue adelante.
- Si la condición NO se cumple, NO se ejecutan dichas instrucciones y se sigue adelante.



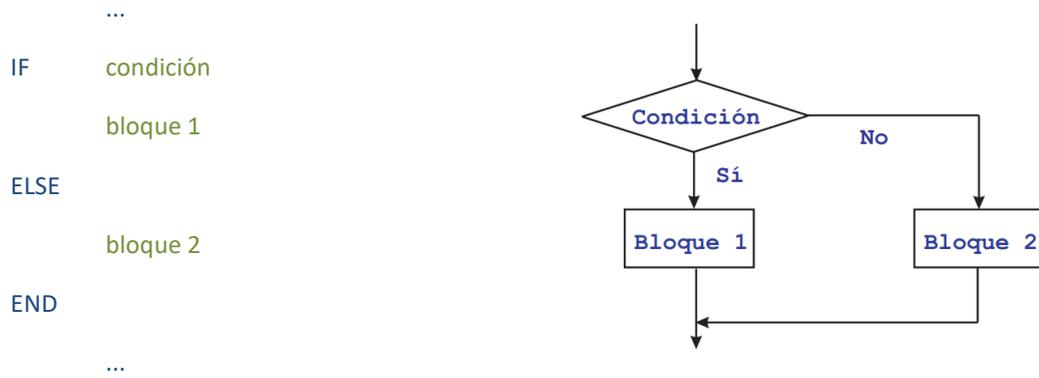
Obsérvese que, en ambos casos (se verifique o no la condición), los “camino” bifurcados se unen posteriormente en un punto, es decir, el flujo del programa recupera su carácter secuencial, y se continúa ejecutando por la instrucción siguiente a la estructura IF.

2.2. Condicional Doble (IF - ELSE)

Este tipo de estructura permite implementar condicionales en los que hay dos acciones alternativas:

- Si se verifica una determinada condición, se ejecuta una serie de instrucciones (bloque 1).
- Si no, esto es, si la condición NO se verifica, se ejecuta otra serie de instrucciones (bloque 2).

En otras palabras, en este tipo de estructuras hay una alternativa: se hace una cosa o se hace la otra. En ambos casos, se sigue por la instrucción siguiente a la estructura IF - ELSE.

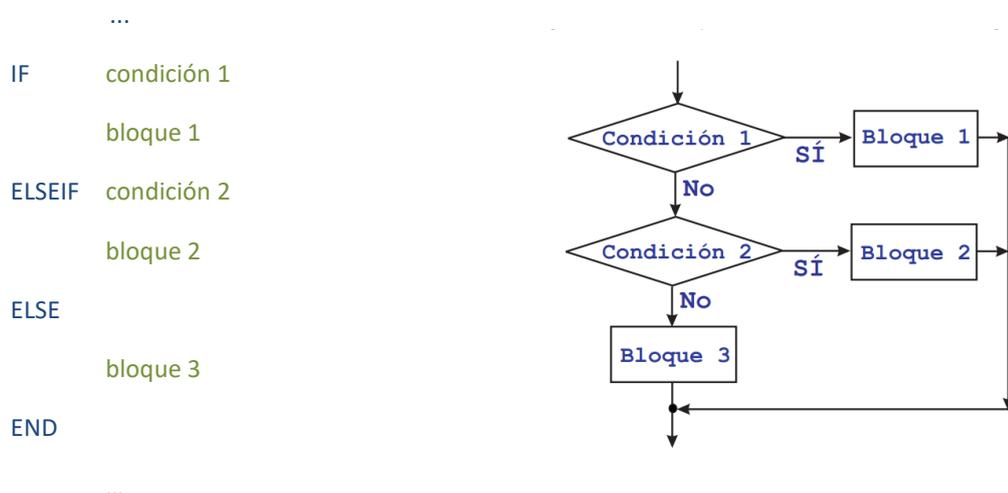


2.3. Condicional Múltiple (IF - ELSEIF - ELSE)

En su forma más general, la estructura IF - ELSEIF - ELSE permite implementar condicionales más complicados, en los que se “encadenan” condiciones en la forma siguiente:

- Si se verifica la condición 1, se ejecutan las instrucciones del bloque 1.
- Si no se verifica la condición 1, pero SÍ se verifica la condición 2, se ejecutan las instrucciones del bloque 2.
- Si no, esto es, si no se ha verificado ninguna de las condiciones anteriores, se ejecutan las instrucciones del bloque 3.

En cualquiera de los casos, el flujo del programa continúa por la instrucción siguiente a la estructura IF - ELSEIF - ELSE. La sintaxis habitual y el diagrama de flujo, se muestran en las figuras a continuación:



En la estructura IF - ELSEIF - ELSE se puede multiplicar la cláusula ELSEIF, obteniéndose así una “cascada” de condiciones, como se muestra en el organigrama, cuyo funcionamiento es claro. En este tipo de estructura condicional, la cláusula ELSE, junto con su bloque de instrucciones, puede no estar presente.

```

IF      condición 1
        instrucciones 1
ELSEIF  condición 2
        instrucciones 2
ELSEIF  condición 3
        instrucciones 3
...
ELSE
        instrucciones k
END

```

Las distintas estructuras condicionales descritas pueden ser anidadas, es decir, puede incluirse una estructura IF (de cualquier tipo), como parte de las instrucciones que forman el bloque de uno de los casos de otro IF. Como es lógico, no puede haber solapamiento. Cada estructura IF debe tener su propio fin (END).

3. FOR

Este tipo de estructura permite implementar la repetición de un cierto conjunto de instrucciones un número predeterminado de veces.

Para ello se utiliza una variable de control del bucle, llamada también índice, que va recorriendo un conjunto prefijado de valores en un orden determinado. Para cada valor del índice en dicho conjunto, se ejecuta una vez el mismo conjunto de instrucciones.

En las siguientes figuras se han representado la forma habitual de escribir esta estructura (organigrama) y el diagrama de flujo correspondiente: el bloque de instrucciones se ejecuta una vez para cada valor del índice, que va tomando sucesivamente el valor de cada componente del vector V, de longitud N.

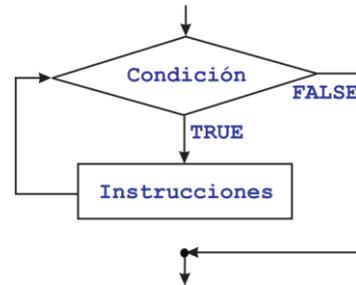


4. WHILE

Permite implementar la repetición de un mismo conjunto de instrucciones mientras que se verifique una determinada condición: el número de veces que se repetirá el ciclo no está definido a priori.

La sintaxis habitual y el diagrama de flujo descriptivo de esta estructura se muestran en las siguientes figuras:

```
...  
WHILE expresión lógica  
    instrucciones  
END  
...
```



5. DO - WHILE

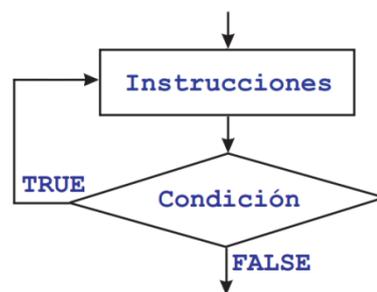
La estructura DO - WHILE es otra estructura repetitiva, la cual ejecuta al menos una vez su bloque repetitivo, a diferencia de WHILE o de FOR que pueden no ejecutar el bloque.

Esta estructura repetitiva se utiliza cuando conocemos de antemano que el bloque de instrucciones debe ejecutarse por lo menos una vez.

La condición de la estructura se encuentra después del bloque a repetir, a diferencia de lo que sucede con WHILE o FOR, en cuyos casos se encuentra en la parte superior.

El organigrama y el correspondiente diagrama de flujo descriptivo de esta estructura se muestran en las figuras a continuación:

```
...  
DO  
    instrucciones  
WHILE expresión lógica  
END  
...
```



6. RUPTURA DE CICLOS DE REPETICIÓN: BREAK y CONTINUE

En ocasiones es necesario interrumpir la ejecución de un ciclo de repetición en algún punto interno del bloque de instrucciones que se repiten. Lógicamente, ello dependerá de que se verifique o no alguna condición.

La interrupción puede hacerse de dos formas:

1. Abandonando el ciclo de repetición definitivamente.
2. Abandonando la iteración en curso, pero comenzando la siguiente.

Las instrucciones para poner esto en práctica tienen nombres diversos en los distintos lenguajes de programación. Habitualmente, la primera opción se implementa con la instrucción BREAK y la segunda con la instrucción CONTINUE. Ambas pueden utilizarse tanto para romper un ciclo FOR como un ciclo WHILE. Cuando se utiliza la orden BREAK dentro de un ciclo FOR, el índice del bucle conserva, fuera del mismo, el último valor que tomó.

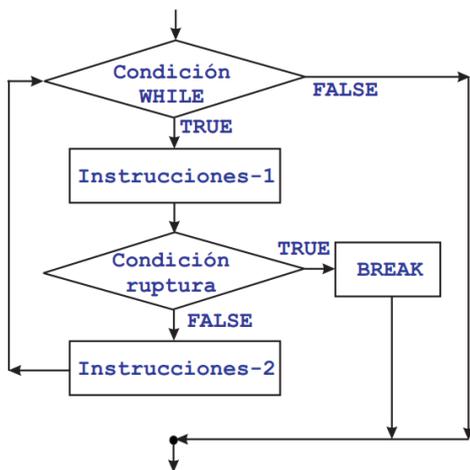


Diagrama de flujo de la ruptura de ciclo BREAK.

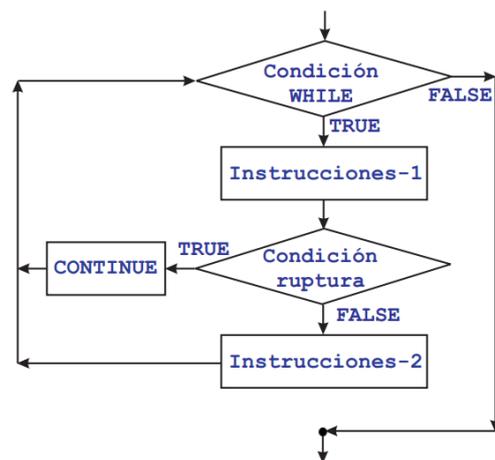


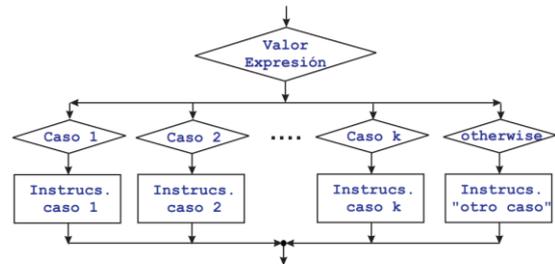
Diagrama de flujo de la ruptura de ciclo CONTINUE.

7. SWITCH

Este tipo de estructura permite decidir entre varios caminos posibles, en función del valor que tome una determinada instrucción.

La sintaxis habitual, así como el diagrama de flujo correspondiente a una de estas estructuras (con cuatro casos), se presentan a continuación:

```
switch expresión
  case valor 1
    instrucciones caso 1
  case valor 2
    instrucciones caso 2
  ...
  case {valores ...}
    instrucciones caso k
  otherwise
    instrucciones
end
```



En cada uno de los casos, el valor correspondiente puede ser o bien un sólo valor, o bien un conjunto de valores, en cuyo caso se indican entre llaves.

La cláusula OTHERWISE y su correspondiente conjunto de instrucciones pueden no estar presentes.

El funcionamiento es el siguiente:

1. Al comienzo se evalúa la expresión.
2. Si la expresión toma el valor (ó valores) especificados junto a la primera cláusula CASE, se ejecuta el conjunto de instrucciones de este caso y después se abandona la estructura SWITCH, continuando por la instrucción siguiente al END.
3. Se repite el procedimiento anterior, de forma ordenada, para cada una de las cláusulas CASE que siguen.
4. Si la cláusula OTHERWISE está presente y la expresión no ha tomado ninguno de los valores anteriormente especificados, se ejecuta el conjunto de instrucciones correspondiente.

Obsérvese que se ejecuta, como máximo el conjunto de instrucciones de uno de los casos, es decir, una vez que se ha verificado un caso y se ha ejecutado su conjunto de instrucciones, no se testea el resto de casos, ya que se abandona la estructura.

Obviamente, si la cláusula OTHERWISE no está presente, puede ocurrir que no se dé ninguno de los casos.