Estructuras de
Control

Cuando se diseña completamente el algoritmo y se requiere escribir el esquema de representación en pseudocódigo o diagrama de flujo, es necesario incorporar estructuras de control que permitan el programa realizar las acciones necesarias dependiente de las condiciones que se encuentren en cada momento de la ejecución, debido a que permiten modificar el flujo de ejecución de las instrucciones de un programa.

Con las estructuras de control se puede:

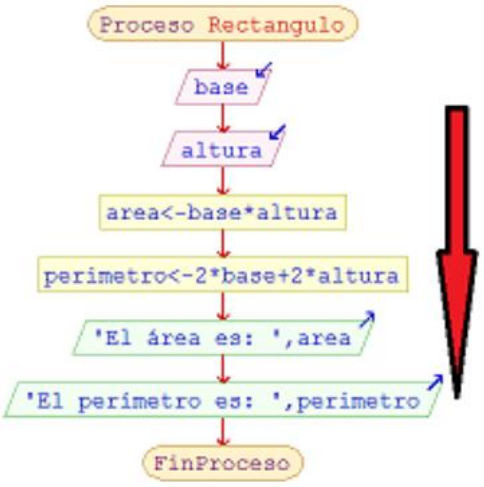
- De acuerdo a una condición, ejecutar un grupo u otro de sentencias (If-Then-Else)
- De acuerdo al valor de una variable, ejecutar un grupo u otro de sentencias (Select-Case)
- Ejecutar un grupo de sentencias mientras se cumpla una condición (Do-While)
- Ejecutar un grupo de sentencias hasta que se cumpla una condición (Do-Until)
- Ejecutar un grupo de sentencias un número determinado de veces (For-Next)

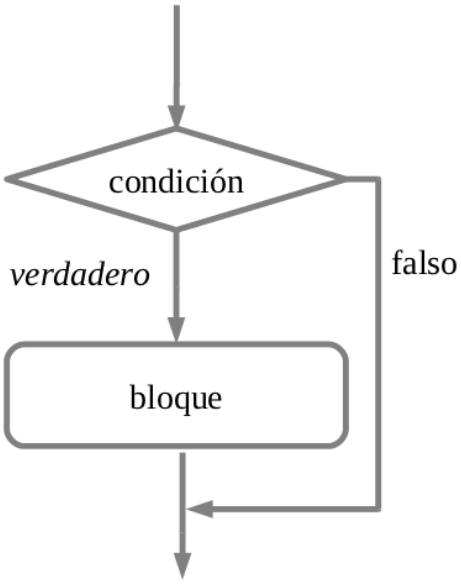
Las estructuras de control permiten que la programación se rija por los principios de la programación estructurada, los lenguajes de programación modernos tienen estructuras de control similares. Básicamente lo que varía entre las estructuras de control de los diferentes lenguajes es su sintaxis, cada lenguaje tiene una sintaxis propia para expresar la estructura.

Las estructuras algorítmicas de control están conformadas en tres grupos:

a) Secuenciales

Pero por lo general las instrucciones se ejecutan una después de la otra, en el orden en que están escritas, es decir, en secuencia.



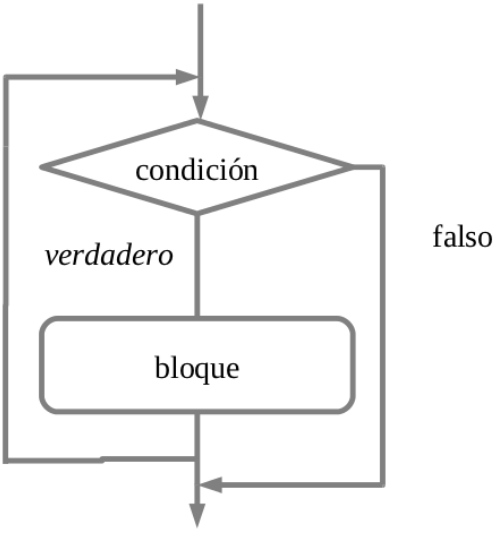


b) Selección

Ejecutan un bloque de instrucciones u otro, o saltan a un subprograma o subrutina según se cumpla o no una condición.

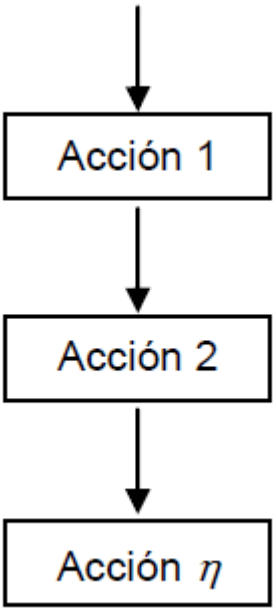
c) Repetitivas o iterativas

Inician o repiten un bloque de instrucciones si se cumple una condición o mientras se cumple una condición.



Estructura Secuencia

Se utiliza cuando queremos mostrar que de una acción le sigue otra y así sucesivamente, hasta el final del proceso. Para indicar el proceso utilizamos el rectángulo y en la parte interna la descripción de la acción.



Todas las instrucciones que se realizan en forma secuencial se ejecutarán una detrás de la otra sin posibilidad de hacer cambios en el orden de solicitud de entradas, procesos o salidas.

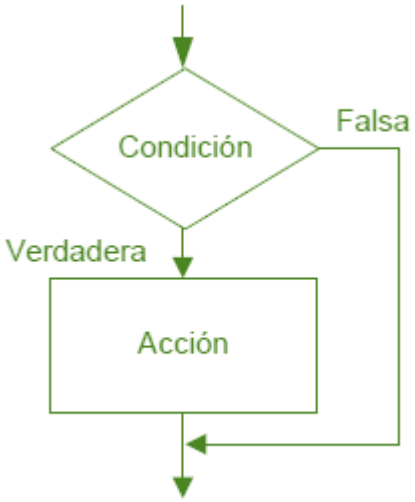
Por lo general lo programas utilizan estructuras de control secuencial para el inicio y fin, teniendo también la posibilidad de ser agregadas dentro de alguna estructura de control de “selección” o “iteración”.

Estructura Selectiva

Las estructuras de selección o condicionales controlan si una sentencia o secuencia de sentencias se ejecutan, en función del cumplimiento o no de una condición o expresión lógica. A continuación se encuentran las principales estructuras selectivas.

a.) SI entonces o alternativa simple.

Se ejecuta una acción cuando se cumple una condición. Esta estructura en inglés se conoce como (If then), su representación es:

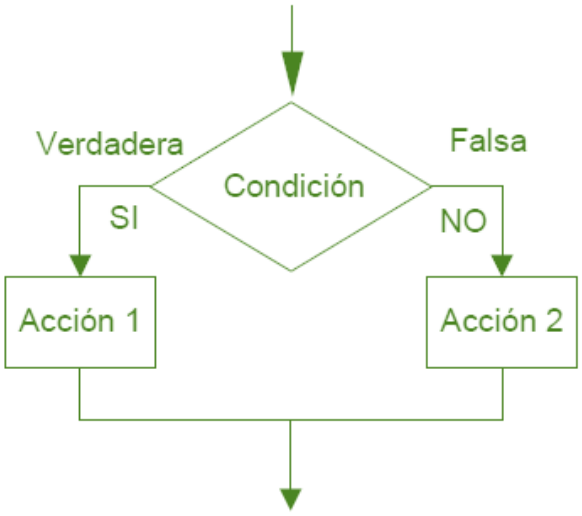


Se trata de una estructura de control que permite desviar un curso de acción según la condición simple, sea falsa o verdadera. Si la condición es verdadera, se ejecuta la condición 1, de lo contrario, se ejecuta la condición 2.

```
IF (Condición) THEN
   (Bloque de sentencias 1)
ELSE
   (Bloque de sentencias 2)
END IF
```

b.) SI entonces o alternativa doble

Se permite elegir entre dos opciones o alternativas posibles, de tal forma que si el resultado es verdadero se sigue por el camino correspondiente y se procede a ejecutar la(s) acción(es); pero si el resultado de la condición es falso entonces se sigue el otro camino y se procede a ejecutar la(s) otra(s) acción(s). Esta estructura en ingles se conoce como (If then else).



Independientemente de cuál camino se ejecute y qué condición siga, al término de las acciones, se continúa con la secuencia normal. Se le llama estructura algorítmica selectiva en cascada o anidada cuando al tomar una decisión y seleccionar un camino, es necesario tomar otra decisión un número finito de veces.

```
IF (Condición 1) THEN
   (Bloque de sentencias 1)
 ELSEIF (Condición 2) THEN
   (Bloque de sentencias 2)
 .....
 ELSEIF (Condición n) THEN
   (Bloque de sentencias n)
 ELSE
   (Bloque de sentencias ELSE)
END IF
```

c.) Según sea, caso de o alternativa múltiple

Permite que el flujo del diagrama se divida por varios caminos en el punto de la toma de decisión, dependiendo el valor que tome el resultado es la acción que se ejecutará, se conoce como “Case”.

Si el resultado toma el valor de 1, se ejecutará la acción 1, si el resultado toma el valor n, se ejecutara la acción n. En el caso de que se tome un valor que no esté entre 1 y n, se continuará con el flujo normal pero se realizará la acción correspondiente al caso, donde la respuesta no se encuentre en los valores anteriores.

- Se evalúa la expresión, dando como resultado un número.
- Luego, se recorren los "Case" dentro de la estructura buscando que el número coincida con uno de los valores.
- Es necesario que coincidan todos sus valores.
- Cuando se encuentra la primera coincidencia, se ejecuta el bloque de sentencias correspondiente y se sale de la estructura Select-Case.
- Si no se encuentra ninguna coincidencia con ningún valor, se ejecuta el bloque de sentencias de la sección "Case Else".

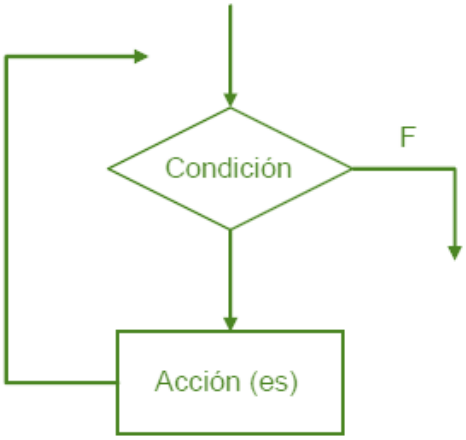
```
SELECT (Expresión)
  CASE Valor1
    (Bloque de sentencias 1)
  CASE Valor2
    (Bloque de sentencias 2)
  CASE Valor n
    (Bloque de sentencias n)
  CASE ELSE
    (Bloque de sentencias "Else")
END SELECT
```

Estructura iterativa

Llamada así porque permite repetir una o varias acciones un número determinado de veces. A las estructuras que repiten una secuencia de instrucciones, un número determinado de veces, se denominan bucles, y se le llama iteración al hecho de repetir la ejecución de una secuencia de acciones.

a. Estructura mientras (en inglés While)

Es aquella que se ejecuta mientras la condición evaluada resulte verdadera, esta estructura es adecuada para utilizar un ciclo que se aplica un número determinado de veces, repitiendo una operación, acción, o tarea. Si la condición es falsa de entrada, no se realizará ninguna de las acciones asignadas




```
DO WHILE (Condición)
    (Bloque de sentencias)
LOOP
```

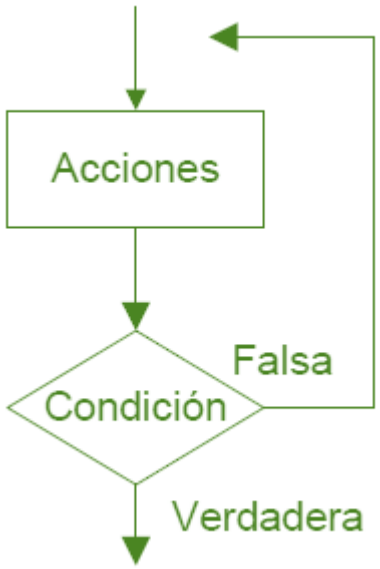
También puede expresarse:

```
WHILE (Condición)
    (Bloque de sentencias)
WEND
```

b. Estructura repetir (en inglés Repeat)

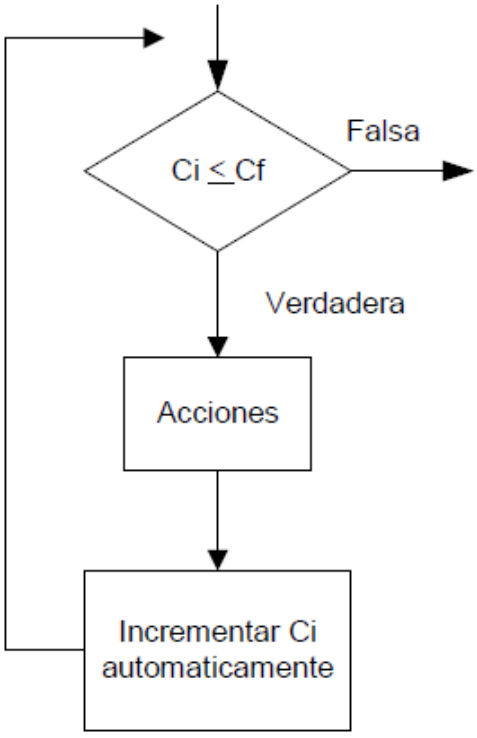
Se ejecuta un número definido de veces (se conoce cuántas veces tenemos que repetir una determinada acción o acciones) y se ejecuta hasta que se cumpla una condición determinada, que se comprueba al final del bucle. En esta estructura, el cuerpo del bucle se ejecuta al menos una vez.

Cuando la instrucción repetir se ejecuta, se hace la ejecución del bucle y se evalúa la expresión booleana resultante de la condición, si es falsa el cuerpo del bucle se repite y la expresión se vuelve a evaluar, después de cada iteración del cuerpo del bucle, se evalúa, si resulta verdadera el bucle termina y pasa a la siguiente instrucción.



c. Estructura desde / para (en inglés FOR)

Se utiliza cuando se conoce el número de veces que se desea ejecutar las acciones de un bucle, en el que el número de iteraciones es fijo. Comienza con un valor inicial índice (contador inicial) y las acciones especificadas se ejecutan si el valor inicial es menor que el valor final (contador final). La variable índice se incrementa en uno y si el valor no excede al final, se ejecuta de nuevo las acciones.



La forma en que se escribe formalmente es:

FOR (Variable) = (Expresión1) TO (Expresión2) STEP (Salto)

(Bloque de sentencias)

NEXT

EDUFUTURO

1204 palabras

Referencias

http://di002.edv.uniovi.es/~dani/asignaturas/transparencias-leccion4.PDF

https://sites.google.com/site/emarinprogramacion/home/6---estructuras-de-control

https://www.fing.edu.uy/inco/cursos/fpr/wiki/index.php/Estructuras_de_Control

http://www.utn.edu.ec/reduca/programacion/estructuras/introduccin.html