

## Estructuras de control

### Introducción

Los algoritmos vistos hasta el momento han consistido en simples secuencias de instrucciones; sin embargo, existen tareas más complejas que no pueden ser resueltas empleando un esquema tan sencillo, en ocasiones es necesario repetir una misma acción un número determinado de veces o evaluar una expresión y realizar acciones diferentes en base al resultado de dicha evaluación.

Para resolver estas situaciones existen las denominadas estructuras de control que poseen las siguientes características:

- Una estructura de control tiene un único punto de entrada y un único punto de salida.
- Una estructura de control se compone de sentencias o de otras estructuras de control.

Tales características permiten desarrollar de forma muy flexible todo tipo de algoritmos aún cuando sólo existen tres tipos fundamentales de estructuras de control:

- Secuencial.
- Alternativa.
- Repetitiva.

A lo largo de esta lección se presentarán las distintas estructuras de control, la forma de representarlas en la notación algorítmica y las correspondientes sentencias FORTRAN para poder utilizarlas en nuestros programas.

### Estructura secuencial

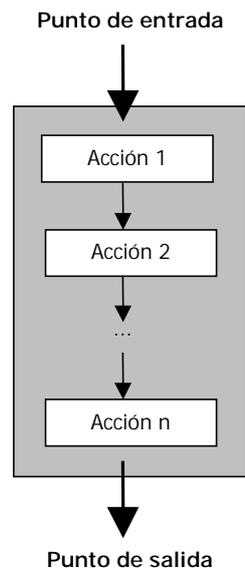
La estructura secuencial es la más sencilla de todas, simplemente indica al procesador que debe ejecutar de forma consecutiva una lista de acciones (que pueden ser, a su vez, otras estructuras de control); para construir una secuencia de acciones basta con escribir cada acción en una línea diferente. A continuación se muestra una composición secuencial de acciones en notación algorítmica y su equivalente FORTRAN.

leer a	read *, a
leer b	read *, b
$c \leftarrow a + b$	$c = a + b$
escribir c	print *, c

Existe una forma alternativa de expresar una estructura secuencial escribiendo varias acciones en la misma línea pero utilizando el punto y coma, `;`, como separador. Sin embargo, esta última notación es desaconsejable puesto que puede llegar a hacer el código bastante difícil de leer.

leer a; leer b	read *, a; read *, b
$c \leftarrow a + b$ ; escribir c	$c = a + b$ ; print *, c

Por último, es necesario señalar un aspecto importante de la composición secuencial y es que no es conmutativa.



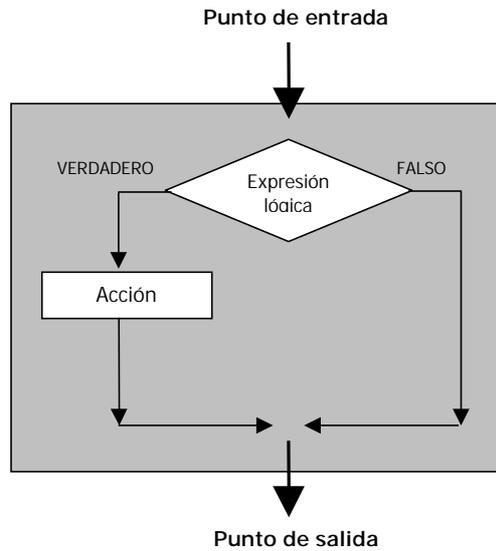
### Estructura alternativa

La estructura alternativa permite bifurcar el “flujo” del programa en función de una expresión lógica; disponemos de tres estructuras alternativas diferentes: alternativa simple, alternativa doble y alternativa múltiple.

#### Estructura alternativa simple

Esta estructura permite evaluar una expresión lógica y en función de dicha evaluación ejecutar una acción (o composición de acciones) o no ejecutarla; también se la suele denominar SI-ENTONCES. A continuación se muestra la notación algorítmica y FORTRAN para la estructura alternativa simple.

si expresión lógica entonces	if (expresión lógica) then
acciones	acciones
fin_si	end if



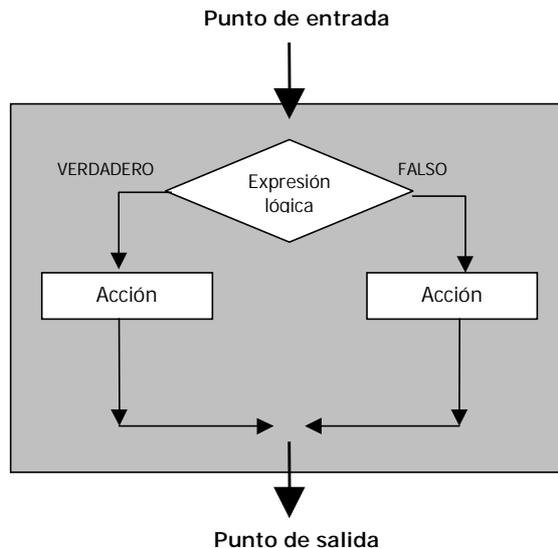
En el siguiente ejemplo se calcula la longitud de una circunferencia si el radio es positivo no nulo:

<code>si radio&gt;0 entonces</code>	<code>if (radio&gt;0) then</code>
<code>  longitud ← 2·pi·radio</code>	<code>  longitud = 2*pi*radio</code>
<code>fin_si</code>	<code>end if</code>

### Estructura alternativa doble

La estructura alternativa doble es similar a la anterior con la salvedad de que en este tipo de estructura se indican acciones no sólo para la rama "verdadera" sino también para la "falsa"; es decir, en caso de la expresión lógica evaluada sea cierta se ejecutan una acción o grupo de acciones y en caso de que sea falsa se ejecuta un grupo diferente. La sintáxis en la notación algorítmica y en FORTRAN son las que se muestran a continuación:

<code>si expresión lógica entonces</code>	<code>if (expresión lógica) then</code>
<code>  acciones</code>	<code>  acciones</code>
<code>si no</code>	<code>else</code>
<code>  acciones</code>	<code>  acciones</code>
<code>fin_si</code>	<code>end if</code>



En el siguiente ejemplo se calcula la longitud de una circunferencia si el radio es positivo no nulo y, en caso contrario, se proporciona un mensaje de error:

```

si radio>0 entonces           if (radio>0) then
  longitud ← 2·pi·radio       longitud = 2*pi*radio
si no                          else
  escribir 'Radio incorrecto' print *, 'Radio incorrecto'
fin_si                          end if

```

### Estructura multialternativa

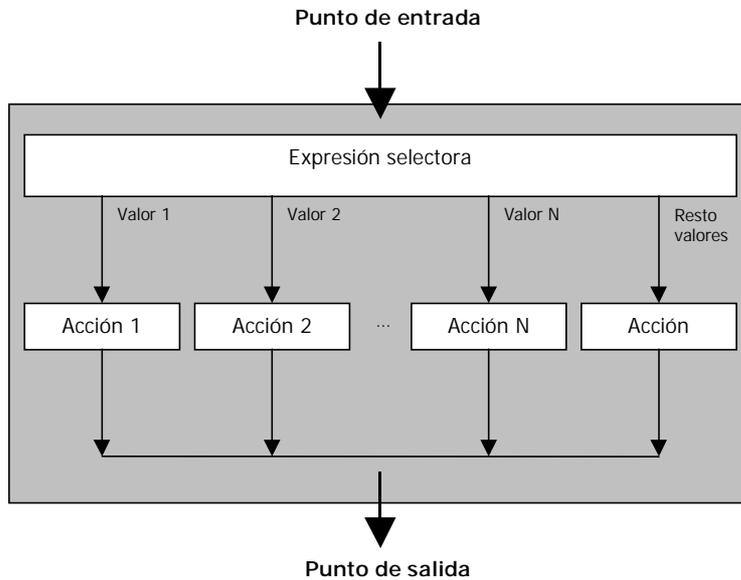
Esta estructura evalúa una expresión que pueda tomar n valores (enteros, caracteres y lógicos pero **nunca** reales) y ejecuta una acción o grupo de acciones diferente en función del valor tomado por la expresión selectora.

La sintaxis de esta estructura es la siguiente:

```

segun expresión              select case (expresión)
  caso valor1:               case (valor1)
    acción 1                  acción 1
  caso valor2:               case (valor2)
    acción 2                  acción 2
  ...                        ...
  caso valorN:               case (valorn)
    acción n                  acción n
  otro caso:                 case default
    acción                    acción
fin según                    end select

```



En el siguiente ejemplo se proporciona como salida el número de días de un mes dado:

```

segun mes                    select case (mes)
  caso 1,3,5,7,8,10,12:      case (1,3,5,7,8,10,12)
    escribir '31'             print *, '31'
  caso 4,6,9,11:            case (4,6,9,11)
    escribir '30'             print *, '30'
  caso 2:                    case (2)
    escribir '28'             print *, '28'
  otro caso:                  case default
    escribir 'Mes incorrecto' print *, 'Mes incorrecto'
fin según                    end select

```

Obsérvese que es posible que un caso conste de múltiples valores.

## Estructura repetitiva

La estructura repetitiva o iterativa permite, como su propio nombre indica, repetir una acción (o grupo de acciones); dicha repetición puede llevarse a cabo un número prefijado de veces o depender de la evaluación de una expresión lógica. Existen tres tipos de estructuras repetitivas: desde-hasta, mientras y repetir-hasta.

### Estructura desde-hasta

Esta estructura permite repetir la ejecución de una acción o de un grupo de acciones un número determinado de veces; la sintaxis es la siguiente:

```
desde indice←inicio hasta fin [con paso valor] hacer      do indice=inicio, fin, paso
  acción                                                    acción
fin desde                                                end do
```

El funcionamiento de la estructura es el siguiente:

- En primer lugar, se asigna a la variable `indice` el valor de `inicio`.
- El bucle se ejecuta mientras `indice` no alcance el valor de `fin`.
- En cada iteración el valor de `indice` es incrementado según el paso indicado y se ejecuta la acción o grupo de acciones encerrados en el bucle.
- En caso de que no se indique ningún paso el que se empleará será `+1`.

A continuación se muestran algunos ejemplos:

```
desde n←1 hasta 10 hacer      do n=1, 10
  escribir n                  print *, n
fin desde                    end do
```

El bucle anterior imprime por pantalla los números del 1 al 10.

```
desde n←10 hasta 1 hacer     do n=10, 1
  escribir n                  print *, n
fin desde                    end do
```

El bucle anterior **no se ejecuta nunca** puesto que no se puede alcanzar 1 comenzando en 10 y avanzando con paso `+1`; ¡**atención!** Un error frecuente es pensar que el bucle se ejecuta de forma infinita.

```
desde n←10 hasta 1 con paso -2 hacer  do n=10, 1, -2
  escribir n                      print *, n
fin desde                          end do
```

Este bucle escribe los números pares de 10 a 2 en orden inverso.

### Estructura mientras

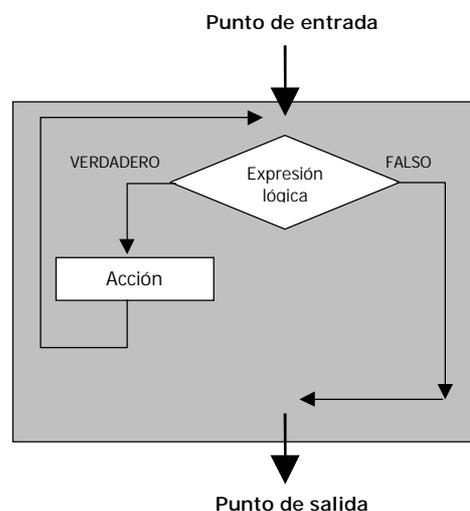
Esta estructura repite una acción o grupo de acciones mientras una expresión lógica sea cierta; la sintaxis en la notación algorítmica y en FORTRAN es la siguiente:

```
mientras expresión lógica hacer      do while (expresión lógica)
  acción                               acción
fin desde                             end do
```

Un aspecto muy importante de la presente estructura de control es que si la expresión lógica es inicialmente falsa el bucle no se ejecuta ni una sola vez; es decir, la estructura **mientras** supone que el bucle iterará 0 ó más veces.

A continuación se muestra un ejemplo que solicita al usuario el radio de una circunferencia mientras el radio introducido sea incorrecto:

```
mientras radio<0 hacer          do while (radio<0)
  escribir 'Radio?'             print *, 'Radio?'
  leer radio                    read *, radio
fin desde                      end do
```

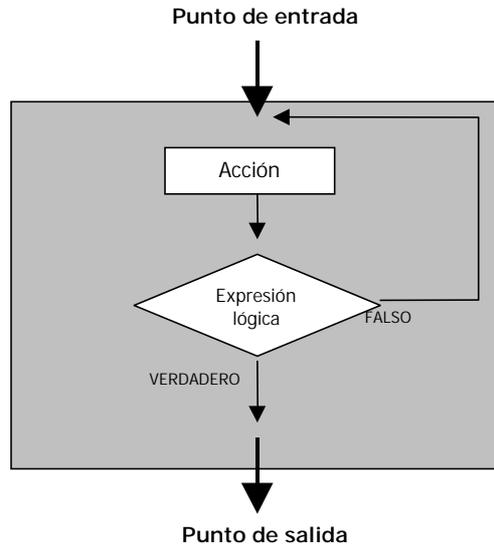


## Estructura **repetir-hasta**

Esta estructura repite una acción o grupo de acciones hasta que una expresión lógica sea cierta; la sintaxis en la notación algorítmica es la siguiente (no existe equivalente en FORTRAN pero sí en otros lenguajes de programación):

```
repetir
  acción
hasta expresión lógica
```

Un aspecto muy importante de la presente estructura de control es que la expresión lógica no se evalúa hasta el final con lo cual el bucle se ejecuta al menos una vez por contraposición a la estructura anterior que podía no ejecutarse ninguna.



## Resumen

1. Las estructuras de control permiten alterar el flujo secuencial de los algoritmos que hemos visto hasta el momento.
2. Una estructura de control:
  - Tiene un único punto de entrada y un único punto de salida.
  - Se compone de sentencias o de otras estructuras de control.
3. Existen tres tipos fundamentales de estructuras de control:
  - Secuencial.
  - Alternativa.
  - Repetitiva.
4. La estructura secuencial es una lista consecutiva de acciones donde el orden de aparición indica el orden de ejecución.
5. La estructura alternativa se presenta en tres formas:
  - Simple.
  - Doble.
  - Multialternativa.
6. La estructura alternativa simple permite evaluar una expresión lógica y ejecutar una acción si dicha expresión es cierta.
7. La estructura alternativa doble permite evaluar una expresión lógica y ejecutar una acción si dicha expresión es cierta y otra acción si dicha expresión es falsa.
8. La estructura multialternativa consiste en una expresión selectora que puede tomar n valores diferentes y ejecuta una acción diferente por cada posible valor de la expresión selectora.

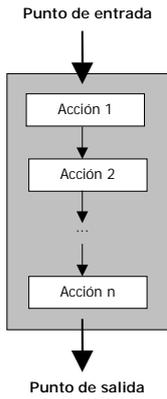
9. La estructura repetitiva se presenta en tres formas:

- Estructura **desde-hasta**
- Estructura **mientras**
- Estructura **repetir-hasta**

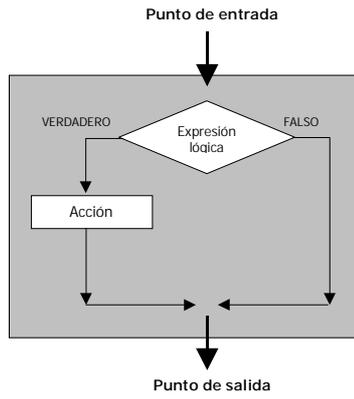
10. La estructura **desde-hasta** permite repetir la ejecución de una acción un número determinado de veces.

11. La estructura **mientras** repite una acción mientras una expresión lógica sea cierta; esta estructura puede ejecutarse **0 ó más veces**.

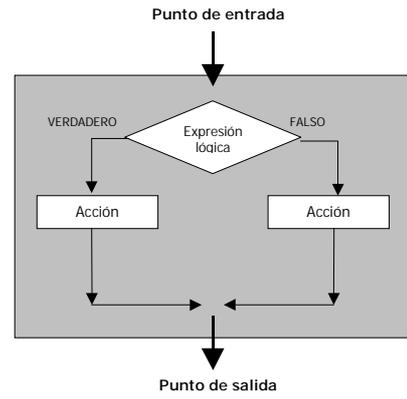
12. La estructura **repetir-hasta** repite una acción hasta que una expresión lógica sea cierta; esta estructura siempre **se ejecuta 1 vez al menos** y no existe en el lenguaje FORTRAN aunque aparece en otros lenguajes de programación.



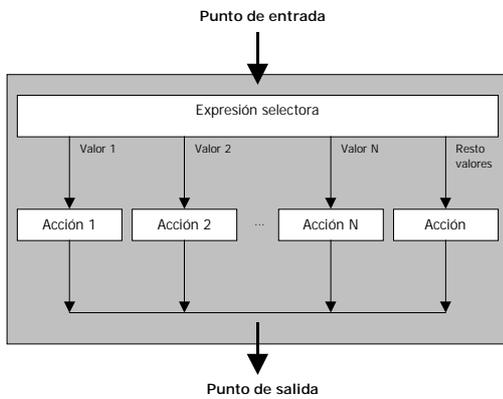
**Estructura secuencial**



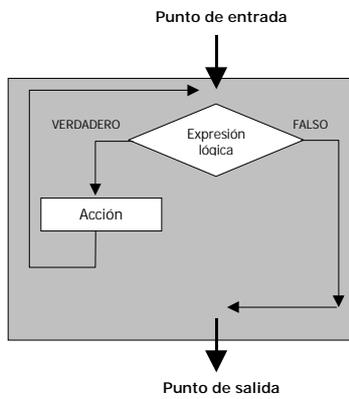
**Estructura alternativa simple**



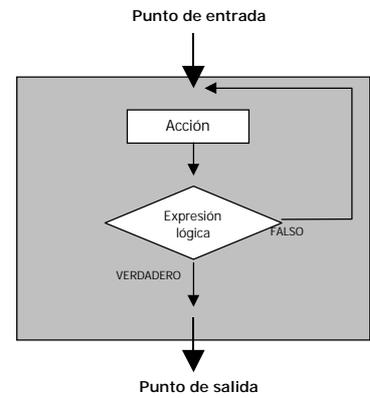
**Estructura alternativa doble**



**Estructura alternativa múltiple**



**Estructura mientras**



**Estructura repetir-hasta**