

Estructuras de Datos

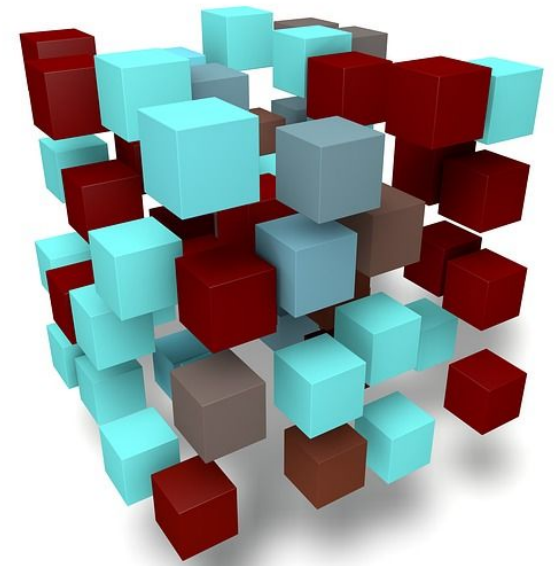
Una “**Estructura de Datos**” es una colección de datos que se caracterizan por su organización y las operaciones que se definen en ella.

Los datos estructurados, llamados también estructuras de datos, son una colección o conjunto de datos simples que tiene el mismo nombre.

Estructuras de Datos

Las Estructuras de Datos se Clasifican en:

- 1- Estructuras de Datos Estáticas**
- 2- Estructuras de Datos Dinámicas**

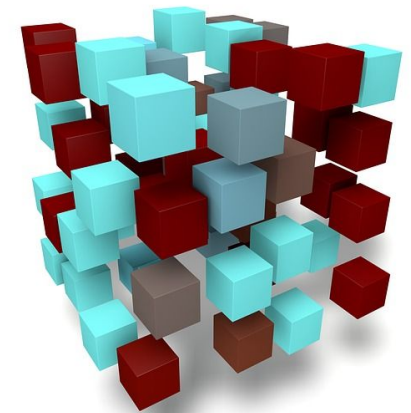


Estructuras de Datos

1- Estructuras de Datos Estáticas

Son aquellas en las que el espacio ocupado en memoria se define en tiempo de compilación y no puede ser modificado durante la ejecución del programa. Entre ellas tenemos:

- Arrays
- Registros
- Archivos



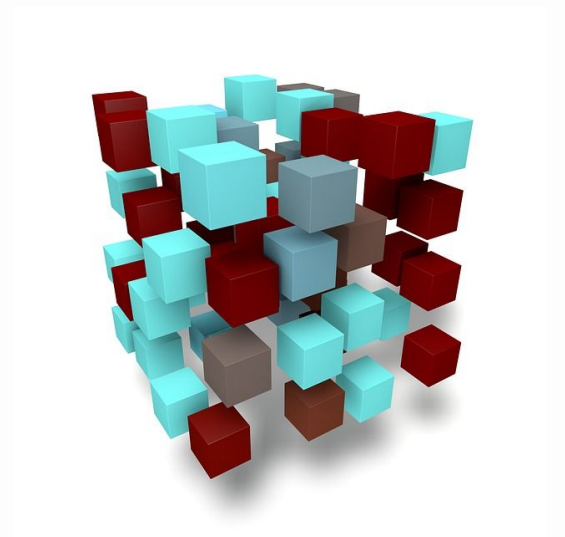
Estructuras de Datos

1- Estructuras de Datos Dinámicas

Son aquellas en las que el espacio ocupado en memoria puede ser modificado en tiempo de ejecución.

Corresponden a este tipo:

- Lineales (Pilas, Colas y listas enlazadas)
- No lineales (árboles y grafos)



Estructuras de Control

Las estructuras de control son aquellas que nos permiten modificar el flujo de ejecución de las instrucciones dentro de un programa.

Estas estructuras de control 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.



Estructuras de Control

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

- Secuencial.
- Alternativa.
- Repetitiva.

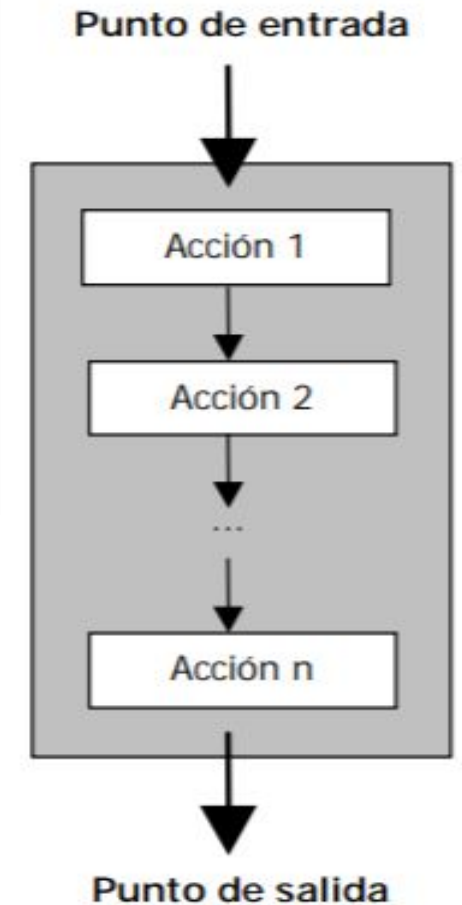


Estructuras de Control

Estructura de Control 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.

```
leer a
leer b
c ← a + b
escribir c
```



Estructuras de Control

Estructura de Control 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
- Alternativa múltiple

Estructuras de Control

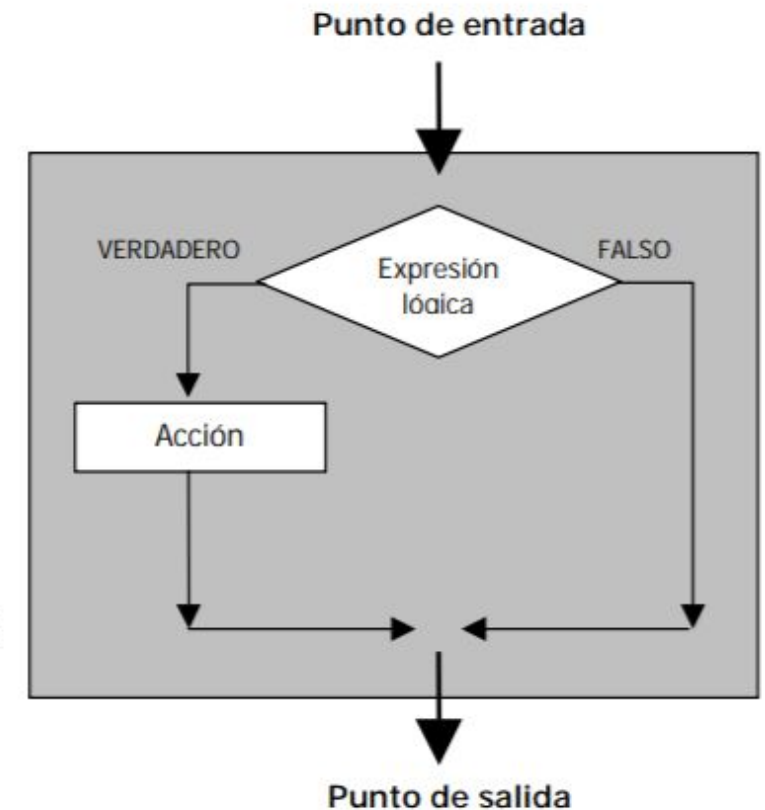
Estructura Alternativa Simple -

Esta estructura permite evaluar una expresión lógica y ejecutar una acción (o composición de acciones) o no ejecutarla, denominar SI-ENTONCES.

<code>si expresión lógica entonces</code>	<code>if (expresión lógica) then</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:

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



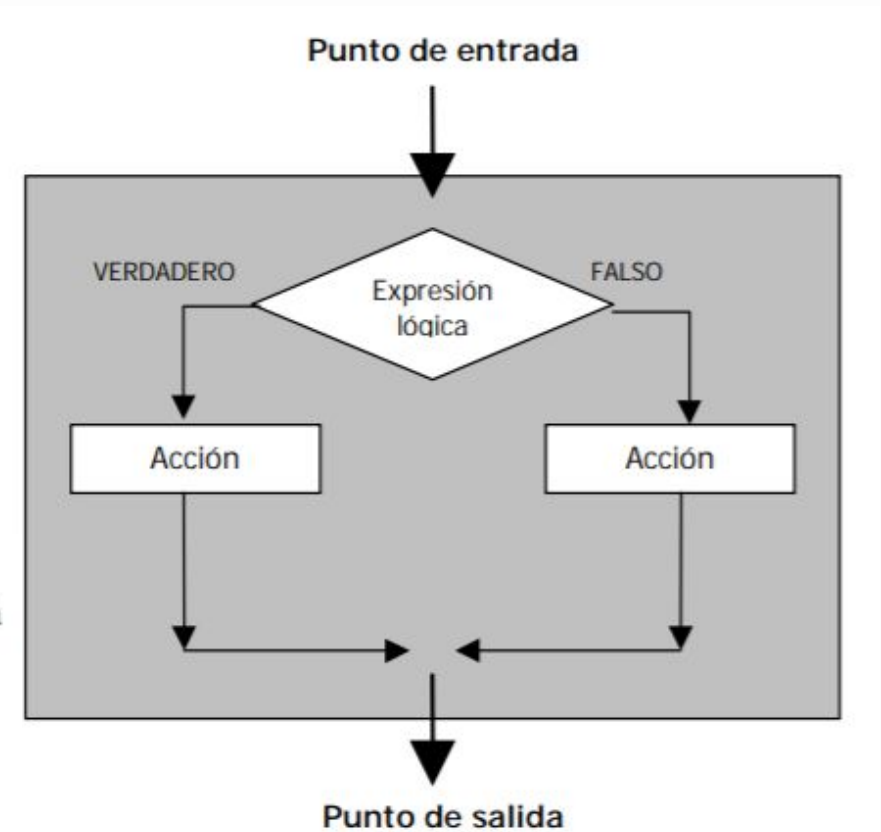
Estructuras de Control

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 ejecuta una acción o grupo de acciones y en caso de que sea falsa se ejecuta un grupo diferente.

```
si expresión lógica entonces  
    acciones  
si no  
    acciones  
fin_si
```

```
if (expresión lógica) then  
    acciones  
else  
    acciones  
end if
```



Estructuras de Control

Estructura Alternativa Doble –

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:

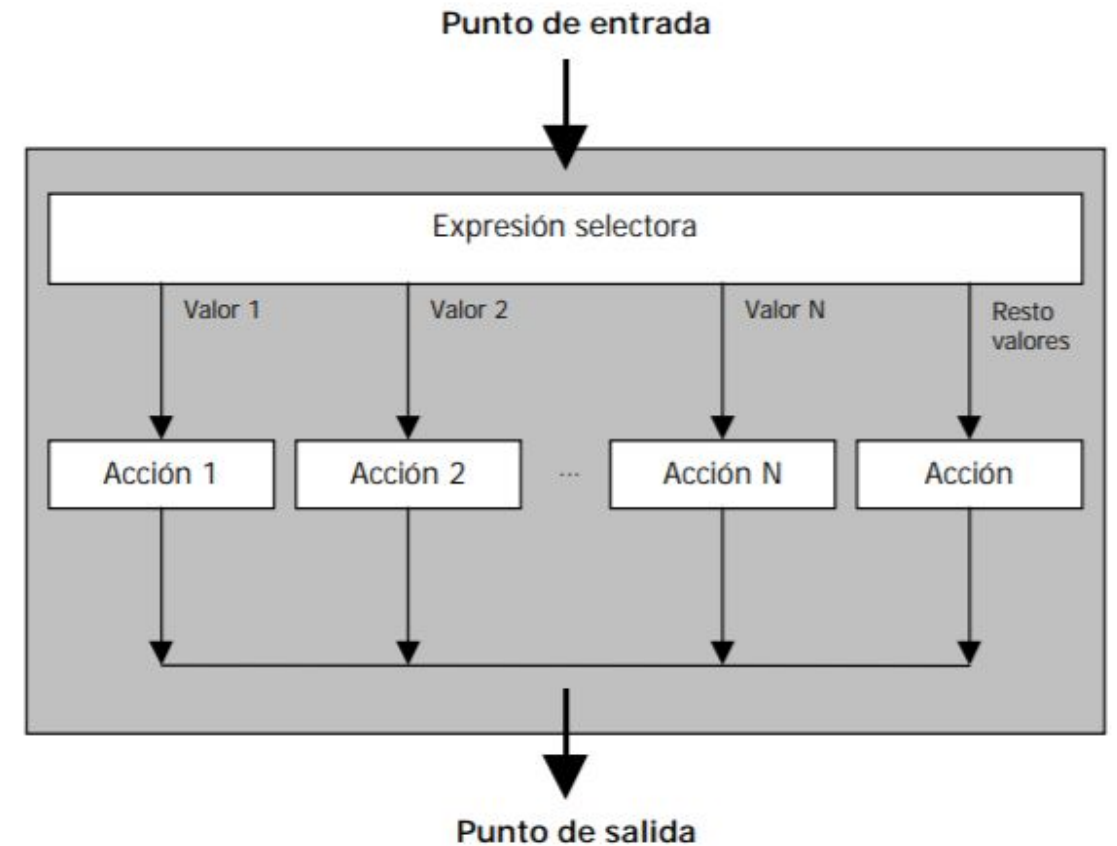
```
si radio>0 entonces  
    longitud ← 2·pi·radio  
si no  
    escribir 'Radio incorrecto'  
fin_si
```

```
if (radio>0) then  
    longitud = 2*pi*radio  
else  
    print *, 'Radio incorrecto'  
end if
```

Estructuras de Control

Estructura Alternativa Multiple –

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:



Estructuras de Control

Estructura Alternativa Multiple –

Pseudocódigo:

```
segun expresión
  caso valor1:
    acción 1
  caso valor2:
    acción 2
  ...
  caso valorN:
    acción n
  otro caso:
    acción
fin según
```

```
select case (expresión)
  case (valor1)
    acción 1
  case (valor2)
    acción 2
  ...
  case (valorn)
    acción n
  case default
    acción
end select
```

Estructuras de Control

Estructura Alternativa Multiple –

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

```
segun mes
  caso 1,3,5,7,8,10,12:
    escribir '31'
  caso 4,6,9,11:
    escribir '30'
  caso 2:
    escribir '28'
  otro caso:
    escribir 'Mes incorrecto'
fin según
```

```
select case (mes)
  case (1,3,5,7,8,10,12)
    print *, '31'
  case (4,6,9,11)
    print *, '30'
  case (2)
    print *, '28'
  case default
    print *, 'Mes incorrecto'
end select
```

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

Estructuras de Control

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
- repetir-hasta

Estructuras de Control

Estructura Repetitiva – “desde-hasta”

El funcionamiento de la estructura es el siguiente:

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

Estructuras de Control

Estructura Repetitiva – “desde-hasta”

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
```

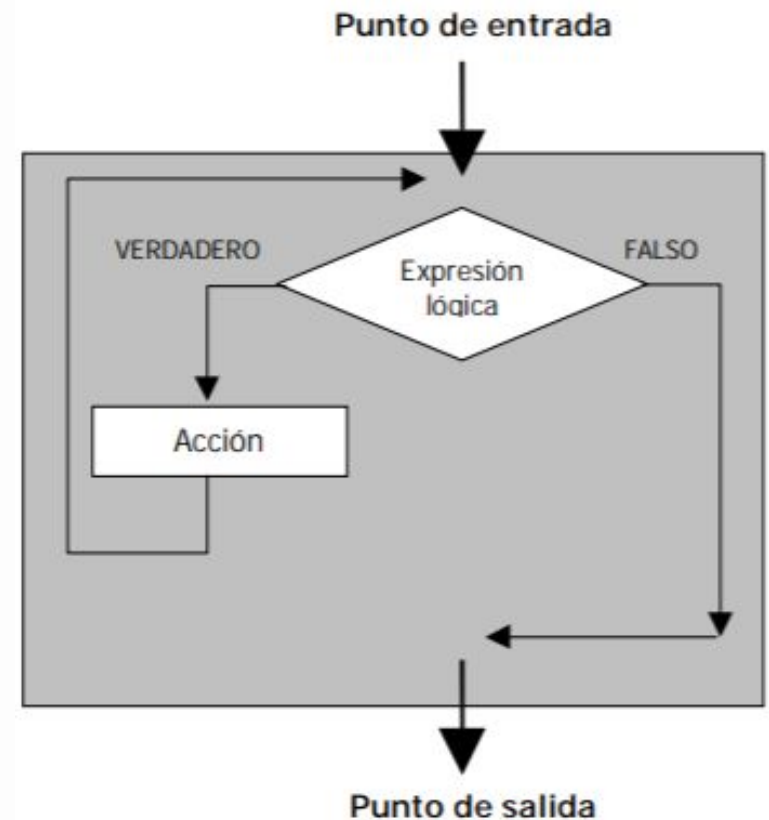
Este bucle escribe los números pares de 10 a 2 en orden inverso.

Estructuras de Control

Estructura Repetitiva – “mientras”

Esta estructura repite una acción o grupo de acciones mientras una expresión lógica sea cierta.

```
mientras expresión lógica hacer  
    acción  
fin desde
```



Estructuras de Control

Estructura Repetitiva – “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:

```
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 podría no ejecutarse ninguna

