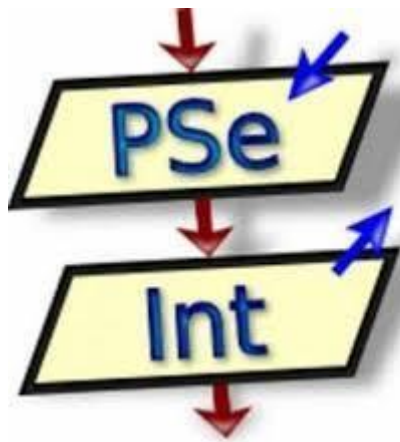


Estructuras de Decisión



2024

Estructuras de Decisión

Estas estructuras de control son de gran utilidad para cuando el algoritmo a desarrollar requiera una descripción más complicada que una lista sencilla de instrucciones. Este es el caso cuando existe un número de posibles alternativas que resultan de la evaluación de una determinada condición. Este tipo de estructuras son utilizadas para tomar decisiones lógicas, es por esto que también se denominan estructuras de decisión, condicionales, o estructuras selectivas.

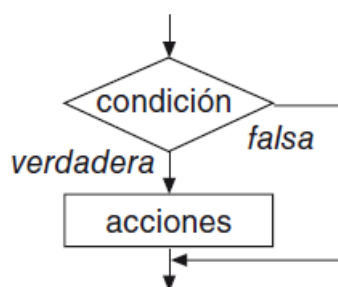
Las estructuras de decisión son sentencias de control que cambian el flujo de ejecución de un programa de acuerdo a si se cumple o no una condición. Cuando el flujo de control del programa llega al condicional, el programa evalúa la condición y determina el camino a seguir. Los condiciones son especificadas utilizando expresiones lógicas.

Las estructuras de decisión pueden ser:

- ✓ Simples
- ✓ Dobles
- ✓ Múltiples

Condición Simple

La estructura selectiva simple “*si-entonces*” lleva a cabo una acción siempre y cuando se cumpla una determinada *condición*, es decir, esta sentencia le indica al programa que ejecute cierta parte del código sólo si la *condición* evaluada es *verdadera*.



La selección si-entonces evalúa la condición y luego:

- ✓ Si la condición es verdadera, ejecuta el bloque de acciones
- ✓ Si la condición es falsa, no ejecuta nada.

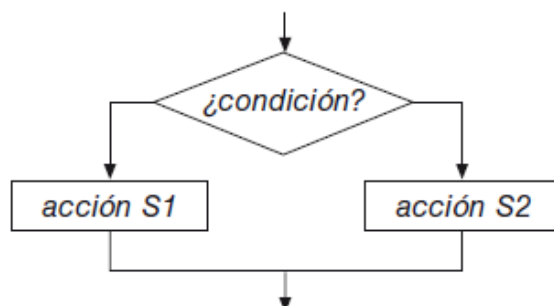
Pseudocódigo en PSeInt:

```
Si condicion Entonces
    acciones
Fin Si
```

En donde, <condición> es una expresión condicional cuyo resultado luego de la evaluación es un dato booleano verdadero o falso. El bloque de instrucciones se ejecuta si, y sólo si, la expresión (que debe ser lógica) se evalúa a verdadero, es decir, se cumple la condición.

Condición Doble

La estructura anterior es muy limitada y normalmente se necesitará una estructura que permita elegir entre dos opciones o alternativas posibles, en función del cumplimiento o no de una determinada condición. Si la *condición* es *verdadera*, se ejecuta la *acción S1* y, si es *falsa*, se ejecuta la *acción S2*.



Pseudocódigo en PSeInt:

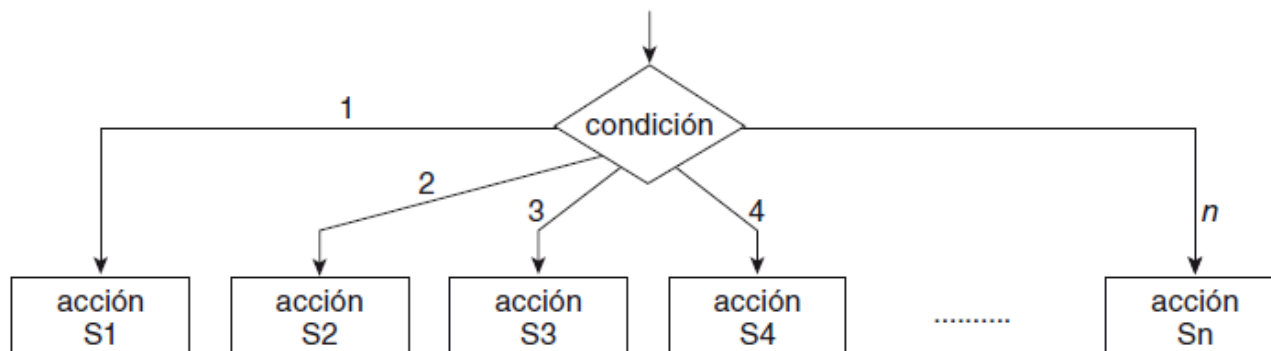
```
Si condicion Entonces
    acciones_por_verdadero
Sino
    acciones_por_falso
Fin Si
```

En donde, <condición> al igual que en el caso anterior es una expresión lógica. El flujo de control del programa funciona de la misma manera, cuando llega a la sentencia *Si*, se evalúa la expresión condicional, si el resultado de la condición es verdadero se ejecutan las sentencias que se encuentran contenidas dentro del bloque de código *Si* (<acciones_por_verdadero>). Contrariamente, en caso de que la condición sea falsa se ejecutan las sentencias contenidas dentro del bloque *Sino* (<acciones_por_falso >).

Condición Múltiple

Se utiliza cuando existen más de dos alternativas para elegir. En esta estructura, se evalúa una *condición* o expresión que puede tomar n valores posibles. Según el valor que la expresión tenga en cada momento se ejecutan las acciones correspondientes al valor.

La estructura de decisión múltiple evaluará una expresión que podrá tomar n valores distintos, 1, 2, 3, 4, ..., n . Según que elija uno de estos valores en la condición, se realizará una de las n acciones, o lo que es igual, el flujo del algoritmo seguirá un determinado camino entre los n posibles.



Pseudocódigo en PSeInt:

```

Segun variable_numerica Hacer
  opcion_1:
    secuencia_de_acciones_1
  opcion_2:
    secuencia_de_acciones_2
  opcion_3:
    secuencia_de_acciones_3
  De Otro Modo:
    secuencia_de_acciones_dom
Fin Segun
  
```

Nota:

Cuando el valor de la variable que se evalúa no coincide con ninguno de los valores que se evalúa, entonces se ejecutan las acciones dentro del bloque “De Otro Modo” (secuencia_de_acciones_dom), el cual equivale a realizar un “Sino” dentro de las estructuras condicionales.

Este problema, se podría resolver por estructuras alternativas simples o dobles, anidadas o en cascada; sin embargo, este método si el número de alternativas es grande puede plantear serios problemas de escritura del algoritmo y naturalmente de legibilidad.

Condicionales Anidados o en Cascada

Es posible también utilizar la instrucción *Si* para diseñar estructuras de selección que contengan más de dos alternativas. Por ejemplo, una estructura *Si-entonces* puede contener otra estructura *Si-entonces*, y esta estructura *Si-entonces* puede contener otra, y así sucesivamente cualquier número de veces; a su vez, dentro de cada estructura pueden existir diferentes acciones.

Pseudocódigo en PSeInt:

```
Si expresion_logica1 Entonces
    acciones_por_verdadero1
Sino
    Si expresion_logica2 Entonces
        acciones_por_verdadero2
    Sino
        Si expresion_logica4 Entonces
            acciones_por_verdadero3
        Sino
            acciones_por_falso
        Fin Si
    Fin Si
Fin Si
```