

Programación

Unidad 4: Estructuras de control

"Controlar la complejidad es la esencia de la programación"

-- Brian Kernigan

Estructuras de Control

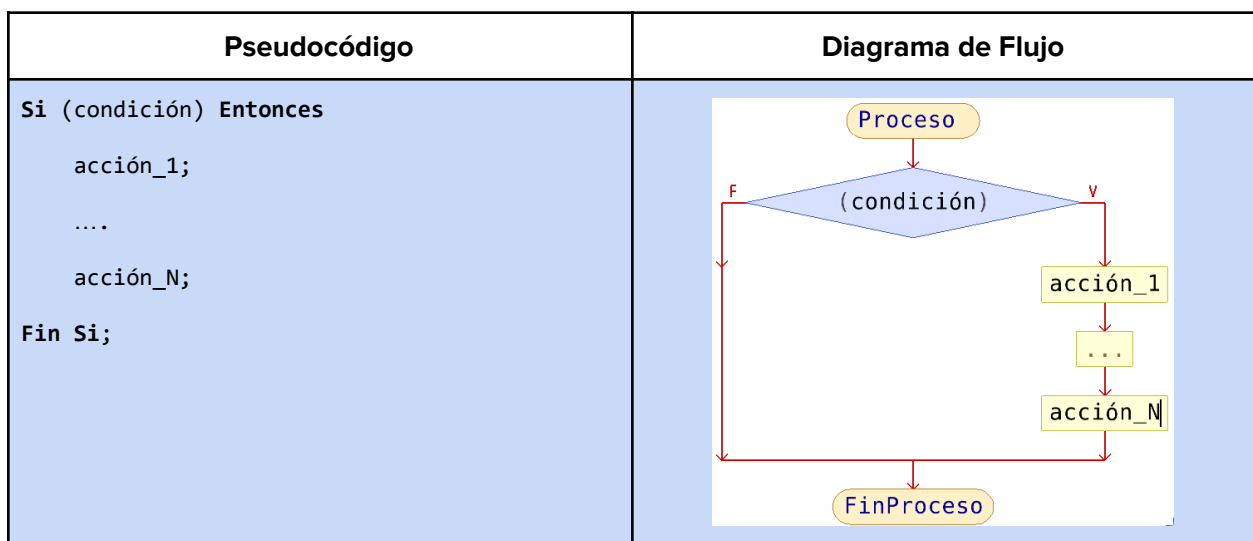
Hasta ahora, en todos los ejemplos que vimos sólo podíamos asignar valores y realizar operaciones con esos valores, pero qué sucede si para mostrar el resultado (la salida) de nuestro programa debo tomar alguna decisión. Por ejemplo, en la unidad temática anterior pudimos calcular la nota promedio de una materia de un alumno, pero... ¿Si tuviésemos que imprimir por pantalla el mensaje “Materia aprobada” o “Materia desaprobada” según el promedio haya sido mayor o menor a 6? Las estructuras de control nos permiten resolver problemáticas como la planteada.

Las estructuras de control, nos permiten controlar el flujo de ejecución del programa, es decir, permiten que el programa tome distintos caminos según se cumplan ciertas condiciones. Estas condiciones deben ser *expresiones lógicas*, expresiones que arrojan un resultado Verdadero o Falso (true o false).

En el caso planteado, nuestra condición a evaluar sería “¿El valor de promedio es mayor a 6?”, en caso Verdadero (true) el programa debería mostrar “Materia aprobada”, en caso contrario debería mostrar “Materia desaprobada”.

Estructura de control “Si...Entonces”

Esta estructura de control posee dos alternativas. La más simple posee la siguiente forma:



¿Cómo funciona esta estructura de control?

Cuando dentro de un programa se alcanza una sentencia “Si...Entonces”, se evalúa la *condición* que se encuentra entre paréntesis. Si el resultado de esta evaluación es Verdadero (true) entonces se ejecutará la o las acciones que se encuentre dentro de este bloque. Si el resultado de evaluar la condición es Falso (false) entonces no se ejecutarán las acciones que están dentro del bloque.

Veamos un ejemplo para clarificar:

Problema: “Ingresar un número entero. Si el número es mayor a 0 (cero) imprimir por pantalla la leyenda ‘Número mayor a 0’. Si es menor a 0 (cero) imprimir por pantalla la leyenda ‘Número menor a 0’. Si es igual a 0 (cero) imprimir por pantalla la leyenda ‘Número igual a 0’.”

```
1  Proceso ValorIngresado
2
3      Definir numero como Entero;
4
5      Mostrar “Ingrese un número: “;
6      Leer numero;
7
8      Si ( numero > 0 ) Entonces
9          Mostrar “Número mayor a 0”;
10     Fin Si
11
12     Si ( numero < 0 ) Entonces
13         Mostrar “Número menor a 0”;
14     Fin Si
15
16     Si ( numero == 0 ) Entonces
17         Mostrar “Número igual a 0”;
18     Fin Si
19 FinProceso
```

En este ejemplo, dependiendo del número ingresado por el usuario se mostrará un mensaje u otro. Es decir, si el número es 4 sólo se mostrará el mensaje “Número mayor a 0”. Los otros mensaje no se mostrarán ya que 4 no es menor a 0 (cero) ni tampoco es igual a 0 (cero).

La segunda alternativa de la estructura “*Si...Entonces*”, nos permite ejecutar algunas acciones en caso de que NO se cumpla la condición evaluada. A esta alternativa la podemos llamar “*Si...Entonces...Sino*”. Esta alternativa tiene la siguiente forma:

Pseudocódigo	Diagrama de Flujo
Si (condición) Entonces acción_1; acción_N; Sino acción_x; acción_z; Fin Si;	<pre> graph TD Proceso([Proceso]) --> Condicion{condición} Condicion -- V --> accion1[acción_1] accion1 --> dots1[...] dots1 --> accionN[acción_N] Condicion -- F --> accionX[acción_x] accionX --> dots2[...] dots2 --> accionZ[acción_z] accionN --> FinProceso([FinProceso]) accionZ --> FinProceso </pre>

En este caso, cuando se alcanza una sentencia “Si...Entonces...Sino”, se evalúa la *condición* que se encuentra entre paréntesis. Si el resultado de esta evaluación es Verdadero (true) entonces se ejecutarán las acciones *acción_1...acción_N*; mientras que si el resultado de evaluar la condición es Falso (false) entonces se ejecutarán las acciones *acción_x...acción_z*.

Veamos un ejemplo para clarificar:

Problema: “Ingresar la nota de un alumno. En caso de que la nota sea mayor o igual a 6, mostrar por pantalla el mensaje ‘Materia aprobada’ . En caso contrario mostrar por pantalla el mensaje ‘Materia desaprobada’ ”

```

1  Proceso Materia
2
3      Definir nota como Real;
4
5      Mostrar “Ingrese la nota del alumno: “;
6      Leer nota;
7
8      Si ( nota >= 6 ) Entonces
9          Mostrar “Materia aprobada “;
10     Sino
11         Mostrar “Materia desaprobada “;
12     Fin Si
13 FinProceso

```

Condiciones múltiples

Independientemente de qué forma del Si estemos utilizando siempre podemos utilizar una condición que evalúe varias cosas y no solamente un valor. Recordemos que las expresiones pueden combinarse por lo cual mientras nuestra expresión genere un valor lógico (verdadero o falso) se puede utilizar como condición en el Si.

Veamos un ejemplo con varias condiciones para clarificar:

Problema: “Ingresar las 2 notas de un alumno e informar por pantalla si aprobó la materia o no. Para aprobar la materia el promedio debe ser mayor o igual a 6, y además debe haber aprobado ambos exámenes”

```
1  Proceso Materia
2
3      Definir nota1 como Real;
4      Definir nota2 como Real;
5      Definir promedio como Real;
6
7      Mostrar “Ingrese la nota del 1er examen del alumno: “;
8      Leer nota1;
9
10     Mostrar “Ingrese la nota del 2do examen del alumno: “;
11     Leer nota2;
12
13     promedio = (nota1 + nota2) / 2;
14
15     Si ( nota1 >= 6 Y nota2 >= 6 Y promedio >= 6 ) Entonces
16         Mostrar “Materia aprobada “;
17     Sino
18         Mostrar “Materia desaprobada “;
19     Fin Si
20 FinProceso
```

Estructura de control “Según”

Esta estructura de control, nos permite seleccionar una entre múltiples alternativas a ejecutar. Esta estructura es especialmente útil cuando la selección de la alternativa a ejecutar se basa en el valor de una variable numérica.

La forma para escribir esta estructura es la siguiente:

Pseudocódigo	Diagrama de Flujo
Segun variable_numérica Hacer opción_1: acción_w; opción_2: acción_k; ... opción_N: acción_m; De Otro Modo: acción_p; Fin Segun;	

¿Cómo funciona esta estructura de control?

Cuando dentro de un programa se alcanza una sentencia “*Segun*”, se evalúa la *variable_numérica* para saber cuál es su valor y luego se compara dicho valor con cada una de las etiquetas (opción_1, opción_2, ..., opción_N). Si alguna de esas comparaciones es Verdadera, entonces se ejecutarán las acciones de dicha etiqueta. Si el valor de la *variable_numérica* no coincide con alguna de las etiquetas, entonces se ejecutarán las acciones agrupadas bajo la etiqueta “De Otro Modo”.

Veamos un ejemplo:

Problema: “Ingresar un número entre 1 y 3. Mostrar por pantalla el número ingresado en letras.”	
1	Proceso Materia
2	
3	Definir numero como Entero;
4	
5	Mostrar “Ingrese un número: “;
6	Leer numero;
7	
8	Segun numero Hacer
9	1:
10	Mostrar “uno”;
11	2:
12	Mostrar “dos”;
13	3:
14	Mostrar “tres”;
15	De Otro Modo:
16	Mostrar “es otro número”;
17	Fin Segun
18	FinProceso

En este ejemplo, si el usuario ingresa el número 1 o el 2 o el 3, se mostrará el mensaje “uno”, “dos” o “tres” respectivamente. Pero si el usuario ingresa el número 5, entonces se mostrará el mensaje “es otro número” ya que 5 no es una opción válida en este caso.

Anidando estructuras

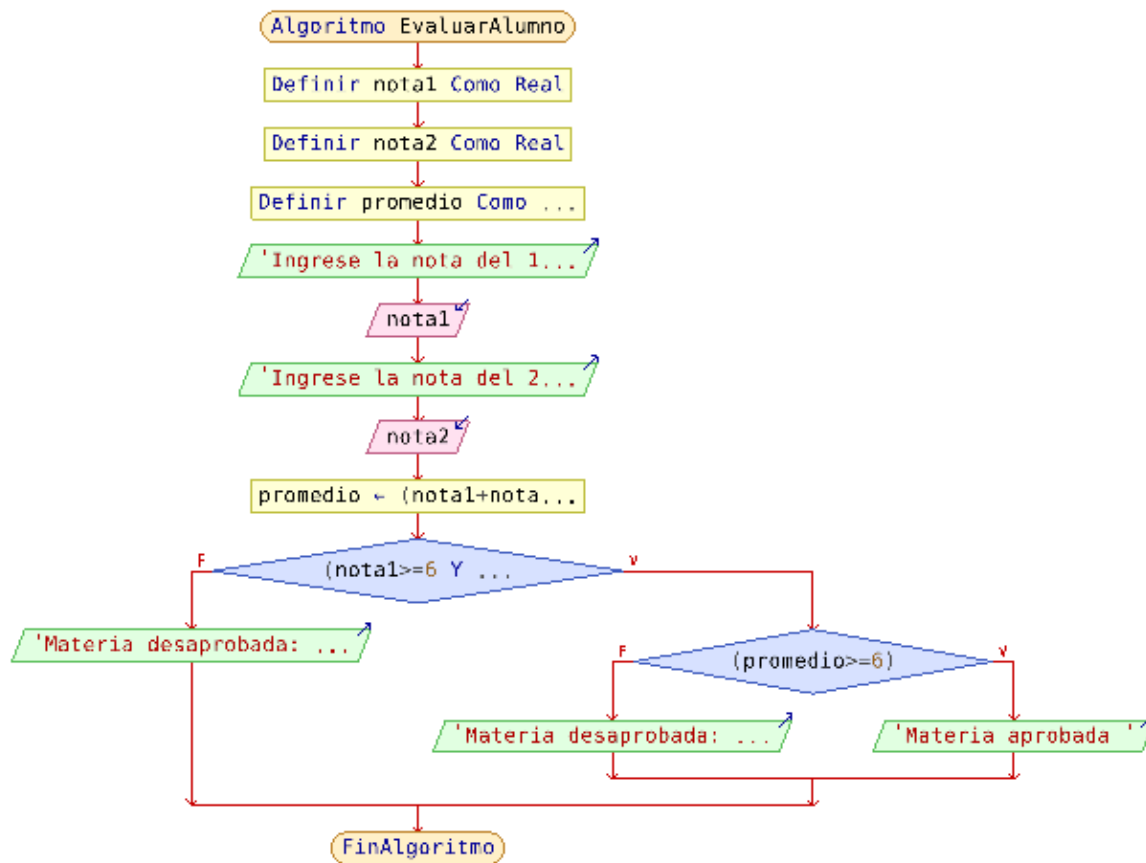
Todas las estructuras pueden combinarse y utilizarse en forma conjunta, es decir se puede realizar un Si dentro de otro Si, o un Si dentro de una de las opciones del Segun, y así infinitamente.

Es decir una estructura puede ser una de las sentencias que se ejecutan dentro de otra estructura.

Veamos algunos ejemplos:

Problema: “Ingresar las 2 notas de un alumno e informar por pantalla si aprobó la materia o no. Para aprobar la materia el promedio debe ser mayor o igual a 6, y además debe haber aprobado ambos exámenes”

```
1  Proceso Materia
2
3      Definir nota1 como Real;
4      Definir nota2 como Real;
5      Definir promedio como Real;
6
7      Mostrar “Ingrese la nota del 1er examen del alumno: “;
8      Leer nota1;
9
10     Mostrar “Ingrese la nota del 2do examen del alumno: “;
11     Leer nota2;
12
13     promedio = (nota1 + nota2) / 2;
14
15     Si ( nota1 >= 6 Y nota2 >= 6) Entonces
16         Si (promedio >= 6) Entonces
17             Mostrar “Materia aprobada “;
18         Sino
19             Mostrar “Materia desaprobada: No alcanza el promedio”;
20         Fin Si
    Sino
        Mostrar “Materia desaprobada: No aprobó ambos exámenes “;
    Fin Si
FinProceso
```



Como se ve en el ejemplo el segundo Si se evaluará sólo si se cumple la condición del primero. De igual forma se podría poner otra estructura dentro del Sino.



Bibliografía utilizada

Luis Joyanes Aguilar, Ignacio Zahonero Martínez. Programación en C. Segunda Edición. España: McGRAW-HILL/INTERAMERICANA DE ESPAÑA. S.A.U., 2005. ISBN-84: 481-9844-1.

- Brian W. Kernighan, Rob Pike. La práctica de la programación. Pearson Educación. Méjico (2000)
- Debugging with CodeBlocks:
http://wiki.codeblocks.org/index.php?title=Debugging_with_Code::Blocks