

Guía #6

Tema: “Estructuras de Repetición en C# [For]”.

Objetivos

- Utilizar las estructuras de repetición para ejecutar instrucciones repetidas en una aplicación.
- Solucionar problemas con programas que integren la estructura de selección y estructuras repetitivas **for**.

Introducción

Los bucles o ciclos repiten una sentencia o una secuencia de sentencias un número de veces. A dichas sentencias se les conoce como **cuerpo del bucle**. A cada repetición del bucle se le llama **iteración**.

El bucle for

Si conocemos exactamente la cantidad de veces que necesitamos repetir un conjunto de sentencias, entonces podemos usar un bucle FOR.

En general, los bucles controlados por contador requieren:

- a) Una variable de control o contador
- b) Un valor inicial para el contador
- c) Un valor final para el contador
- d) Una condición para verificar si la variable de control alcanzo su valor final
- e) Un valor de incremento (o decremento) con el cual se modifica la variable de control en cada bucle

La forma general del bucle FOR es:

```
for (inicialización; condición_iteración; incremento_o_decremento)
{
    Sentencia 1....;
    Sentencia 2...;
    ...
    Sentencia n;
}
```

Por ejemplo tenemos:

```
for (x=1;x<=5;x=x+1)
{
.....
}
```

❖ **Contadores:** un contador es una variable casi siempre de tipo entero cuyo valor se incrementa o decremente en cada repetición del bucle. Es habitual llamar a esta variable “cont” (contador) o “i” (índice). El contador suele utilizarse de este modo:

1. Se inicializa antes de que comience el bucle. Es decir, se le da un valor inicial-

Por ejemplo: cont = 1

2. Se modifica dentro del cuerpo del bucle. Lo más habitual es que se incremente su valor en una unidad:

Por ejemplo: cont = cont + 1;

Esto quiere decir que el valor de la variable **cont** se incrementa en una unidad y es asignado de nuevo a la variable contador. Es decir, si **cont** valía 1 antes de esta instrucción, **cont** valdrá 2 después. Otra forma típica del contador es: cont = cont - 1.

Formas de incrementar los contadores:

Primera Forma	Segunda Forma	Tercera Forma
cont = cont + 1	cont++	cont += 1

❖ **Acumuladores:** Las variables acumuladores tienen la misión de almacenar los resultados sucesivos, es decir, de acumular resultados.

Las variables acumuladores también deben ser inicializadas. Si llamamos “acum” a un acumulador, escribiremos antes de iniciar el bucle algo como esto: acum = 0.

Por supuesto, el valor inicial puede cambiar, dependiendo de la naturaleza del problema. Más tarde, en el cuerpo del bucle, la forma en la que solemos encontrarla es:

acum = acum + N;

Otra manera de escribir esta sentencia sería: acum += N.

Material y Equipo

- Guía de laboratorio No. 6.
- Computadora con Visual Studio 2013 o superior.
- Dispositivo de almacenamiento (USB).

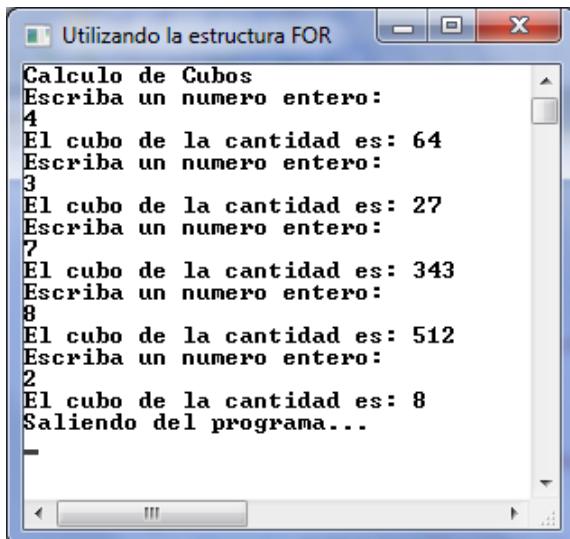
Procedimiento

Ejemplo1

Realizar un programa que nos permita digitar 5 números enteros y nos permita ver en la pantalla el cubo de cada uno de ellos. Hacer uso de una estructura for.

```
1 static void Main(string[] args)
2 {
3     Console.Title = "Utilizando la estructura FOR";
4     Double cubo;
5     int cantidad;
6     int i;
7     Console.WriteLine("Calculo de Cubos");
8     for (i = 1; i <= 5; i++)
9     {
10         Console.WriteLine("Escriba un numero entero: ");
11         cantidad = int.Parse(Console.ReadLine());
12         cubo = Math.Pow(cantidad,3);
13         Console.WriteLine("El cubo de la cantidad es: " + cubo);
14     }
15     Console.WriteLine("Saliendo del programa...");
16     Console.ReadKey();
17 }
```

Corrida del ejercicio:



Ejemplo2

Hacer uso de una estructura **for**. Realizar un programa que permita calcular el promedio de un alumno que tiene 5 calificaciones en la materia de Calculo Integral.

```
1 static void Main(string[] args)
2 {
3     Console.ForegroundColor = ConsoleColor.Black;
4     Console.BackgroundColor = ConsoleColor.White;
5     Console.Clear();
6     Console.Title = ("Promedio de Notas de Calculo Diferencial");
7     String alumno;
8     int x;
9     Double nota, sum, prom;
10    sum = 0;
11    Console.Write("Digitar el nombre del estudiante: ");
12    alumno = Console.ReadLine();
13    Console.Clear();
14    Console.WriteLine("El nombre del estudiante es: {0}", nombre);
15    Console.WriteLine("\n");
16    for (x = 1; x <= 5; x = x + 1)
17    {
18        Console.Write("Digitar la nota {0} : ", x);
19        nota = Double.Parse(Console.ReadLine());
20        sum = sum + nota;// aki se van sumando las notas (acumulador)
21    }
22    // procesos afuera del ciclo for
23    prom = sum / 5;
24    Console.WriteLine("\n");
25    Console.WriteLine("El promedio del alumno {0} es de: {1}", alumno,
Math.Round(prom,3));
26    Console.WriteLine("\n");
27    Console.WriteLine("-->Fin del programa");
28    Console.ReadKey();
29 }
```

No. Corrida	Datos entradas	Resultado
1	alumno: pepito nota1= 5.5 nota2 = 4.5 nota3 = 7.5 nota4 = 9.5 nota5 = 2.5	El promedio de es de
2	alumno: margarita nota1= 9.5 nota2 = 8.5 nota3 = 8.5 nota4 = 9.5 nota5 = 9.5	El promedio de es de

Ejemplo3

Leer 4 números enteros cualesquiera e imprimir solamente los números positivos

```
1 static void Main(string[] args)
2 {
3     Console.ForegroundColor = ConsoleColor.Black;
4     Console.BackgroundColor = ConsoleColor.White;
5     Console.Clear();
6     Console.Title = ("Numeros Positivos");
7     int y;
8     Double num;
9     for (y = 1; y <= 4; y = y + 1)
10    {
11        Console.Write("Digitar el numero {0} : ", y);
12        num = Double.Parse(Console.ReadLine());
13        if (num > 0)
14        {
15            Console.WriteLine(" El numero: {0} es positivo", num);
16        }
17    }
18    Console.WriteLine("\n");
19    Console.WriteLine("-->Fin del programa");
20    Console.ReadKey();
21 }
```

No. Corrida	Datos entradas	Resultado
1	numero1 = 9 numero2 = -5 numero3 = -3 numero4 = -8	
2	numero1 = 6 numero2 = 0 numero3 = -2 numero4 = -7	

Ejemplo4

Ejecute el siguiente código de programa en su cuaderno y escriba la salida en el cuadro que está debajo del código.

```
1 static void Main(string[] args)
2 {
3     Console.Title = "Programa extra del ciclo FOR";
4     int x, y;
5     Console.WriteLine("Programa que imprime una secuencia de numeros");
6     Console.WriteLine("\n");
7     for (x=1;x<10;x=x+2)
```

```
8     {
9         y=x+5;
10        Console.WriteLine(x + "," + y +"," );
11    }
12    Console.ReadKey();
13 }
14 }
```

Ejemplo5

Digitar una lista de 7 números enteros desde el teclado. Informar el mayor, el menor y la sumatoria de los que se encuentran entre 100 y 500 inclusive.

```
1 static void Main(string[] args)
2 {
3     Console.ForegroundColor = ConsoleColor.Black;
4     Console.BackgroundColor = ConsoleColor.White;
5     Console.Clear();
6     Console.Title = ("Mayor, Menor y Sumatoria");
7     int i;
8     Int32 n, max=0, min=0, sum=0;
9     Console.WriteLine("Ingrese 7 enteros:");
10    for (i=1;i<=7;i=i+1)
11    {
12        n = Int32.Parse(Console.ReadLine());
13        if (i == 1)
14        {
15            max = n;
16            min = n;
17        }
18        if (n > max)
19        {
20            max = n;
21        }
22        if (n < min)
23        {
24            min = n;
25        }
26        if ((n >= 100) && (n <= 500)) // los que se estan en el rango
```

```

27     {
28         sum = sum + n;
29     }
30 }
31 Console.WriteLine("\nEl numero mayor es: " + max);
32 Console.WriteLine("\nEl numero menor es: " + min);
33 Console.WriteLine("\nLa suma de los que estan entre 100 y 500 es:" + sum);
34 Console.WriteLine("\n");
35 Console.WriteLine("-->Fin del programa");
36 Console.ReadKey();
37 }

```

No. Corrida	Datos entradas	Resultado
1	n1 = n2 = n3 = n4 = n5 = n6 = n7 =	
2	n1 = n2 = n3 = n4 = n5 = n6 = n7 =	

Análisis de Resultados

1. Desarrollar un programa que capture un número desde teclado e imprima su tabla de multiplicar.
2. Escribir un programa que pida ingresar coordenadas (x,y) que representen puntos en el plano. Informar cuantos puntos se han ingresado en el primer, segundo, tercer y cuarto cuadrante. Al comenzar el programa se pide que se ingrese la cantidad de pares de puntos a procesar.
3. Realizar un programa que lea los datos de “n” triángulos e informar:
 - a. De cada uno de ellos, que tipo de triangulo es: equilátero (tres lados iguales), isósceles (dos lado iguales?, o escaleno (ningún lado igual)
 - b. Cantidad de triángulos de cada tipo
 - c. Tipo de triangulo que posee menor cantidad
4. Se realiza la entrada de 10 numeros enteros. Se desea conocer:
 - a. La cantidad de numeros negativos
 - b. La cantidad de números positivos
 - c. La cantidad de multiplos de 15
 - d. El valor acumulado de los números ingresados que son pares

5. Desarrollar un programa que permita imprimir en pantalla los números del 1 al 30, excepto el 25.

Investigación Complementaria

1. Desarrollar un programa, que dados como datos 10 números enteros, obtenga la suma de los números impares y el promedio de los números pares.
2. Escribir un programa en C# que determine de un conjunto de 10 números enteros, ¿Cuántos son menores de 5, mayores de 85 y cuantos entre 50 y 150?
3. Suponga que se proporciona una secuencia de “X” números enteros tales como: 6,-4, 5, 29, 6, 43,-567,-23, 6, 90, etc. y se desea contar e imprimir ¿cuántos números 6 hay, cuántos son pares y cuántos son positivos?
4. Escribir un programa que lea 10 notas de alumnos y nos informe cuántos tienen notas mayores o iguales a 7 y cuántos menores.
5. Confeccionar un programa que lea “n” pares de datos, cada par de datos corresponde a la medida de la base y la altura de un triángulo.

El programa deberá informar:

- a . De cada triángulo la medida de su base, su altura y su superficie.
- b . La cantidad de triángulos cuya superficie es mayor a 12.
6. Escribir un programa que lea 10 números enteros y luego muestre cuántos valores ingresados fueron múltiplos de 3 y cuántos de 5. Debemos tener en cuenta que hay números que son múltiplos de 3 y de 5 a la vez.
7. En una empresa de computadoras, los salarios de los empleados se van a aumentar según el contrato actual:

Contrato	Aumento %
\$ 0.0 - \$9,000.00	20
\$9,001 – \$15,000	10
\$15,001 – \$20,000	5
Más de \$20,000	0

Calcular el nuevo salario del empleado, tomando en cuenta que son 4 empleados en el área de ventas.

Bibliografía

- Deitel, Harvey M. y Paul J. Deitel, Cómo Programar en C#, Segunda Edición, México, 2007.