

INSTITUTO SUPERIOR DE FORMACIÓN TÉCNICA N° 177



Algoritmos y Estructuras de Datos 1

Estructura de Repetición for

Prof. Lic. Walter Carnero

wdcarnero@gmail.com

APUNTE: AyEDD04 – ESTRUCTURA DE REPETICIÓN FOR

LA ESTRUCTURA DE REPETICIÓN FOR

La estructura de repetición for realiza el incremento o decremento de la variable de control automáticamente. A esta estructura de repetición controlada mediante un contador se la suele llamar repetición definida ya que sabemos de antemano el número de veces que se ejecutará. La variable de control se incrementa (generalmente de uno en uno) cada vez que el grupo de instrucciones a repetir se ejecuta. Cuando el valor de la variable de control indica que el número de repeticiones se ha alcanzado (mediante una condición), el ciclo termina y el programa continúa con la ejecución de la instrucción que se encuentra después del ciclo de repetición.

La estructura de repetición for se caracteriza por:

1. El nombre de la variable de control (contador de ciclo).
2. El valor inicial de la variable de control.
3. El incremento o decremento mediante el cual se modifica la variable de control cada vez que se ejecuta un ciclo.
4. La condición que evalúa el valor final de la variable de control.

Las características citadas del ciclo for se implementan en la estructura mediante tres partes, siendo el diagrama de flujo el siguiente:

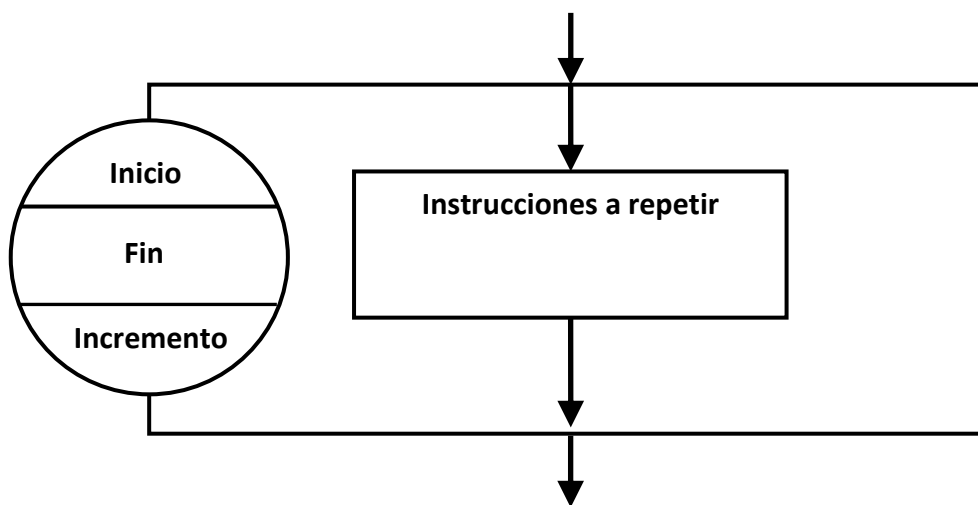


Figura 1

La sintaxis o la forma en que escribimos el ciclo for en el lenguaje C# es la siguiente

```
for (inicio, condición, incremento)
{
    Secuencia de instrucciones;
}
```

El siguiente algoritmo permite mostrar por pantalla los números desde el 1 hasta el 40 inclusive separados por un guion.

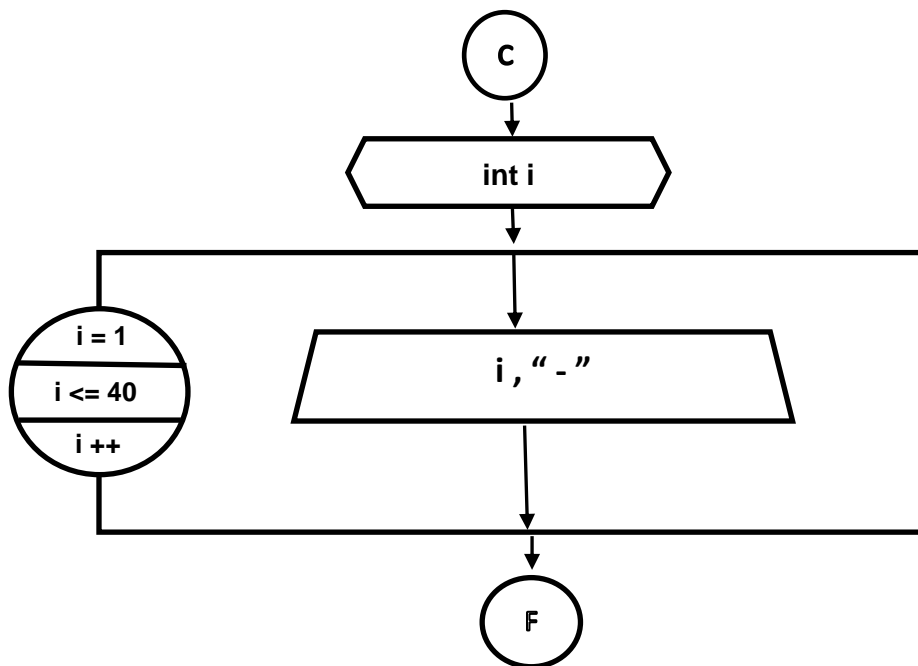


Figura 2

La salida que se observa por pantalla luego de ejecutar el algoritmo de la *figura 2*

1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40

Como se puede visualizar la estructura de repetición se ejecuta mientras la variable *i* es menor o igual a 40.

Sobre los diagramas de Jackson

Otra posibilidad es analizar la estructura de repetición en base a los diagramas de Jackson. Para el ejercicio propuesto con anterioridad, el resultado obtenido sería el siguiente:

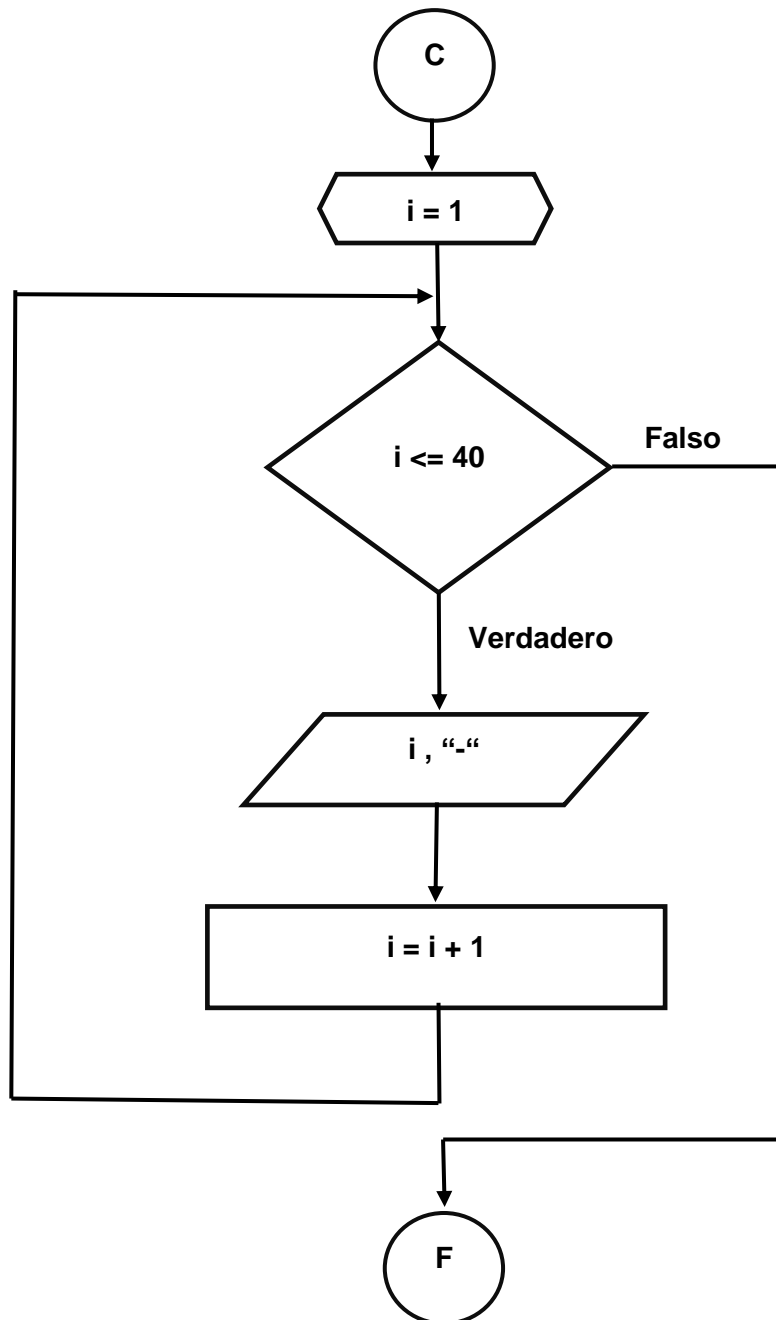


Figura 3

Sobre el código

Se detalla a continuación la codificación aplicando la sintaxis del lenguaje C# para el ejercicio propuesto, el programa que presenta la solución es el siguiente:

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```



```
namespace ConsoleApplication
{
    class Program
    {
        static void Main(string[] args)
        {
            int i;
            for(i = 1; i <= 40; i++)
            {
                Console.Write(i + "-");
            }
            Console.ReadKey();
        }
    }
}
```

A continuación se muestran algunos ejemplos de aplicación mediante la sintaxis de C#

Ejemplo 1: Programa que permite determinar el factorial de un número ingresado por el usuario:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication6
{
    class Program
    {
        static void Main(string[] args)
        {
            int valor = 0, i;
            Int32 fac = 1;
            Console.Write("Ingrese un número: ");
            valor = int.Parse(Console.ReadLine());
            for (i = 1; i <= valor; i++)
            {
                fac = fac * i;
            }
            Console.Write("El factorial de " + valor + " es " + fac);
            Console.ReadKey();
        }
    }
}
```



Ejemplo 2: Programa que permite mostrar por pantalla el abecedario desde la Z a la A.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication6
{
    class Program
    {
        static void Main(string[] args)
        {
            char letra;

            for (letra = 'Z'; letra >= 'A'; letra--)
            {
                Console.WriteLine(letra);
            }
            Console.ReadKey();
        }
    }
}
```

Ejemplo 3: Se muestran por pantalla los números desde el 100 hasta el 5 en forma descendente con decremento de 5.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication6
{
    class Program
    {
        static void Main(string[] args)
        {
            for (int i = 100 ; i >= 5; i = i - 5)
            {
                Console.Write(i + " ");
            }
            Console.ReadKey();
        }
    }
}
```