

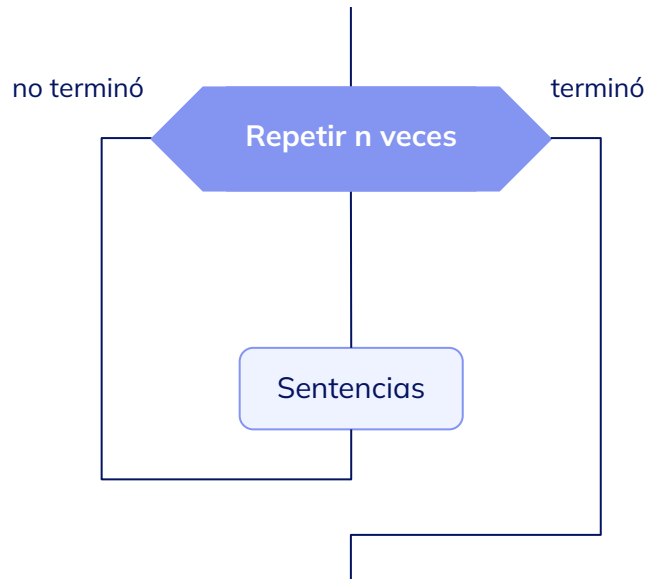
# Introducción a C# .NET

Módulo 3

# Instrucción for

El significado de la instrucción es el siguiente: se ejecutan las instrucciones de <inicialización>, que suelen usarse para definir e inicializar variables que luego se usarán en <instrucciones> (sentencias).

Luego se evalúa <condición>, y si es falsa se continúa ejecutando la instrucción siguiente al **for**; mientras que si es cierta se ejecutan las <instrucciones> indicadas, luego se ejecutan las instrucciones de <modificación> (que como su nombre indica suelen usarse para modificar los valores de variables que se usen en <instrucciones>), y luego se reevalúa <condición> repitiéndose el proceso hasta que ésta última deje de ser verdadera.



# for

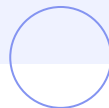
La **instrucción for** ejecuta un ciclo durante un número determinado de veces.

Consta de tres partes:

- La sección de **inicialización**, que se ejecuta al comienzo del ciclo.
- La **condición**, que se evalúa al entrar al ciclo y luego al comienzo de cada iteración.
- La sección de **modificación**, que se ejecuta al final de cada iteración, y previene que no se produzca un ciclo infinito (se ejecuta al final de cada iteración).

Su sintaxis es:

```
for(<inicialización>; <condición>; <modificación>)  
{  
    <instrucciones>  
}
```



En `<inicialización>` se puede incluir, en realidad, cualquier número de instrucciones que no tienen porqué ser utilizadas exclusivamente para inicializar variables o modificarlas, aunque lo anterior sea su uso más habitual. En caso de ser varias instrucciones, se han de separar mediante comas (,) ya que el carácter de punto y coma (;) habitualmente se usa en el **for** para separar los bloques de `<inicialización>`, `<condición>` y `<modificación>`.

Además, la instrucción nula no se puede usar en este caso ni tampoco pueden combinarse definiciones de variables con instrucciones de otros tipos.

Con `<modificación>` pasa algo similar, ya que se puede incluir código que nada tenga que ver con modificaciones, pero en este caso no se pueden incluir definiciones de variables.

Como en el resto de las instrucciones hasta ahora vistas, en `<instrucciones>` puede haber tanto una única instrucción como un bloque de instrucciones. Además, las variables que se definan en `<inicialización>` serán visibles sólo dentro de esas `<instrucciones>`.

El código que veremos a continuación es equivalente al ejemplo de **while** ya visto, sólo que hace uso del **for** para compactar más su código.

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

namespace For
{
    class NumerosHasta5
    {
        static void Main(string[] args)
        {
            for (int numero = 1; numero <= 5; numero++)
            {
                Console.WriteLine(numero);
            }
            Console.ReadKey();
        }
    }
}
```

Al igual que con **while**, dentro de las <instrucciones> del **for** también es posible incluir instrucciones **continue**; y **break**; que puedan alterar el funcionamiento normal del ciclo.



**¡Sigamos  
trabajando!**