

Introducción a C# .NET

Módulo 2



Control de flujo de programa

Introducción

Como se ha visto hasta ahora, los programas ejecutan las líneas de código en orden secuencial, una instrucción detrás de otra. Sin embargo, hay muchas situaciones en las que es preciso alterar ese orden, o bien puede ocurrir que sea necesario que se efectúen una serie de operaciones que pueden ser distintas en otras circunstancias.

Por ejemplo, si el programa pide una clave de acceso de un usuario, deberá continuar con la ejecución normal en caso que la clave introducida sea correcta, y deberá avisar al usuario y salir del programa en caso contrario.


Pues bien: para todas estas cuestiones que, por otra parte, son muy frecuentes, tenemos las **instrucciones de control de flujo de programa**. Estas instrucciones de control son **compartidas por todos los lenguajes de programación**, puede haber ligeras diferencias en los nombres de las instrucciones, pero su propósito y semántica es la misma.



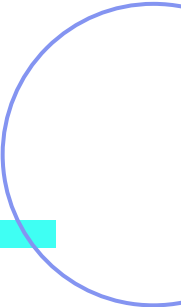


En C# contamos con varios tipos:

Las **instrucciones de decisión** (if...else, switch) son aquellas que permiten ejecutar bloques de instrucciones sólo si se cumple una determinada condición, o si no se cumple esa condición.

Las **instrucciones cíclicas o iterativas** (while, do, for, foreach) son aquellas que permiten ejecutar repetidas veces una instrucción o un bloque de instrucciones mientras se cumpla una condición. Es decir, permiten definir ciclos donde ciertas instrucciones se ejecuten varias veces.



Las **instrucciones de salto** (break, continue, return, goto) permiten variar el orden normal en que se ejecutan las instrucciones de un programa, que consiste en ejecutarlas una tras otra en el mismo orden en que se hubiesen escrito en el código fuente. Si bien se mencionan, no es buena práctica usar las instrucciones de salto **continue** y **goto**.



Instrucción *if...else*

La **instrucción if** permite ejecutar ciertas instrucciones sólo si se da una determinada condición. Su sintaxis de uso es la sintaxis:

```
if(<condición>){  
    <instruccionesIf>  
}  
else{  
    <instruccionesElse>  
}
```

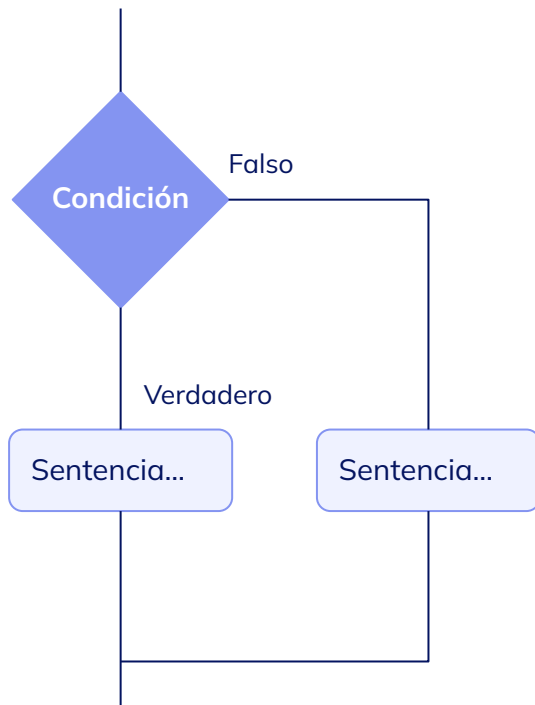


El significado de la instrucción que vimos es el siguiente:

- Se evalúa la expresión <condición>, que ha de devolver un valor lógico **true** o **false**.
- Si la condición es cierta (devuelve **true**) se ejecutan las <instruccionesIf>, y si la condición es falsa (**false**) se ejecutan las <instruccionesElse>.

La rama **else** es opcional, y si se omite y la condición es falsa se seguirá ejecutando a partir de la instrucción siguiente al **if**.

En realidad, tanto <instruccionesIf> como <instruccionesElse> pueden ser una única instrucción o un bloque de instrucciones.



Un ejemplo de aplicación de esta instrucción podría ser:

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

namespace Saludo_if
{
    class Saludo
    {
        static void Main(string[] args)
        {
            string Nombre = "";
            Console.WriteLine("Ingrese un nombre de no más de 10 caracteres: ");
            Nombre = Console.ReadLine();
        }
    }
}
```

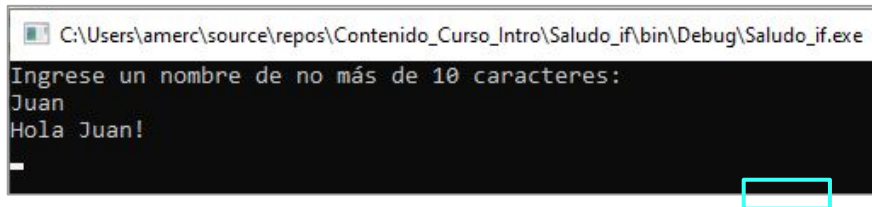
...

```
if (Nombre.Length <= 10)
{
    Console.WriteLine("Hola " + Nombre + "!");
}
else
{
    Console.WriteLine("El nombre ingresado contiene más de 10 caracteres !");
}

Console.ReadKey();
}
```

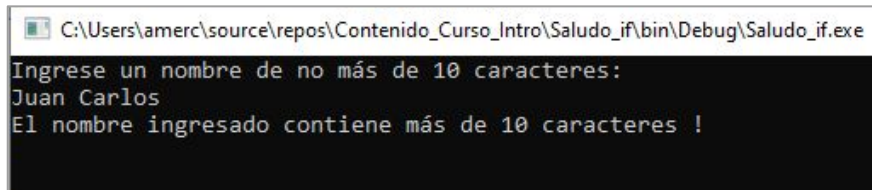

Si ejecutamos el programa anterior:

Ingresado un nombre de hasta 10 caracteres, veremos que el mensaje que se muestra es un mensaje de bienvenida: **“Hola”** concatenando (uniendo) el nombre de hasta 10 caracteres ingresado.



```
C:\Users\amerc\source\repos\Contenido_Curso_Intro\Saludo_if\bin\Debug\Saludo_if.exe
Ingresa un nombre de no más de 10 caracteres:
Juan
Hola Juan!
```

Pero si ingresamos un nombre de más de 10 caracteres, mostrará un mensaje indicando esta situación.



```
C:\Users\amerc\source\repos\Contenido_Curso_Intro\Saludo_if\bin\Debug\Saludo_if.exe
Ingresa un nombre de no más de 10 caracteres:
Juan Carlos
El nombre ingresado contiene más de 10 caracteres !
```

**¡Sigamos
trabajando!**