

Universidad Rafael Landívar.
Facultad de Ingeniería.
Ingeniería en Informática y Sistemas.
Pensamiento Computacional Sección 08
Ing. Luis Enrique Aguilar Rojas

CONDICIONALES PANTALLAZOS

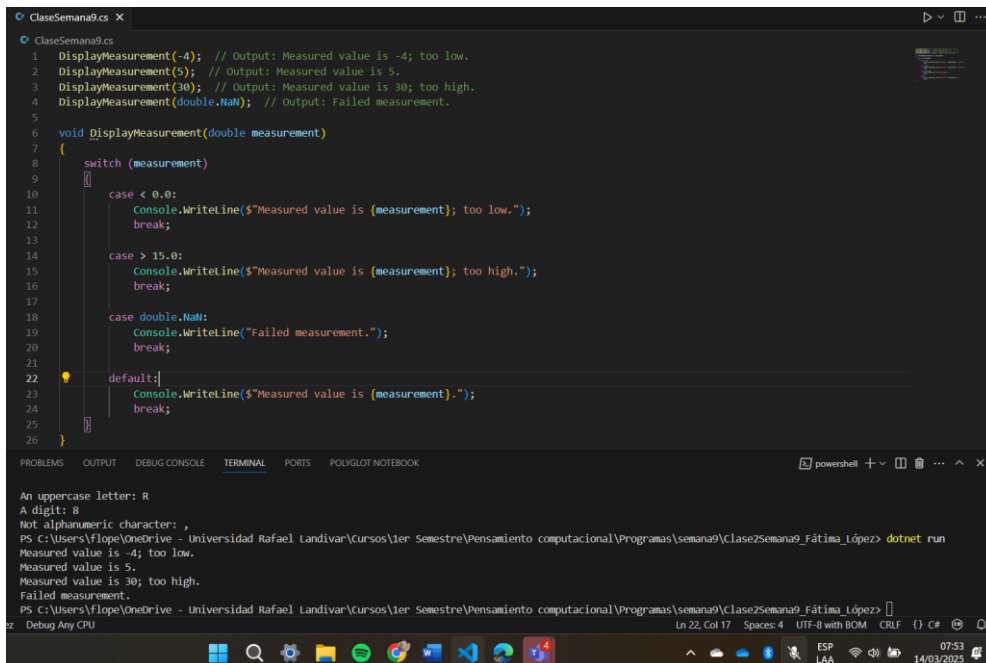
Actividad 9 – Semana 9

Estudiante: Fátima Joanna López Quiñonez
Carné: 1088825

Guatemala, 14 de marzo de 2025

2. Else if

3. Switch



```
1 DisplayMeasurement(-4); // Output: Measured value is -4; too low.
2 DisplayMeasurement(5); // Output: Measured value is 5.
3 DisplayMeasurement(30); // Output: Measured value is 30; too high.
4 DisplayMeasurement(double.NaN); // Output: Failed measurement.
5
6 void DisplayMeasurement(double measurement)
7 {
8     switch (measurement)
9     {
10         case < 0.0:
11             Console.WriteLine($"Measured value is {measurement}; too low.");
12             break;
13
14         case > 15.0:
15             Console.WriteLine($"Measured value is {measurement}; too high.");
16             break;
17
18         case double.NaN:
19             Console.WriteLine("Failed measurement.");
20             break;
21
22         default:
23             Console.WriteLine($"Measured value is {measurement}.");
24             break;
25     }
26 }
```

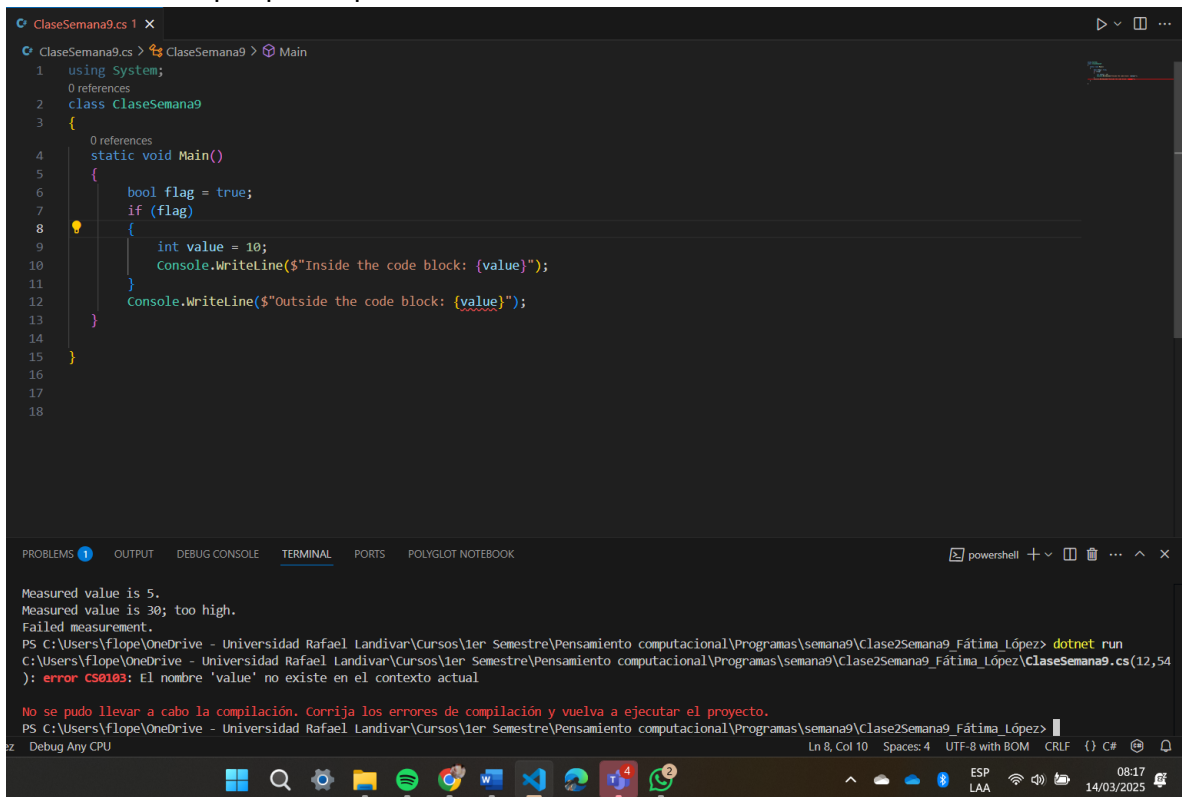
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POLYGLOT NOTEBOOK

An uppercase letter: R
A digit: 8
Not alphanumeric character: ,
PS C:\Users\flope\OneDrive - Universidad Rafael Landivar\cursos\1er Semestre\Pensamiento computacional\Programas\semana9\Clase2Semana9_Fátima López> dotnet run
Measured value is -4; too low.
Measured value is 5.
Measured value is 30; too high.
Failed measurement.
PS C:\Users\flope\OneDrive - Universidad Rafael Landivar\cursos\1er Semestre\Pensamiento computacional\Programas\semana9\Clase2Semana9_Fátima López> |

Ln 22, Col 17 Spaces: 4 UTF-8 with BOM CRLF {} C# 07:53 14/03/2025

4. Error de iniciación de variable *value*.

En este código se tiene un error de compilación puesto que nunca se inicializa el valor de *value* afuera del *if*. Por lo que al imprimir la variable afuera de esta, se tiene un error porque no podrá devolver nada.



```
1 using System;
2 class ClaseSemana9
3 {
4     0 references
5     static void Main()
6     {
7         bool flag = true;
8         if (flag)
9         {
10             int value = 10;
11             Console.WriteLine($"Inside the code block: {value}");
12         }
13         Console.WriteLine($"Outside the code block: {value}");
14     }
15 }
16
17
18
```

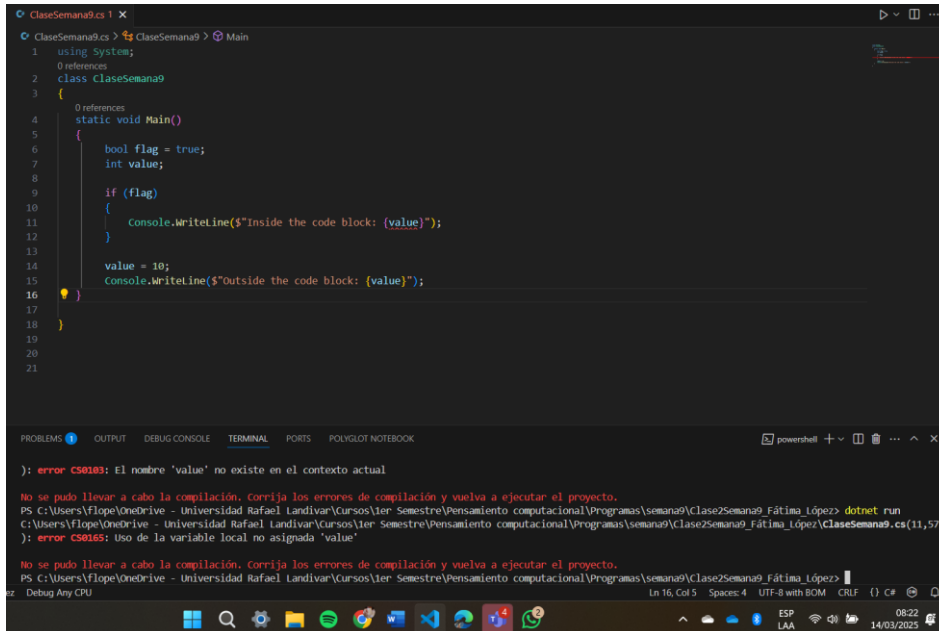
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS POLYGLOT NOTEBOOK

Measured value is 5.
Measured value is 30; too high.
Failed measurement.
PS C:\Users\flope\OneDrive - Universidad Rafael Landivar\cursos\1er Semestre\Pensamiento computacional\Programas\semana9\Clase2Semana9_Fátima López> dotnet run
C:\Users\flope\OneDrive - Universidad Rafael Landivar\cursos\1er Semestre\Pensamiento computacional\Programas\semana9\Clase2Semana9_Fátima López> ClaseSemana9.cs(12,54): error CS0103: El nombre 'value' no existe en el contexto actual
No se pudo llevar a cabo la compilación. Corrija los errores de compilación y vuelva a ejecutar el proyecto.
PS C:\Users\flope\OneDrive - Universidad Rafael Landivar\cursos\1er Semestre\Pensamiento computacional\Programas\semana9\Clase2Semana9_Fátima López> |

Ln 8, Col 10 Spaces: 4 UTF-8 with BOM CRLF {} C# 08:17 14/03/2025

5. Error de iniciación variable *value*.

En este caso el error se encuentra en que la condición if no tiene inicializado ningún valor para la variable *value*, puesto que se hace hasta que se sale del bloque de código. Con ambos problemas anteriores, se concluye que siempre se debe inicializar la(s) variable(s) a utilizar para evitar errores de compilación.



```
1 using System;
2 class ClaseSemana9
3 {
4     static void Main()
5     {
6         bool flag = true;
7         int value;
8
9         if (flag)
10        {
11            Console.WriteLine($"Inside the code block: {value}");
12        }
13
14        value = 10;
15        Console.WriteLine($"Outside the code block: {value}");
16    }
17 }
18
19
20
21
```

error CS0163: El nombre 'value' no existe en el contexto actual

No se pudo llevar a cabo la compilación. Corrija los errores de compilación y vuelva a ejecutar el proyecto.

PS C:\Users\Flope\OneDrive - Universidad Rafael Landívar\Cursos\1er Semestre\Pensamiento computacional\Programas\semana9\Clase2Semana9_Fátima López> dotnet run

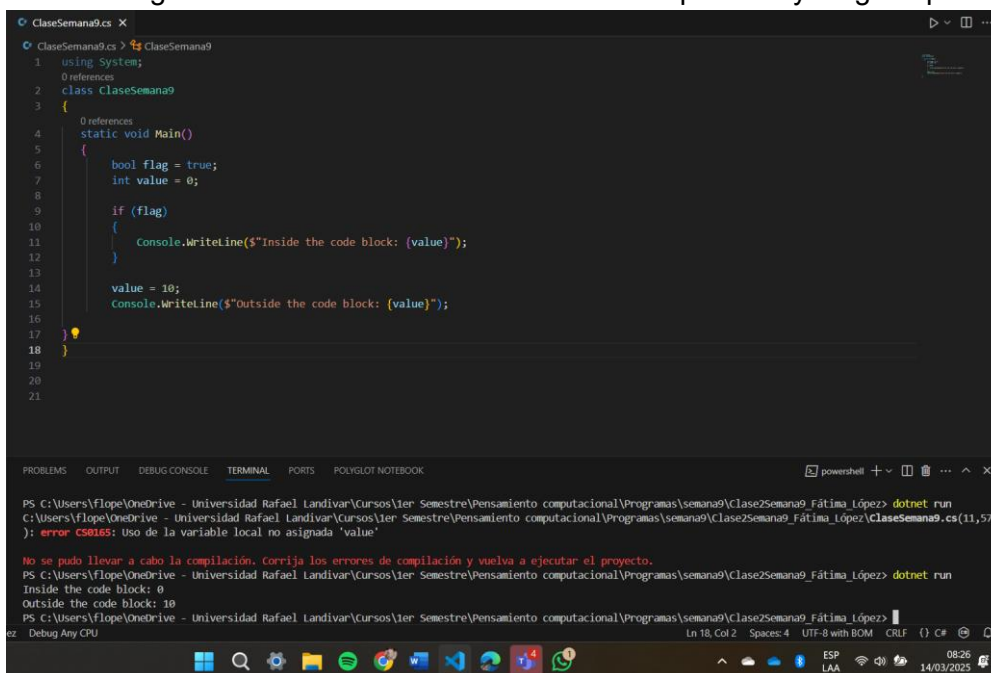
C:\Users\Flope\OneDrive - Universidad Rafael Landívar\Cursos\1er Semestre\Pensamiento computacional\Programas\semana9\Clase2Semana9_Fátima López\ClaseSemana9.cs(11,57): error CS0163: uso de la variable local no asignada 'value'

No se pudo llevar a cabo la compilación. Corrija los errores de compilación y vuelva a ejecutar el proyecto.

PS C:\Users\Flope\OneDrive - Universidad Rafael Landívar\Cursos\1er Semestre\Pensamiento computacional\Programas\semana9\Clase2Semana9_Fátima López>

6. Inicialización de variable *value*.

En este caso, la impresión de las variables muestra valores diferentes porque antes de pasar por el bloque del if se inicializa con valor cero y lo evalúa con ese valor. Luego de esa evaluación se cambia el valor por diez y luego imprime ese valor.



```
1 using System;
2 class ClaseSemana9
3 {
4     static void Main()
5     {
6         bool flag = true;
7         int value = 0;
8
9         if (flag)
10        {
11            Console.WriteLine($"Inside the code block: {value}");
12        }
13
14        value = 10;
15        Console.WriteLine($"Outside the code block: {value}");
16    }
17 }
18
19
20
21
```

PS C:\Users\Flope\OneDrive - Universidad Rafael Landívar\Cursos\1er Semestre\Pensamiento computacional\Programas\semana9\Clase2Semana9_Fátima López> dotnet run

C:\Users\Flope\OneDrive - Universidad Rafael Landívar\Cursos\1er Semestre\Pensamiento computacional\Programas\semana9\Clase2Semana9_Fátima López\ClaseSemana9.cs(11,57): error CS0163: uso de la variable local no asignada 'value'

No se pudo llevar a cabo la compilación. Corrija los errores de compilación y vuelva a ejecutar el proyecto.

PS C:\Users\Flope\OneDrive - Universidad Rafael Landívar\Cursos\1er Semestre\Pensamiento computacional\Programas\semana9\Clase2Semana9_Fátima López> dotnet run

Inside the code block: 0

Outside the code block: 10

PS C:\Users\Flope\OneDrive - Universidad Rafael Landívar\Cursos\1er Semestre\Pensamiento computacional\Programas\semana9\Clase2Semana9_Fátima López>

7. Errores de rutas de acceso de ejecución

En el ejemplo de código 1 (*code sample 1*), existe un error porque la inicialización de la variable está dentro del `if` y tiene un límite para poder accederse en todo el código (subrayado en **verde**), esto se concluyó en ejemplos anteriores. Sin embargo, en el ejemplo de código 2 (*code sample 2*), no muestra este error por una única diferencia, el `if` se evalúa usando **true** (subrayado en **amarillo**). En el ejemplo 1, se usaba una variable booleana con valor *true*, sin embargo, el compilador lo interpreta como “Bueno, es de valor verdadero, pero este puede ser que cambie porque es una variable”, así que reporta un error porque si llega ser falso, al imprimir el valor de *value* no obtendrá nada porque nunca se inicializó el valor. En el ejemplo 2 no pasa esto, porque se utiliza el constante `true` dentro del `if`, y el `true` siempre será `true`, por lo que lo interpreta como “Esta condición siempre se ejecutará porque tiene valor de verdadero”.

```
c# Copiar

// Code sample 1
bool flag = true;
int value;

if (flag)
{
    value = 10;
    Console.WriteLine($"Inside the code block: {value}");
}

Console.WriteLine($"Outside the code block: {value}");
```

```
c# Copiar

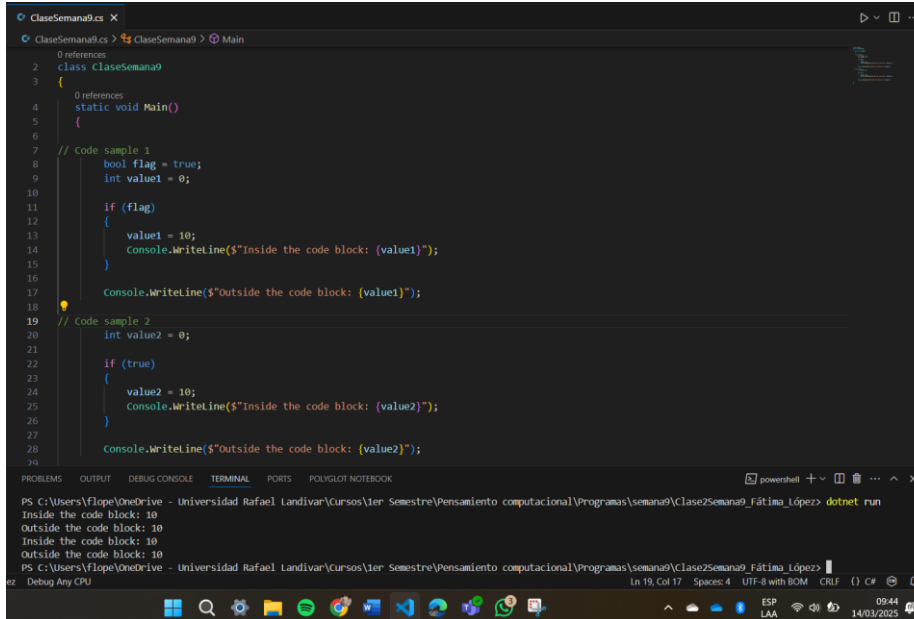
// Code sample 2
int value;

if (true)
{
    value = 10;
    Console.WriteLine($"Inside the code block: {value}");
}

Console.WriteLine($"Outside the code block: {value}");
```

8. Corrección del ejemplo 1 y ejecución correcta de ambos casos

Como se muestra en el código, para que el ejemplo 1 sí se llegue a ejecutar se debe colocar su iniciación de valor 0. Al imprimirse, este cambia a 10 porque se realiza una reasignación de dicho valor. En el ejemplo 2 no se realizó ningún cambio y sí se ejecutó por la explicación del pantallazo anterior.



```
1 0 references
2 class ClaseSeman9
3 {
4     0 references
5     static void Main()
6     {
7         // Code sample 1
8         bool flag = true;
9         int value1 = 0;
10
11         if (flag)
12         {
13             value1 = 10;
14             Console.WriteLine($"Inside the code block: {value1}");
15         }
16
17         Console.WriteLine($"Outside the code block: {value1}");
18
19         // Code sample 2
20         int value2 = 0;
21
22         if (true)
23         {
24             value2 = 10;
25             Console.WriteLine($"Inside the code block: {value2}");
26         }
27
28         Console.WriteLine($"Outside the code block: {value2}");
29
30     }
31 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POLYGLOT NOTEBOOK

PS C:\Users\Flope\OneDrive - Universidad Rafael Landívar\Cursos\1er Semestre\Pensamiento computacional\Programas\semana9\Clase2Seman9_Fátima_López> dotnet run

Inside the code block: 10
Outside the code block: 10
Inside the code block: 10
Outside the code block: 10

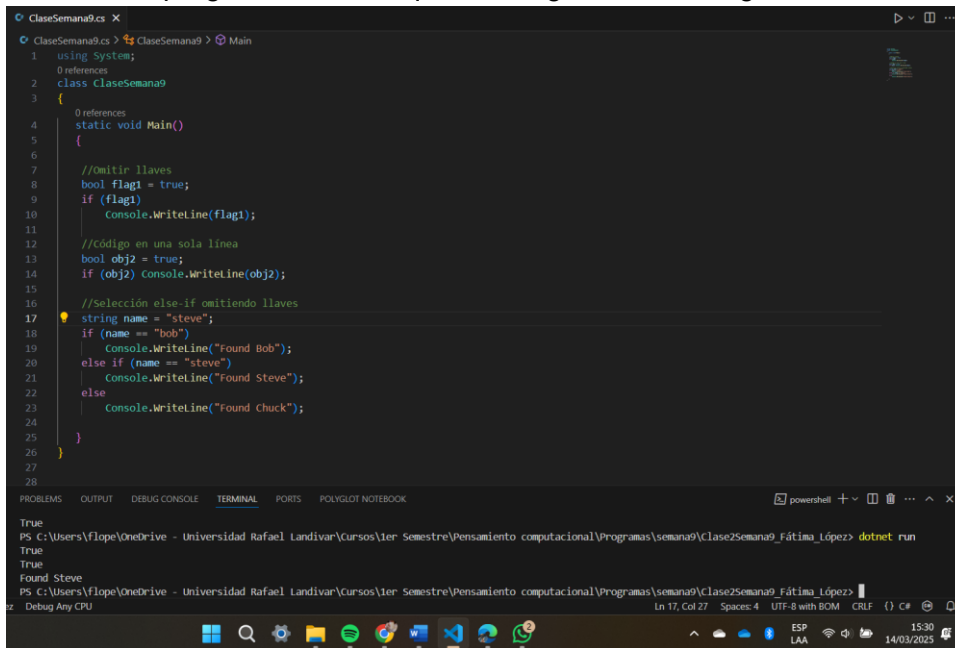
PS C:\Users\Flope\OneDrive - Universidad Rafael Landívar\Cursos\1er Semestre\Pensamiento computacional\Programas\semana9\Clase2Seman9_Fátima_López>

Debug Any CPU

Ln 19, Col 17 Spaces: 4 UTF-8 with BOM CRLF () C# 09:44 14/01/2025

9. Quitar las llaves y usar líneas únicas en un bloque de selección if

En todos los casos, existe una única línea de código enumerada después de las instrucciones if o if-elseif-else, se puede quitar las llaves y el salto de línea. Microsoft recomienda mantener una coherencia de uso u omisión de llaves y saltos de línea, por reglas de convención. Siempre es recomendable el uso de llaves, sin embargo, si el programador cree que el código será más legible sin ellos, se pueden omitir.



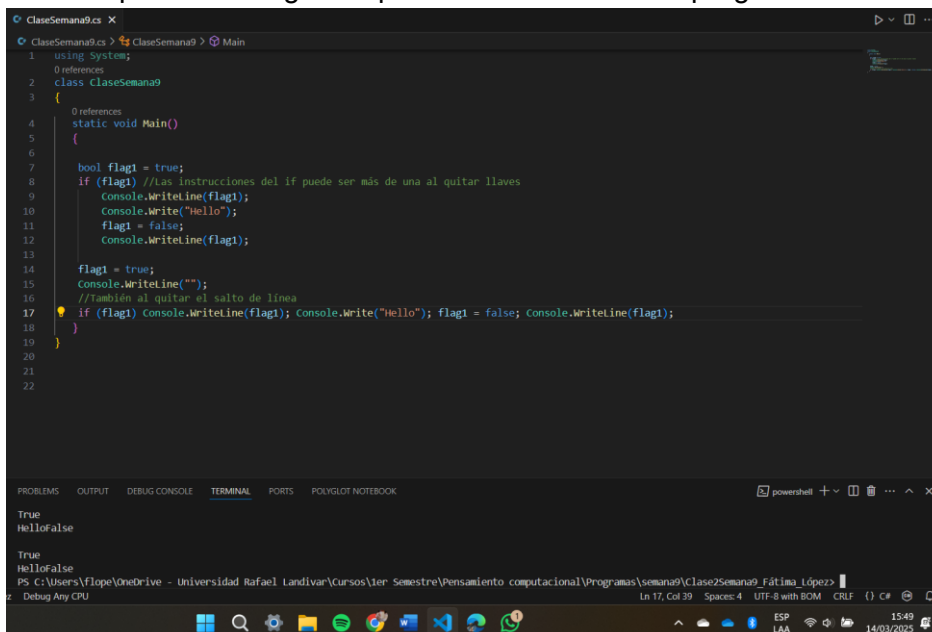
```
1 using System;
2 class ClaseSeman9
3 {
4     static void Main()
5     {
6         //Omitir llaves
7         bool flag1 = true;
8         if (flag1)
9             Console.WriteLine(flag1);
10
11         //Código en una sola línea
12         bool obj2 = true;
13         if (obj2) Console.WriteLine(obj2);
14
15         //Selección else-if omitiendo llaves
16         string name = "steve";
17         if (name == "bob")
18             Console.WriteLine("Found Bob");
19         else if (name == "steve")
20             Console.WriteLine("Found Steve");
21         else
22             Console.WriteLine("Found Chuck");
23     }
24 }
25
26
27
28
```

Terminal output:

```
True
PS C:\Users\Flope\OneDrive - Universidad Rafael Landivar\Cursos\1er Semestre\Pensamiento computacional\Programas\semana9\Clase2Seman9_Fátima López> dotnet run
True
True
Found Steve
PS C:\Users\Flope\OneDrive - Universidad Rafael Landivar\Cursos\1er Semestre\Pensamiento computacional\Programas\semana9\Clase2Seman9_Fátima López>
```

10. Más de una línea de instrucción quitando llaves y saltos de línea

Como observa, el compilador sí lo acepta, pero no quiere decir que siempre sea la opción más legible o práctica al momento de programar.



```
1 using System;
2 class ClaseSeman9
3 {
4     static void Main()
5     {
6         bool flag1 = true;
7         if (flag1) //Las instrucciones del if puede ser más de una al quitar llaves
8             Console.WriteLine(flag1);
9             Console.WriteLine(flag1);
10             Console.WriteLine(flag1);
11             Console.WriteLine(flag1);
12
13         flag1 = true;
14         Console.WriteLine("");
15         //También al quitar el salto de línea
16         if (flag1) Console.WriteLine(flag1); Console.WriteLine("Hello"); flag1 = false; Console.WriteLine(flag1);
17     }
18 }
19
20
21
22
```

Terminal output:

```
True
HelloFalse
True
HelloFalse
PS C:\Users\Flope\OneDrive - Universidad Rafael Landivar\Cursos\1er Semestre\Pensamiento computacional\Programas\semana9\Clase2Seman9_Fátima López>
```