

[Año]

# Depurando

DEPURANDO

JESÚS ALEJANDRO GONZALEZ DIAZ

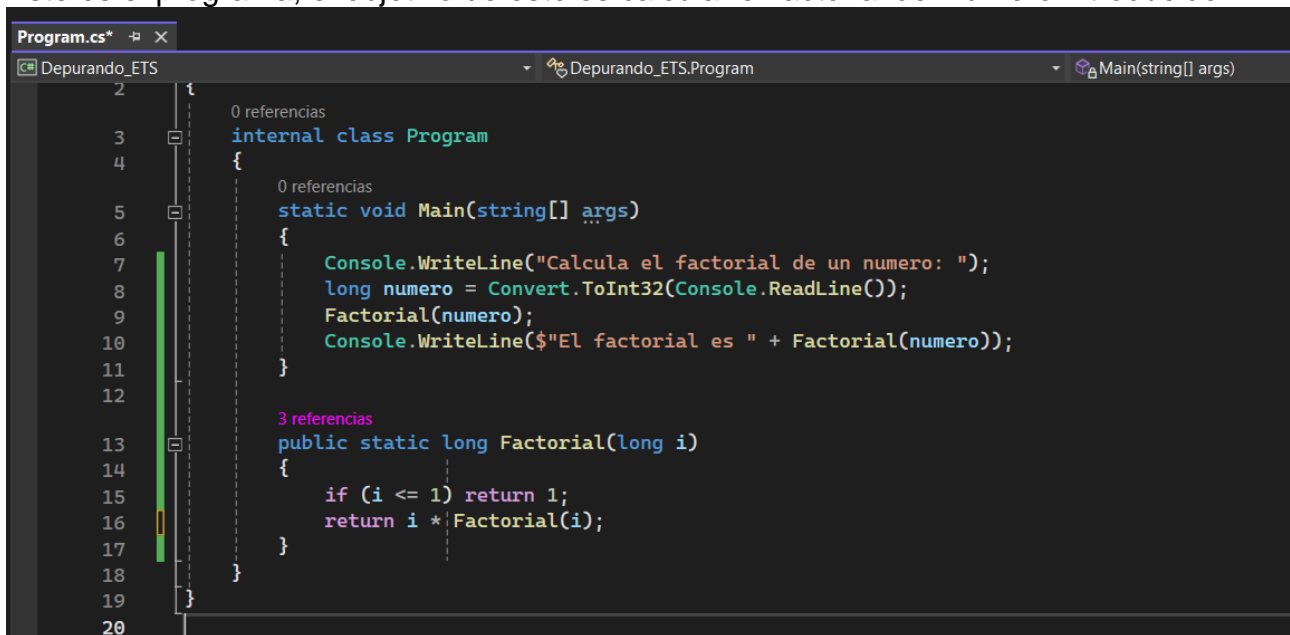
## AUT03\_03 Depuración.

En la tarea del ejercicio tienes el código fuente para calcular el factorial de un número. Hay varios errores que corregir:

Escribe, escribe y escribe en un documento lo que haces y piensas al respecto.

(a) Corrige los errores para que el programa funcione correctamente (puedes ayudarte de los breakpoints). Realiza tu código que lea un número entero y devuelva su factorial.

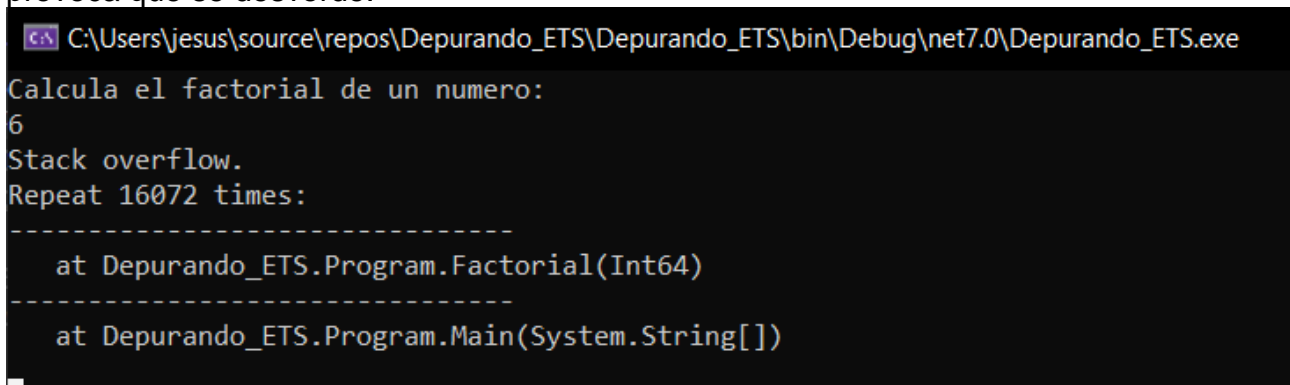
Este es el programa, el objetivo de este es calcular el factorial del numero introducido:



```
Program.cs* [X]
Depurando_ETS
0 referencias
2
3 internal class Program
4 {
5     0 referencias
6     static void Main(string[] args)
7     {
8         Console.WriteLine("Calcula el factorial de un numero: ");
9         long numero = Convert.ToInt32(Console.ReadLine());
10        Factorial(numero);
11        Console.WriteLine($"El factorial es " + Factorial(numero));
12    }
13
14    3 referencias
15    public static long Factorial(long i)
16    {
17        if (i <= 1) return 1;
18        return i * Factorial(i);
19    }
20 }
```

Al introducirle un número muestra esto:

Es un aviso de que el bucle se ha repetido demasiadas veces sumando un valor que provoca que se desborde:



```
C:\Users\jesus\source\repos\Depurando_ETS\Depurando_ETS\bin\Debug\net7.0\Depurando_ETS.exe
Calcula el factorial de un numero:
6
Stack overflow.
Repeat 16072 times:
-----
at Depurando_ETS.Program.Factorial(Int64)
-----
at Depurando_ETS.Program.Main(System.String[])
-----
```

El programa se queda en bucle porque a cada vuelta i siempre vale el numero introducido:

The screenshot shows a C# code editor with the following code:

```
11  
12  
13 3 referencias  
14 static long Factorial(long i)  
15 {  
16     if (i <= 1) return 1;  
17     return i * Factorial(i);  
18 }  
19  
20
```

The variable `i` is highlighted in the call stack, showing its value is 6. The status bar indicates "No se encontraron problemas." (No problems found).

Below the code editor, the "Inspección 1" (Inspection 1) window is open, showing a table with the following data:

Nombre	Valor	Tipo
i	6	long
Agregar elemento a inspección		

Ahora con las variables restando 1 a cada vuelta el valor disminuirá y a cada vuelta realizará su función para así terminar el bucle cuando el valor sea 1:

The screenshot shows the same C# code editor, but the recursive call is now `return i-- * Factorial(i--);`. The variable `i` is highlighted in the call stack, showing its value is 3. The status bar indicates "No se encontraron problemas." (No problems found).

Below the code editor, the "Inspección 1" (Inspection 1) window is open, showing a table with the following data:

Nombre	Valor	Tipo
i	3	long
Agregar elemento a inspección		

```
Consola de depuración de Microsoft Visual Studio
Calcula el factorial de un numero:
6
El factorial es 720

C:\Users\jesus\source\repos\Depurando_ETTS\Depurando_ETTS\bin\De
el código 0.
Para cerrar automáticamente la consola cuando se detiene
Cerrar la consola automáticamente al detenerse la depurac
Presione cualquier tecla para cerrar esta ventana. . .
```

(b) ¿A partir de qué número se producirá un desbordamiento al calcular el factorial? Coloca un breakpoint que se activará si el n.º introducido provoca un desbordamiento.

El factorial de 21 produce un desbordamiento en el programa:

```
Consola de depuración de Microsoft Visual Studio
Calcula el factorial de un numero:
21
El factorial es -4249290049419214848

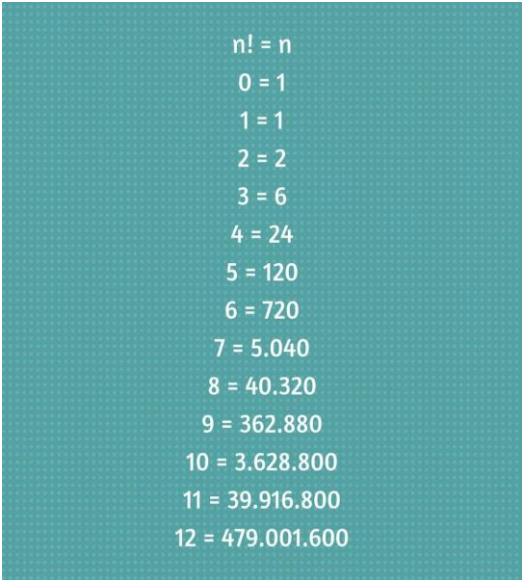
C:\Users\jesus\source\repos\Depurando_ETTS\Depurando_ETTS\bin\De
el código 0.
Para cerrar automáticamente la consola cuando se detiene la de
Cerrar la consola automáticamente al detenerse la depuración.
Presione cualquier tecla para cerrar esta ventana. . .
```

(c) Si utilizaras como tipo de dato “**byte**” para recoger el n.º a calcular el factorial, ¿en qué cifra se produciría el desbordamiento? ¿y en el caso de una variable **ulong**?

(d) ¿Qué se te ocurre que podrías hacer para evitar el desbordamiento? Aplícalo al código y haz pruebas para verificar que funciona.

Código con el que trabajar.

```
long Factorial(long i)
{
    if (i <= 1) return 1;
    return i * Factorial(i);
}
```



A teal rectangular box with a fine grid pattern, containing a list of factorial values. The text is white and aligned to the left.

$n!$	$=$	$n$
0	=	1
1	=	1
2	=	2
3	=	6
4	=	24
5	=	120
6	=	720
7	=	5.040
8	=	40.320
9	=	362.880
10	=	3.628.800
11	=	39.916.800
12	=	479.001.600