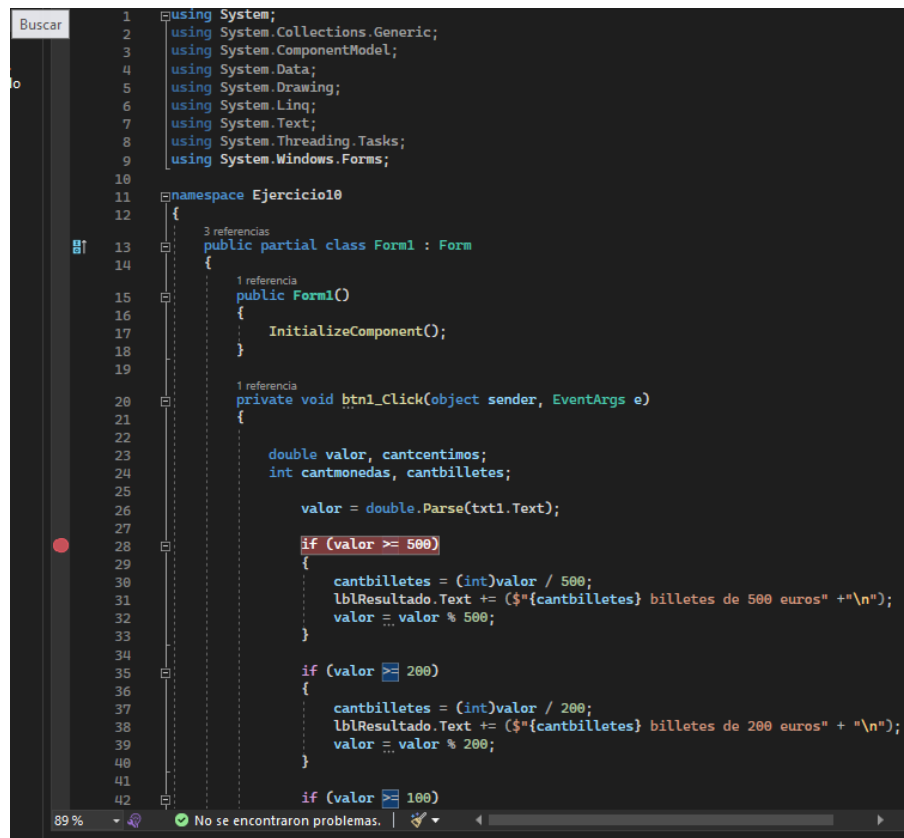


## 5. Prácticas

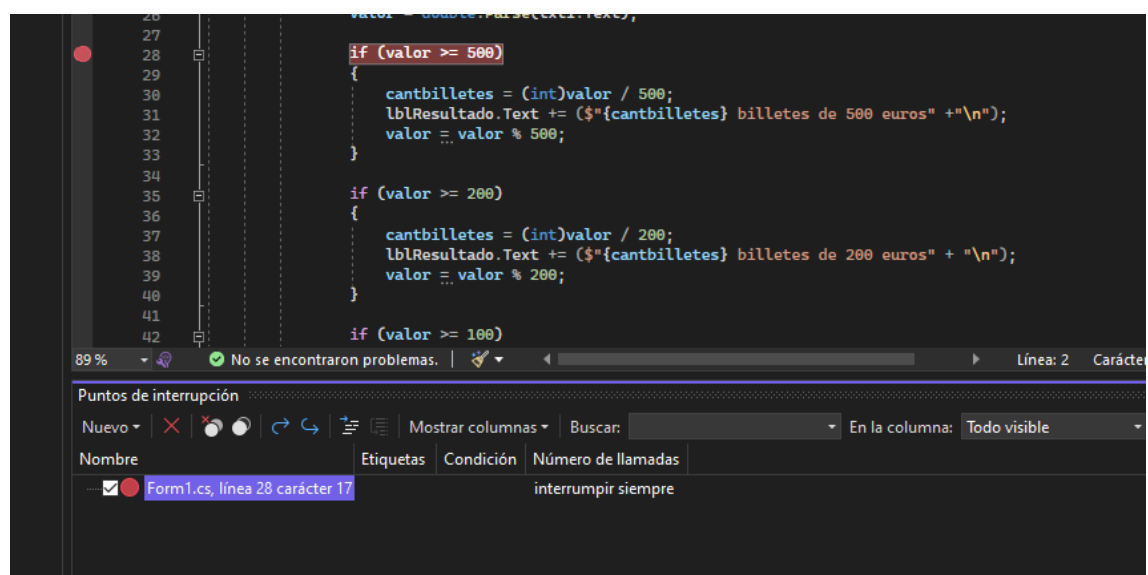
### Ejercicio 1

El primer ejercicio consistirá en depurar el ejercicio 10 del tema de programación (usando la solución que os proporcionan en el módulo). Si no disponéis de ella, podéis descargarla del Moodle. Debéis realizar lo siguiente:

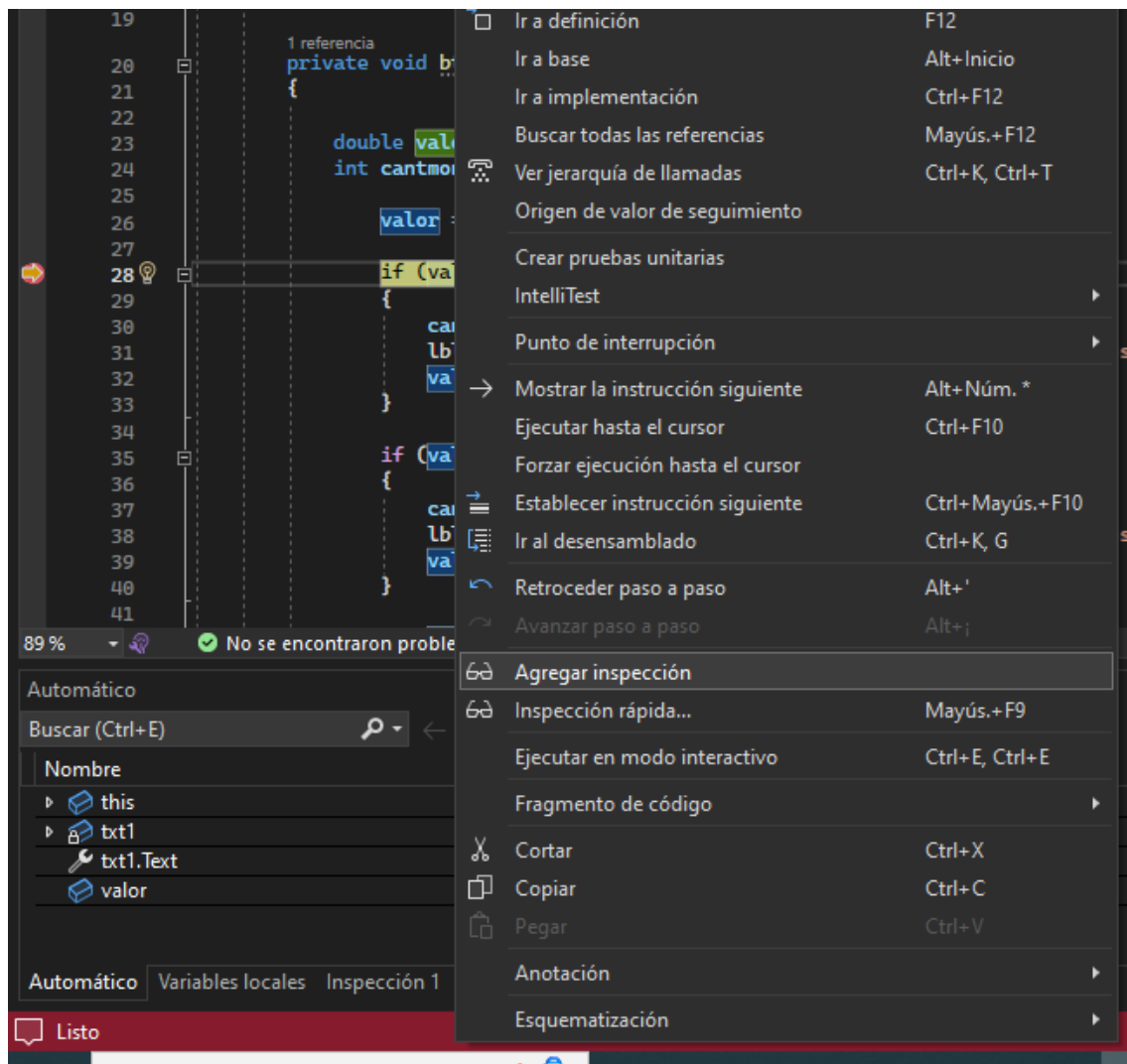
- Colocar un punto de interrupción para detener la ejecución al inicio del cálculo. ¿Dónde lo colocas?

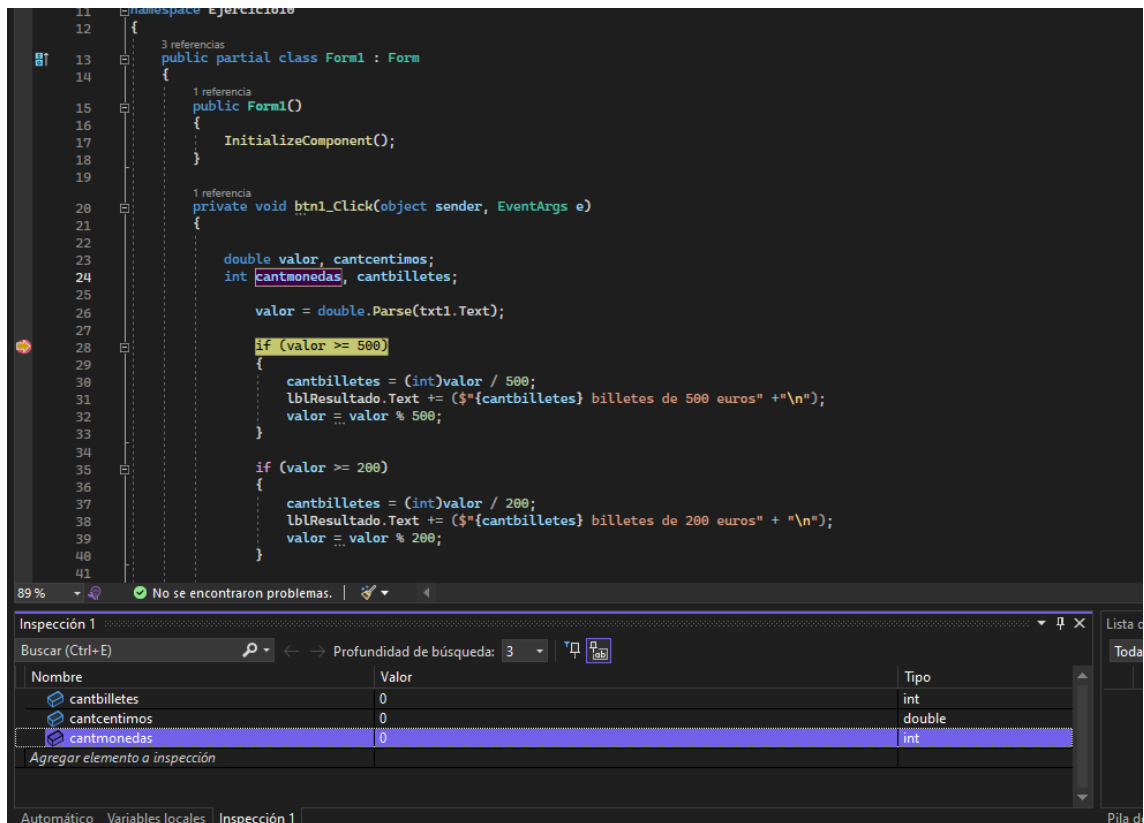


```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace Ejercicio10
12 {
13     3 referencias
14     public partial class Form1 : Form
15     {
16         1 referencia
17         public Form1()
18         {
19             InitializeComponent();
20         }
21
22         1 referencia
23         private void btn1_Click(object sender, EventArgs e)
24         {
25             double valor, cantcentimos;
26             int cantmonedas, cantbilletes;
27
28             valor = double.Parse(txt1.Text);
29
30             if (valor >= 500)
31             {
32                 cantbilletes = (int)valor / 500;
33                 lblResultado.Text += ("{"cantbilletes} billetes de 500 euros" + "\n");
34                 valor = valor % 500;
35             }
36
37             if (valor >= 200)
38             {
39                 cantbilletes = (int)valor / 200;
40                 lblResultado.Text += ("{"cantbilletes} billetes de 200 euros" + "\n");
41                 valor = valor % 200;
42             }
43
44             if (valor >= 100)
45             {
46                 cantbilletes = (int)valor / 100;
47                 lblResultado.Text += ("{"cantbilletes} billetes de 100 euros" + "\n");
48                 valor = valor % 100;
49             }
50
51             cantcentimos = (int)valor;
52             lblResultado.Text += ("{"cantcentimos} centimos" + "\n");
53             valor = valor % 100;
54
55             cantmonedas = (int)valor;
56             lblResultado.Text += ("{"cantmonedas} monedas de 1 euro" + "\n");
57             valor = valor % 1;
58
59             lblResultado.Text += "\nTotal: " + valor.ToString("F2") + " euros";
60         }
61     }
62 }
```

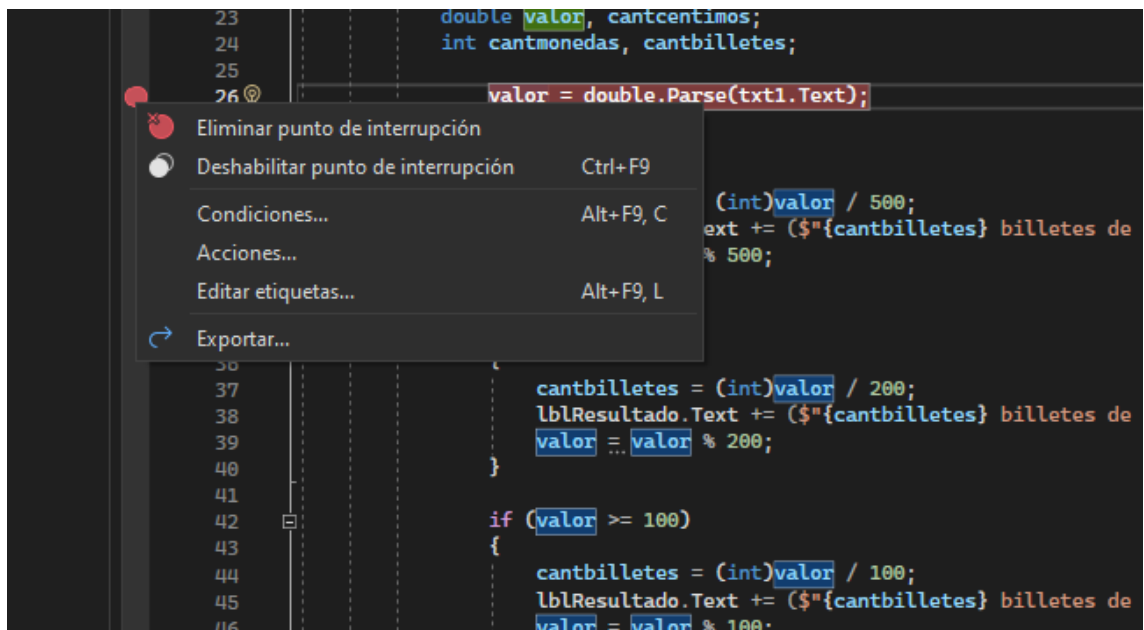


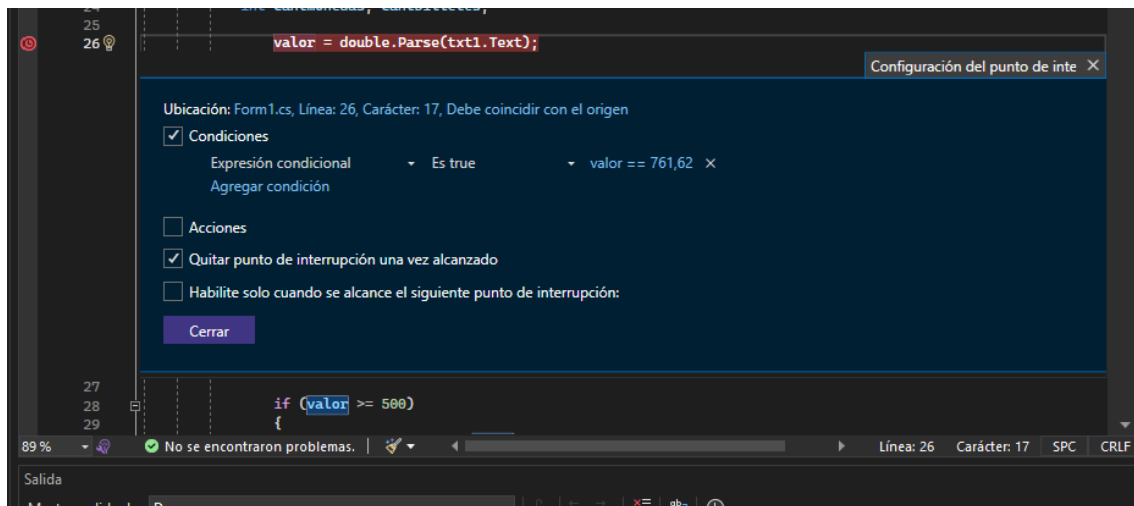
- Agregar a la ventana de inspección las variables cantidad, billetes, monedas, céntimos y lResultado.Text.





- Depurar el código con la cantidad 761,62. Adjuntar una captura por cada vez que se produzca un cambio en alguna de las variables. Indicar también en qué condicionales no entra el programa.





## Ejercicio 2

Se ha desarrollado un módulo que permite calcular el precio de un telegrama en base a los siguientes requisitos:

- Los telegramas pueden ser ordinarios o urgentes.

☐ urgente?

- El coste de un telegrama ordinario son 2,5 euros, hasta un máximo de 10 palabras. A partir de ahí, cada palabra adicional tiene un coste de 0,50€.

```
public Form1()
{
    InitializeComponent();
}

1 referencia
private void btnCalcular_Click(object sender, EventArgs e)
{
    string textoTelegrama;
    char tipoTelegrama = ' ';
    int numPalabras = 0;
    double coste;

    //Leo el telegrama
    textoTelegrama = txtTelegrama.Text;

    // telegrama urgente?
    if (cbUrgente.Checked)
        tipoTelegrama = 'u';

    //Obtengo el número de palabras que forma el telegrama
    numPalabras = textoTelegrama.Length;

    //Si el telegrama es ordinario
    if (tipoTelegrama == 'o')
    {
        if (numPalabras <= 10)
            coste = 25;
        else
            coste = 0.5 * numPalabras;
    }
    else
    {
        //Si el telegrama es urgente
        if (tipoTelegrama == 'u')
        {
            if (numPalabras <= 10)
                coste = 25;
            else
                coste = 0.5 * numPalabras;
        }
    }
}
```

- El coste de un telegrama urgente son 5 euros, hasta un máximo de 10 palabras. A partir de ahí, cada palabra adicional tiene un coste de 0,75€.

```

    else
        coste = 0.5 * numPalabras;
    else
        //Si el telegrama es urgente
        if (tipoTelegrama == 'u')
            if (numPalabras <= 10)
                coste = 5;
            else
                coste = 5 + 0.75 * (numPalabras - 10);
        else
            coste = 0;
        txtPrecio.Text = coste.ToString() + " euros";
    }
}

```

A. Crea un aplicación que muestre el siguiente formulario y ejecute el código adjunto cuando se pulse el botón 'Coste'. Para que el código funcione correctamente, debe tenerse en cuenta lo siguiente:

- (1) El checkbox debe llamarse cbUrgente

Diseño	
(Name)	cbUrgente
Anchor	Top, Left
AutoSize	True
Dock	None

- (2) El textbox para el coste se llamará txtPrecio

Tag	
Diseño	
(Name)	txtPrecio
Anchor	Top, Left
Dock	None
GenerateMember	True
Location	348; 312

- (3) El textbox para el telegrama se llamará txtTelegrama y será multilínea

ImeMode	NoControl
MaxLength	32767
Multiline	True
PasswordChar	
ReadOnly	False

B. Introduce el siguiente telegrama ordinario: “Este es un telegrama de prueba”. Añadid al texto vuestras iniciales y el curso actual. Su coste debería ser de 2,5€.

```
if (numPalabras <= 10)
    coste = 25 / 10;
else
    coste = 0.5 * numPalabras;
```

Esto es un telegrama de prueba L 1DAW N

☐ urgente?

Valor

Calcular

C. Ejecuta el código hasta que se obtenga el tipo de telegrama (ordinario o urgente). Cuando situemos la flecha en dicha instrucción, deberemos pulsar F11, para poder ejecutarla. Inspecciona su valor. ¿El valor obtenido coincide con lo esperado? Si no es así, corrige el código y vuelve a comenzar.

D. Ejecuta la siguiente línea para obtener el número de palabras. ¿Tiene el valor esperado? Si no es así, corrige el código y vuelve a comenzar.

E. Al tratarse de un telegrama ordinario, el depurador debería entrar en la siguiente rama: `if (tipoTelegrama == 'o' ) if (numPalabras <= 10)` ¿es así?

F. Ejecuta la siguiente instrucción y calcula el coste del telegrama. ¿Es el valor esperado? Si no es así, corrige los posibles errores.

G. Vuelve a ejecutar el programa con los siguientes textos

```
1 referencia
private void btnCalcular_Click(object sender, EventArgs e)
{
    string textoTelegrama;
    char tipoTelegrama = 'o';
    int numPalabras = 0;
    double coste;

    //Leo el telegrama
    textoTelegrama = txtTelegrama.Text;
    string[] words = textoTelegrama.Split(' ');

    // telegrama urgente?
    if (cbUrgente.Checked)
    {
        tipoTelegrama = 'u';

        //Obtengo el número de palabras que forma el telegrama
        numPalabras = words.Length;

        //Si el telegrama es ordinario
        if (tipoTelegrama == 'o')
        {
            if (numPalabras <= 10)
            {
                coste = 2.5;
            }
            else
            {
                coste = 0.5 * numPalabras;
            }
        }
        else
        {
            //Si el telegrama es urgente
        }
    }
}
```

- “Este es un telegrama de prueba creado para la práctica de Entornos de desarrollo” (ordinario).

The screenshot shows a Windows application window titled 'Form1'. It contains a text box with the text 'Este es un telegrama de prueba creado para la practica de Entornos de Desarrollo'. Below the text box is a checkbox labeled 'urgente?' which is unchecked. To the right of the checkbox is a text box labeled 'Valor' containing the text '7 euros'. Below the 'Valor' text box is a button labeled 'Calcular'.

- “Este es un telegrama de prueba” (urgente).

The screenshot shows the same Windows application window titled 'Form1'. The text box now contains the text 'Este es un telegrama de prueba'. The checkbox labeled 'urgente?' is now checked. The text box labeled 'Valor' now contains the text '5 euros'. The button labeled 'Calcular' remains below the 'Valor' text box.

- “Este es un telegrama de prueba creado para la práctica de Entornos de desarrollo” (urgente). Para este caso, situa un breakpoint condicional (condición: numPalabras > 10) en la siguiente línea: `coste = 5 + 0.75 * (numPalabras - 10);` ¿Se detiene la ejecución? Añade a cada uno de ellos tus iniciales y el curso actual. Comprueba si existen más errores en el código.

```

31 textoTelegrama = txtTelegrama.Text;
32 string[] words = textoTelegrama.Split(' ');
33
34 // telegrama urgente?
35
36 if (cbUrgente.Checked)
37
38     tipoTelegrama = 'u';
39
40 //Obtengo el número de palabras que forma el telegrama
41 numPalabras = words.Length;
42
43 //Si el telegrama es ordinario
44 if (tipoTelegrama == 'o')
45 {
46     if (numPalabras <= 10)
47
48         coste = 2.5;
49
50     else

```

Form1

Esto es un telegrama de prueba L 1DAW N

Esto es un telegrama de prueba creado para la practica de Entornos de desarrollo

☒ urgente?

Valor

```

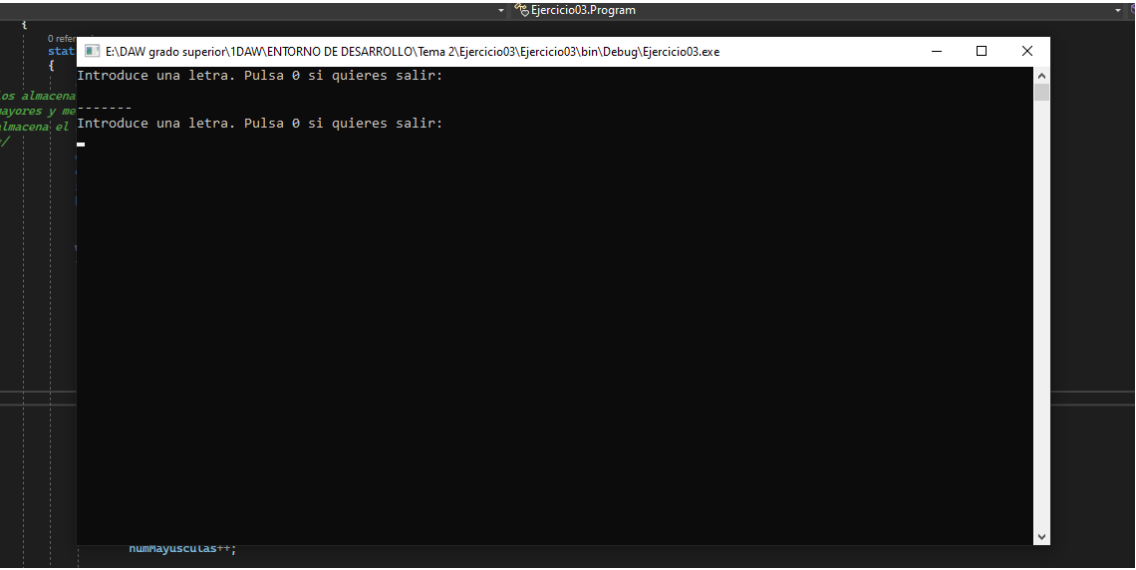
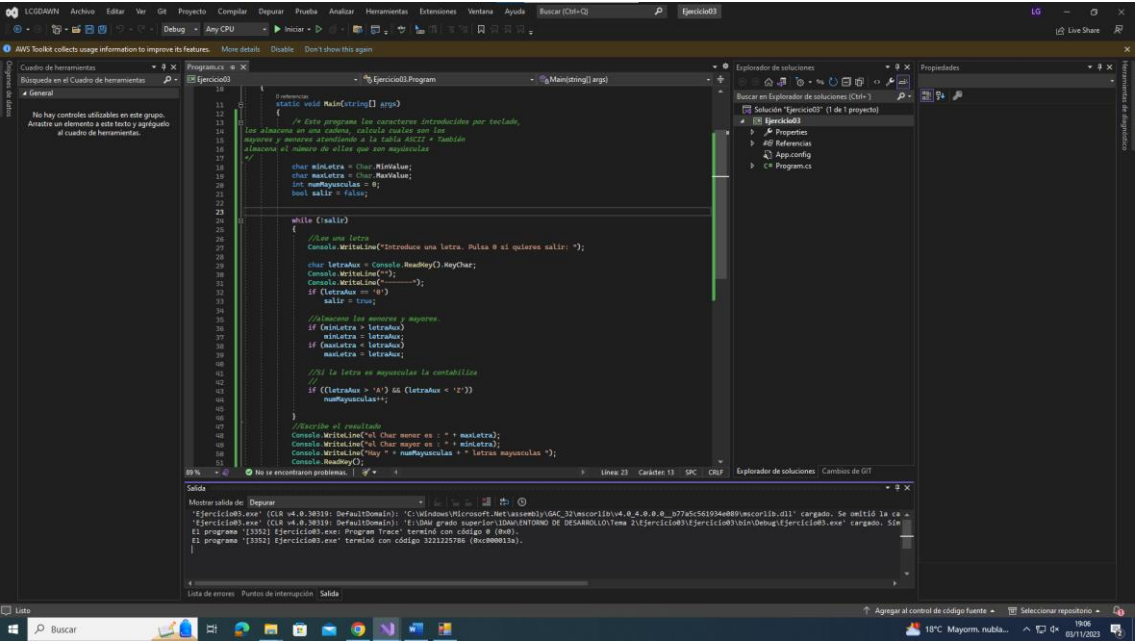
62     if (numPalabras <= 10)
63         coste = 5;
64
65     else
66         coste = 5 + 0.75 * (numPalabras - 10);
67
68     else
69         coste = 0;
70     txtPrecio.Text = coste.ToString() + " euros";
71
72
73
74
75

```

SI SE DETIENE EN EL BREAKPOINT



### Ejercicio 3



Documenta el uso de las ventanas de locales, automático e inspección para la variable Letras. Usa dos puntos de interrupción para detener el código cuando se ha encontrado un carácter máximo y uno mínimo. ¿Es correcto el funcionamiento?

```
/* Este programa lee caracteres introducidos por teclado,
los almacena en una cadena, calcula cuales son los
mayores y menores atendiendo a la tabla ASCII * También
almacena el número de ellos que son mayúsculas
*/

char minLetra = Char.MinValue;
char maxLetra = Char.MaxValue;
int numMayusculas = 0;
bool salir = false;

while (!salir)
{
    //Leo una letra
    Console.WriteLine("Introduce una letra. Pulsa 0 si quieres salir: ");

    char letraAux = Console.ReadKey().KeyChar;
    Console.WriteLine("");
    Console.WriteLine("-----");
    if (letraAux == '0')
        salir = true;
    else
    {
        //almaceno los menores y mayores.
        if (minLetra < letraAux)
            minLetra = letraAux;
        if (maxLetra > letraAux)
            maxLetra = letraAux;

        //Si la letra es mayusculas la contabiliza
        //
        if ((letraAux >= 'A') && (letraAux <= 'Z'))
            numMayusculas++;
    }
}

//Escribe el resultado
Console.WriteLine("el Char menor es : " + maxLetra);
Console.WriteLine("el Char mayor es : " + minLetra);
Console.WriteLine("Hay " + numMayusculas + " letras mayusculas ");
Console.ReadKey();
}
```

LCGDawn Archivo Editar Ver Git Proyecto Compilar Depurar Prueba Analizar Herramientas Extensiones Ventana Ayuda Buscar (Ctrl+Q)

Proceso: [12220] Ejercicio03.exe Subproceso: [1784] Subproceso principal Marco de pila: Ejercicio03.Program.Main

AWS Toolkit collects usage information to improve its features. More details Disable Don't show this again

Program.cs Ejercicio03

```
13  /* Este programa lee caracteres introducidos por teclado,
14  los almacena en una cadena, calcula cuales son los
15  mayores y menores atendiendo a la tabla ASCII * También
16  almacena el número de ellos que son mayúsculas
17  */
18  char minLetra = Char.MinValue;
19  char maxLetra = Char.MaxValue;
20  int numMayusculas = 0;
21  bool salir = false;
22
23
24  while (!salir)
25  {
26      //Leo una letra
27      Console.WriteLine("Introduce una letra. Pulsa 0 si quieres salir: ");
28
29      char letraAux = Console.ReadKey().KeyChar;
30      Console.WriteLine("");
31      Console.WriteLine("-----");
32      if (letraAux == '0')
33          salir = true;
34
35      //almaceno los menores y mayores.
36      if (minLetra > letraAux)
37          minLetra = letraAux;
38      if (maxLetra < letraAux)
39          maxLetra = letraAux;
40
41      //Si la letra es mayusculas la contabiliza
42      //
43      if ((letraAux > 'A') && (letraAux < 'Z'))
44          numMayusculas++;
45
46      //Escribe el resultado
47      Console.WriteLine("el Char menor es : " + maxLetra);
48      Console.WriteLine("el Char mayor es : " + minLetra);
49      Console.WriteLine("Hay " + numMayusculas + " letras mayusculas ");
50      Console.ReadKey();
51  }
52
53
54
```

89 % No se encontraron problemas.

Automático Buscar (Ctrl+E) Profundidad de búsqueda: 3

Nombre	Valor	Tipo
Char.MinValue	0 '\0'	char
args	{string[0]}	string[]
minLetra	0 '\0'	char

Salida  
Mostrar salida de: Depu  
'Ejercicio03.exe'  
'Ejercicio03.exe'  
'Ejercicio03.exe'

Automático Variables locales Inspección 1

Lista

89 % No se encontraron problemas.

Automático Buscar (Ctrl+E) Profundidad de búsqueda: 3

Nombre	Valor	Tipo
Char.MinValue	0 '\0'	char
args	{string[0]}	string[]
minLetra	0 '\0'	char

Automático Variables locales Inspección 1

Lista

Variables locales Buscar (Ctrl+E) Profundidad de búsqueda: 3

Nombre	Valor	Tipo
args	{string[0]}	string[]
minLetra	0 '\0'	char
maxLetra	0 '\0'	char
numMayusculas	0	int
salir	false	bool

Automático Variables locales Inspección 1

Situa un breakpoint condicional (condición: letraAux == '0') en la siguiente línea: salir = true; Cuando se detenga la ejecución, continúa ejecutando paso a paso. ¿es este comportamiento normal? Corrige todos los posibles errores.

