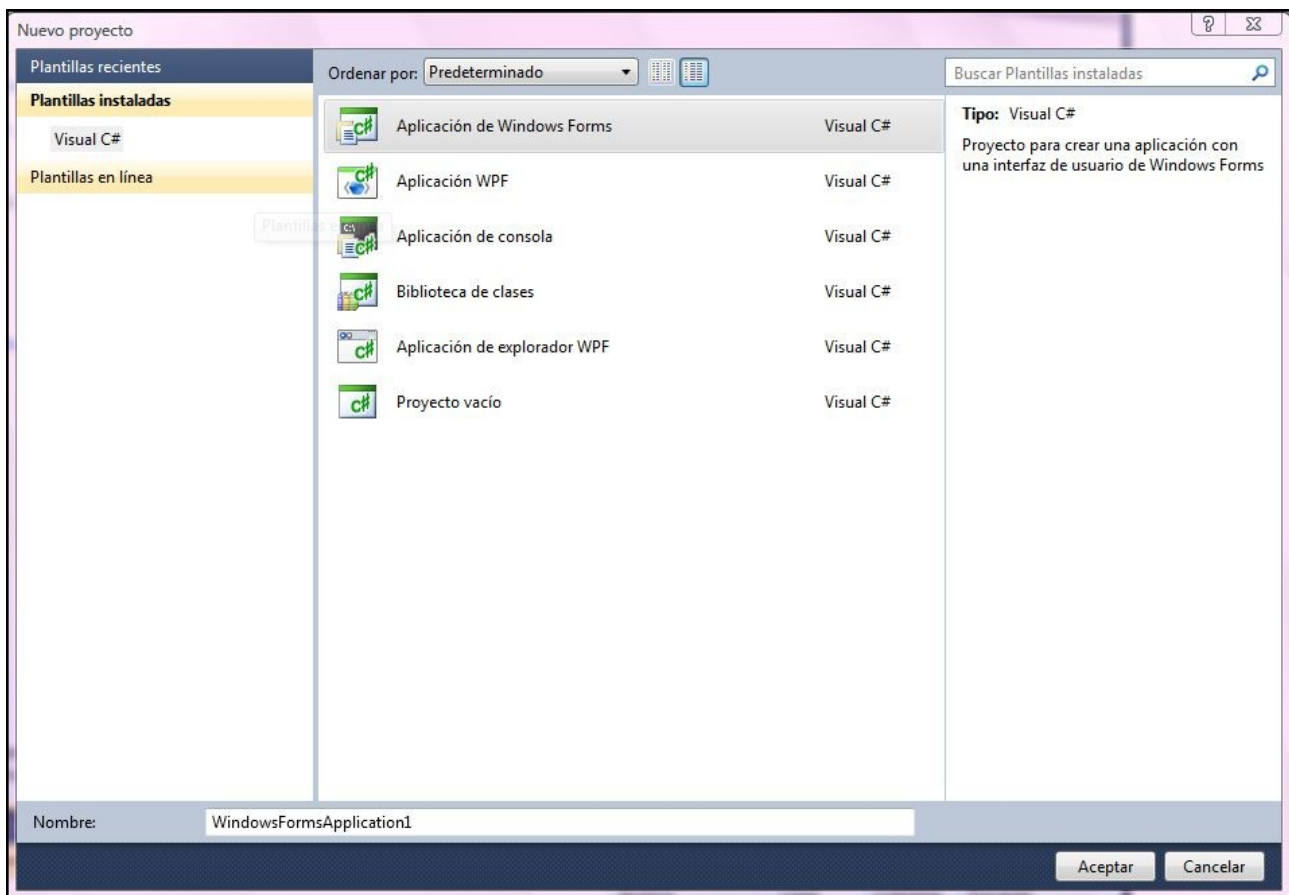


Procesar datos con C#

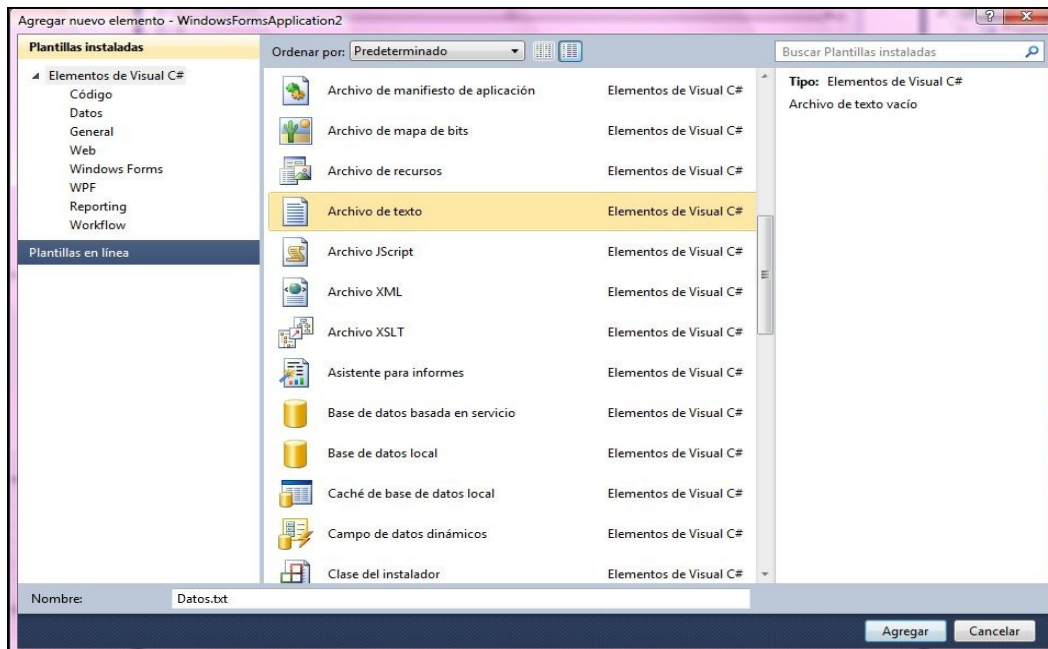
Procesar datos con C#

Se trata de crear una aplicación para poder abrir un archivo de texto, obtener su contenido y procesar la información que hay en él para poder utilizarla.

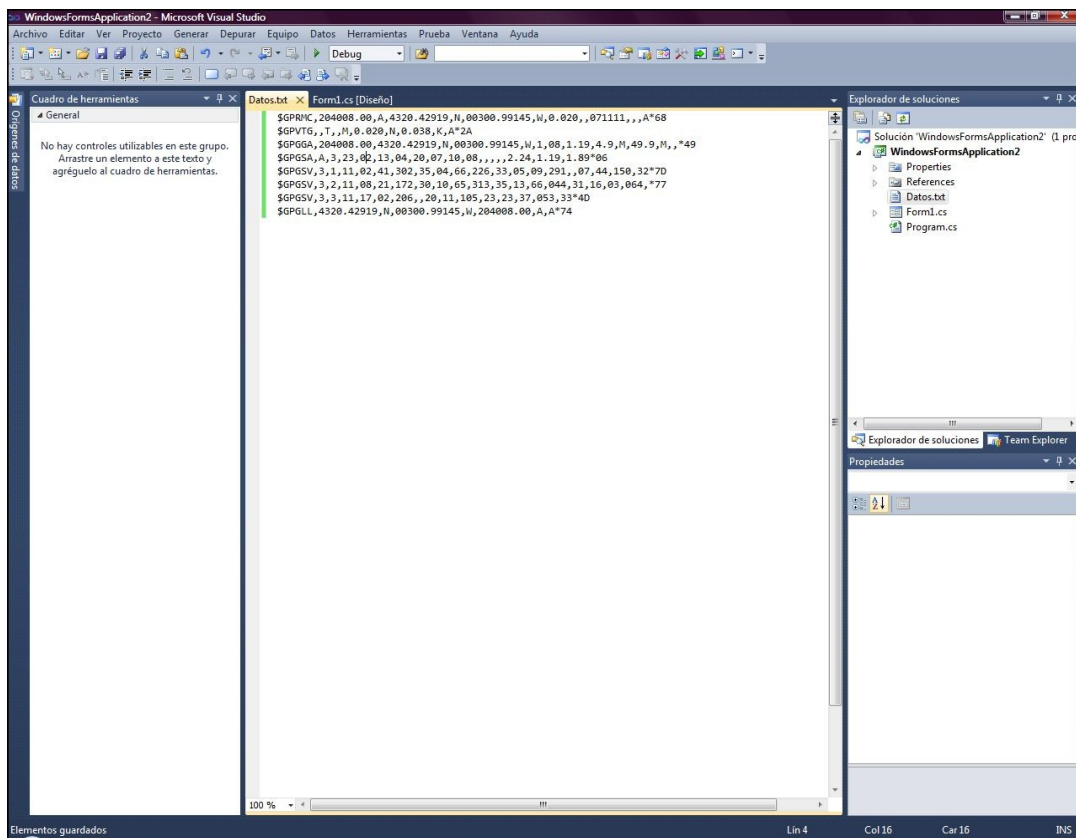
Abrimos un nuevo proyecto de [Microsoft Visual C# 2010](#).



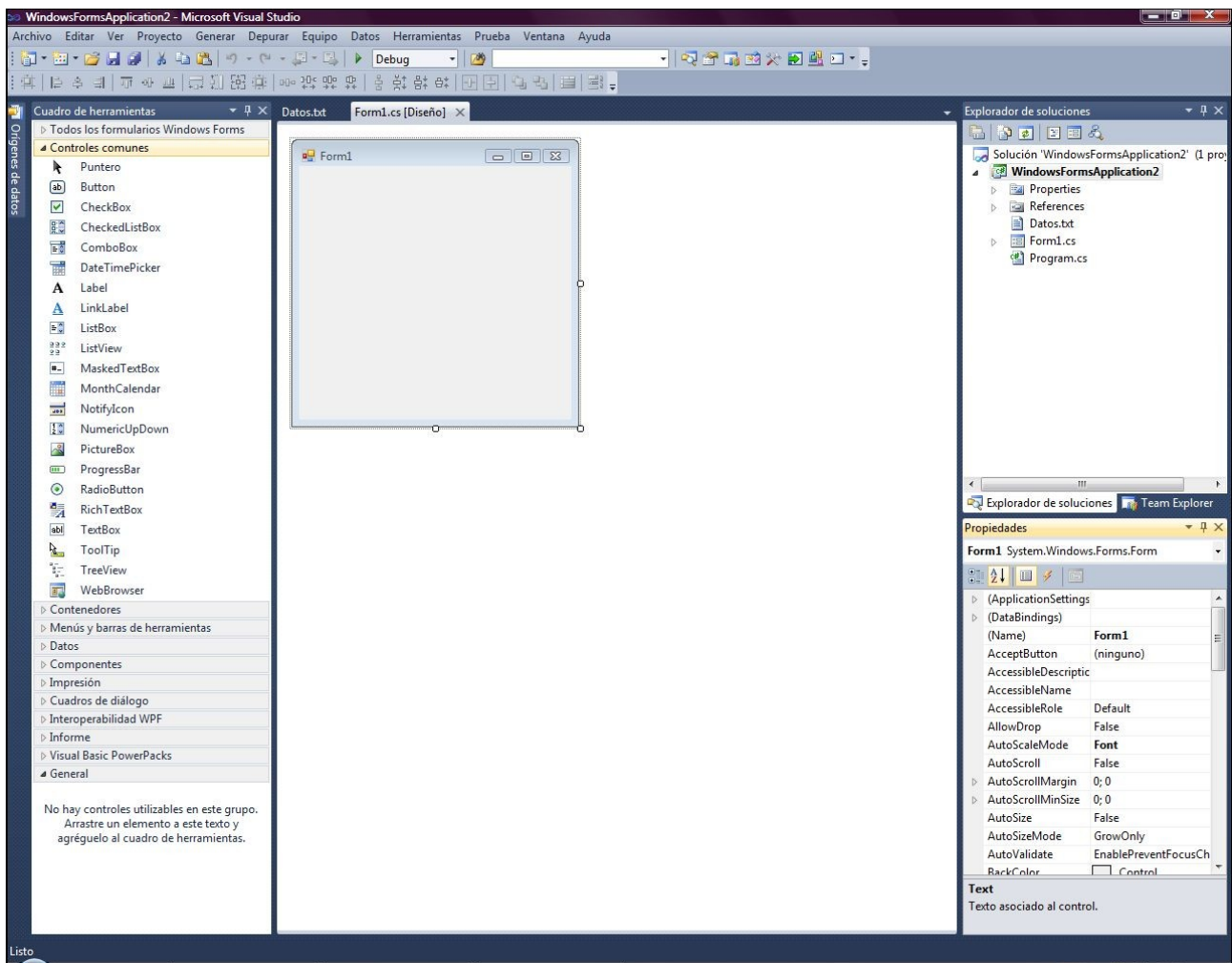
Para poder ver lo que hacemos con este ejemplo, previamente, crearemos un fichero de texto donde guardaremos la información que aparece al final de este documento como **Texto ejemplo**. Para ello, en la parte superior derecha de la pantalla principal, pinchamos con el botón secundario en el nombre de nuestro proyecto (dentro del **Explorador de soluciones**) y elegimos **Agregar, Nuevo elemento**. Seleccionamos **Archivo de texto** y lo guardamos con el nombre *Datos.txt*.



Seguidamente se nos abrirá una ventana vacía en la que copiaremos el texto:



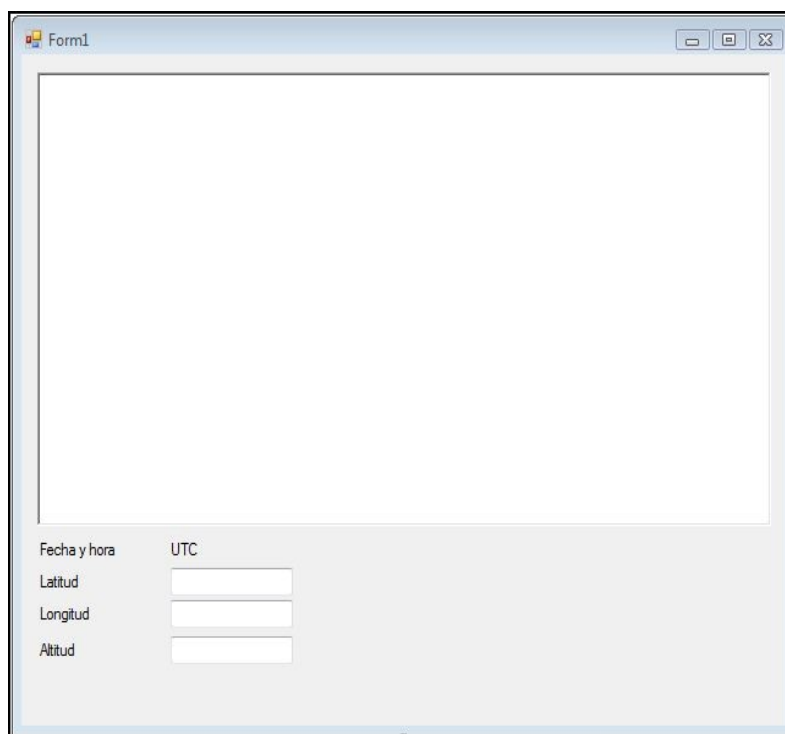
Volvemos a la pestaña de *Diseño del formulario*. Para realizar esta aplicación debemos añadir al formulario un **RichTextBox** (cuadro de texto enriquecido), 5 **Label** (etiqueta) y 3 **TextBox** (cuadro de texto) desde el **Cuadro de Herramientas**, que se encuentra en el lateral izquierdo de la pantalla.



Cambiaremos las siguientes **Propiedades** (cuadro inferior derecho) de los objetos añadidos:

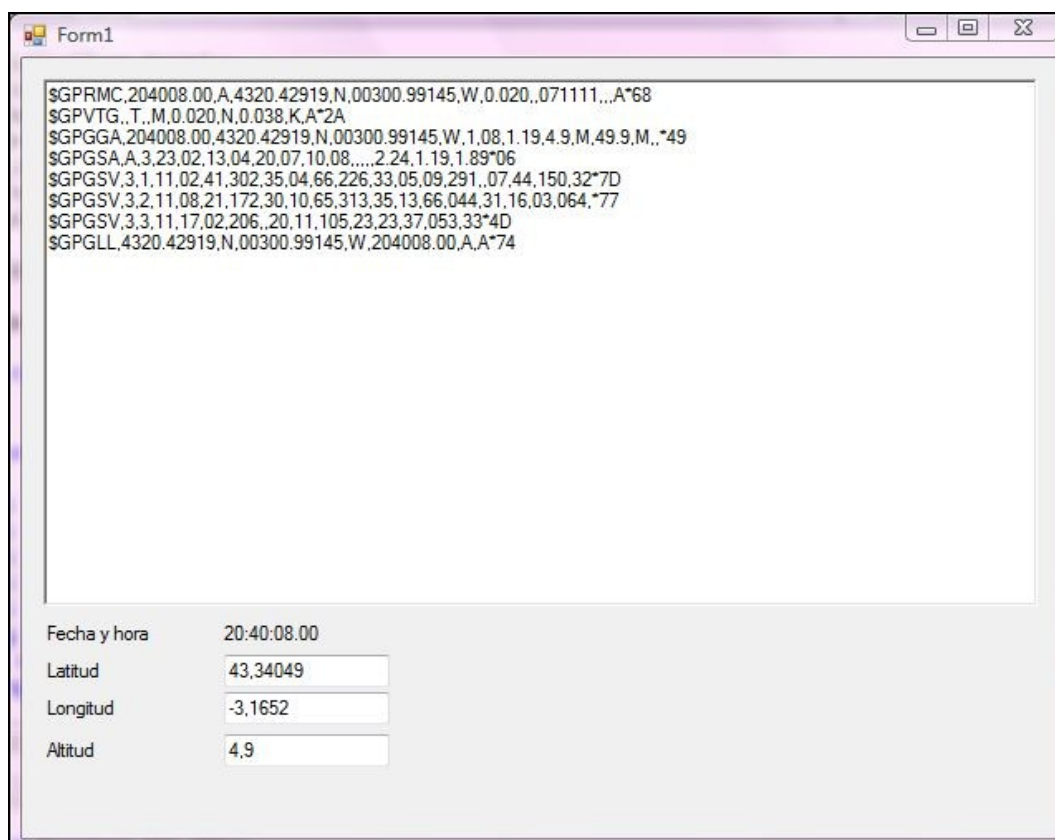
- **(Name)**: asignaremos el nombre *rtbDatos* para el cuadro **de texto enriquecido**, los nombres *lbFecha*, *lbUTC*, *lbLat*, *lbLong* y *lbAlt* para las **etiquetas** y *tbLat*, *tbLong* y *tbAlt* para los **cuadros de texto**.
- **Text**: asignaremos los textos *Fecha y hora*, *UTC*, *Latitud*, *Longitud* y *Altitud* para los nombres *lbFecha*, *lbUTC*, *lbLat*, *lbLong* y *lbAlt* respectivamente.

De esta forma, tendremos un formulario como este:



The screenshot shows a Windows application window titled "Form1". It contains a large, empty text box at the top. Below the text box, there are four labels and their corresponding input fields: "Fecha y hora" (Date and time), "UTC", "Latitud" (Latitude), "Longitud" (Longitude), and "Altitud" (Altitude). The input fields are currently empty.

Para que este ejemplo dé el resultado que se ve a continuación, debemos ir a [Ver](#), pinchar en **Código** y escribir el **Código** (en “Ubicación_del_archivo” debemos indicar la *Ruta de acceso* al archivo deseado).



The screenshot shows the same "Form1" window, but now it displays a list of NMEA sentences in the text area. The sentences are:

```
$GPRMC,204008.00,A,4320.42919,N,00300.99145,W,0.020,071111,...A*68
$GPVTG,.T,M,0.020,N,0.038,K,A*2A
$GPGGA,204008.00,4320.42919,N,00300.99145,W,1.08,1.19,4.9,M,49.9,M,.,*49
$GPGSA,A,3,23,02,13,04,20,07,10,08,....,2.24,1.19,1.89*06
$GPGSV,3,1,11,02,41,302,35,04,66,226,33,05,09,291,07,44,150,32*7D
$GPGSV,3,2,11,08,21,172,30,10,65,313,35,13,66,044,31,16,03,064,*77
$GPGSV,3,3,11,17,02,206,20,11,105,23,23,37,053,33*4D
$GPGLL,4320.42919,N,00300.99145,W,204008.00,A,A*74
```

Below the text area, the input fields are now populated with the following values:

Label	Value
Fecha y hora	20:40:08.00
Latitud	43,34049
Longitud	-3,1652
Altitud	4,9

Comentarios sobre el código

- Se escribe la línea `using System.IO` para poder acceder a los datos mediante la función `StreamReader`. También existe una función para escribir en un fichero (`StreamWriter`).
- Dependiendo la configuración del ordenador en el que estemos trabajando, los decimales se indicaran mediante un punto o una coma. Los datos a los que accedemos vienen separados por puntos, por lo que se crea una función `PuntoPorComa(string)` que los sustituye.
- `Trim()` se utiliza para quitar los espacios en blanco que puede haber en el texto.
- `Split('\n')` se utiliza para dividir un texto en tantas partes como veces aparezca el carácter de separación que hayamos introducido.
- `try catch` se utiliza para intentar realizar las operaciones que indiquemos dentro del “try” y, si esto no es posible, realizar otras operaciones distintas que agruparemos en el “catch”. También existe la palabra `finally` para, bien se haya pasado por el grupo de código primero o segundo, acabar ejecutando las sentencias incluidas dentro del “finally”.

Texto ejemplo

```
$GPRMC,204008.00,A,4320.42919,N,00300.99145,W,0.020,,071111,,,A*68
$GPVTG,,T,M,0.020,N,0.038,K,A*2A
$GPGGA,204008.00,4320.42919,N,00300.99145,W,1.08,1.19,4.9,M,49.9,M,,*49
$GPGSA,A,3,23,02,13,04,20,07,10,08,,,,,2.24,1.19,1.89*06
$GPGSV,3,1,11,02,41,302,35,04,66,226,33,05,09,291,,07,44,150,32*7D
$GPGSV,3,2,11,08,21,172,30,10,65,313,35,13,66,044,31,16,03,064,*77
$GPGSV,3,3,11,17,02,206,,20,11,105,23,23,37,053,33*4D
$GPGLL,4320.42919,N,00300.99145,W,204008.00,A,A*74
```

Código

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

using System.IO; //Para poder leer los datos

namespace WindowsFormsApplication2
{
    public partial class Form1 : Form
    {
```

```
#region Variables globales

public string stLatitud;
public string stLongitud;
public string stAltitud;
public double doLatitud;
public double doLongitud;
string Datos;
const string Ruta = @"C:\Ubicación_del_archivo\Datos.txt";
#endregion

#region Funciones
#region Formato hora
string FormatoHora(string Cadena)
{
    Cadena = Cadena.Insert(4, ":");
    Cadena = Cadena.Insert(2, ":");

    return Cadena;
}
#endregion

#region Signo de las coordenadas
string SignoCoor(string Cadena1, string Cadena2)
{
    double doCadena1;

    doCadena1 = Convert.ToDouble(Cadena1);

    switch (Cadena2)
    {
        case "S":
            doCadena1 = (-1) * doCadena1;
            Cadena1 = Convert.ToString(doCadena1);
            break;
        case "W":
            doCadena1 = (-1) * doCadena1;
            Cadena1 = Convert.ToString(doCadena1);
            break;
        default:
            Cadena1 = Convert.ToString(doCadena1);
            break;
    }

    return Cadena1;
}
#endregion

#region Cambiar un punto por una coma
string PuntoPorComa(string Cadena)
{
    Cadena = Cadena.Replace('.', ',');

    return Cadena;
}
#endregion
#endregion

#region Obtención de los datos
string Obtencion_datos()
{

```

```

        StreamReader sr = new StreamReader(Ruta);
        string Actual = sr.ReadToEnd();
        sr.Close();

        return Actual;
    }
}
#endregion

#region Procesar los datos
private void Procesar_datos(string Datos)
{
    string stLat;
    string stLong;
    string stAlt;
    string stUTC;

    rtbDatos.Text = Datos;
    Datos = Datos.Trim();

    string[] stArray = Datos.Split('\n');

    for (int i = 0; i < stArray.Length; i++)
    {
        string strTemp = stArray[i];
        string[] stArrayLinea = strTemp.Split(',');
        if (stArrayLinea[0] == "$GPGGA")
        {
            try
            {
                //UTC
                stUTC = FormatoHora(stArrayLinea[1]);
                lbUTC.Text = stUTC;

                //Latitud
                Double doLat = Convert.ToDouble(PuntoPorComa(stArrayLinea[2]));
                doLat = doLat / 100;
                string[] lat = doLat.ToString().Split(',');
                stLat = lat[0].ToString() + "," + ((Convert.ToDouble(lat[1]) / 60)).ToString("#####");
                stLat = SignoCoor(stLat, Convert.ToString(stArrayLinea[3]));
                tbLat.Text = stLat.ToString();

                //Longitud
                Double doLong = Convert.ToDouble(PuntoPorComa(stArrayLinea[4]));
                doLong = doLong / 100;
                string[] lon = doLong.ToString().Split(',');
                stLong = lon[0].ToString() + "," + ((Convert.ToDouble(lon[1]) / 60)).ToString("#####");
                stLong = SignoCoor(stLong, stArrayLinea[5]);
                tbLong.Text = stLong.ToString();

                //Altitud
                stAlt = stArrayLinea[9];
                tbAlt.Text = PuntoPorComa(stAlt);
            }
            catch
            {
                //No se pueden leer los valores GPS
                tbLat.Text = "GPS inválido";
                tbLong.Text = "GPS inválido";
            }
        }
    }
}

```



```
        }
    }
}

#endregion

public Form1()
{
    InitializeComponent();
    Datos = Obtencion_datos();
    Procesar_datos(Datos);
}
}
```