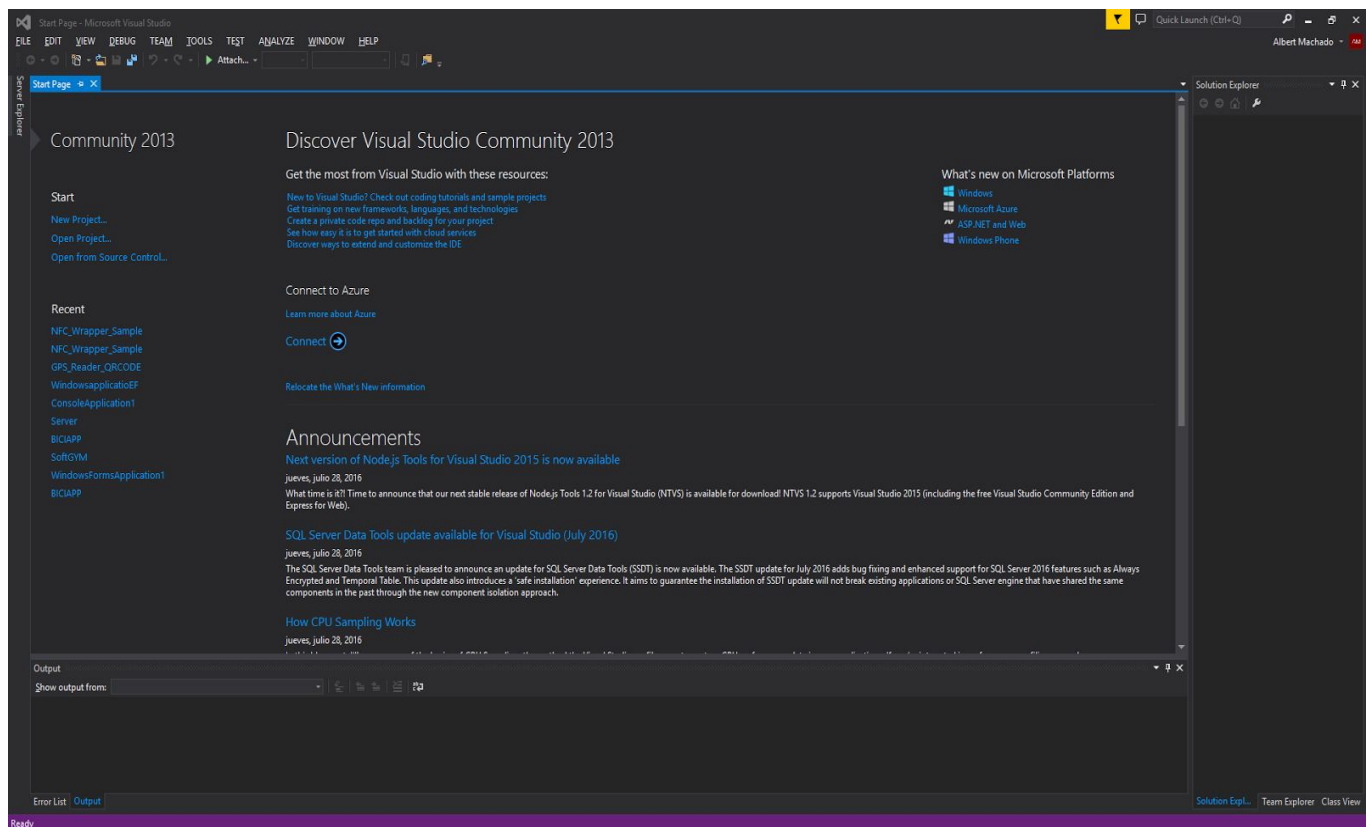
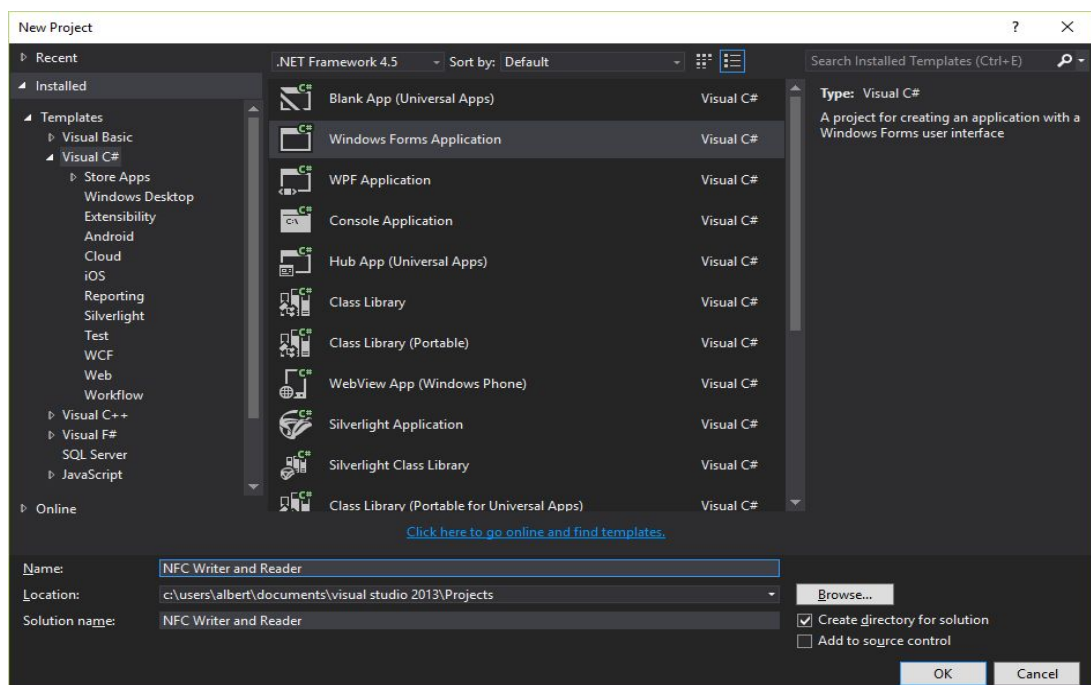


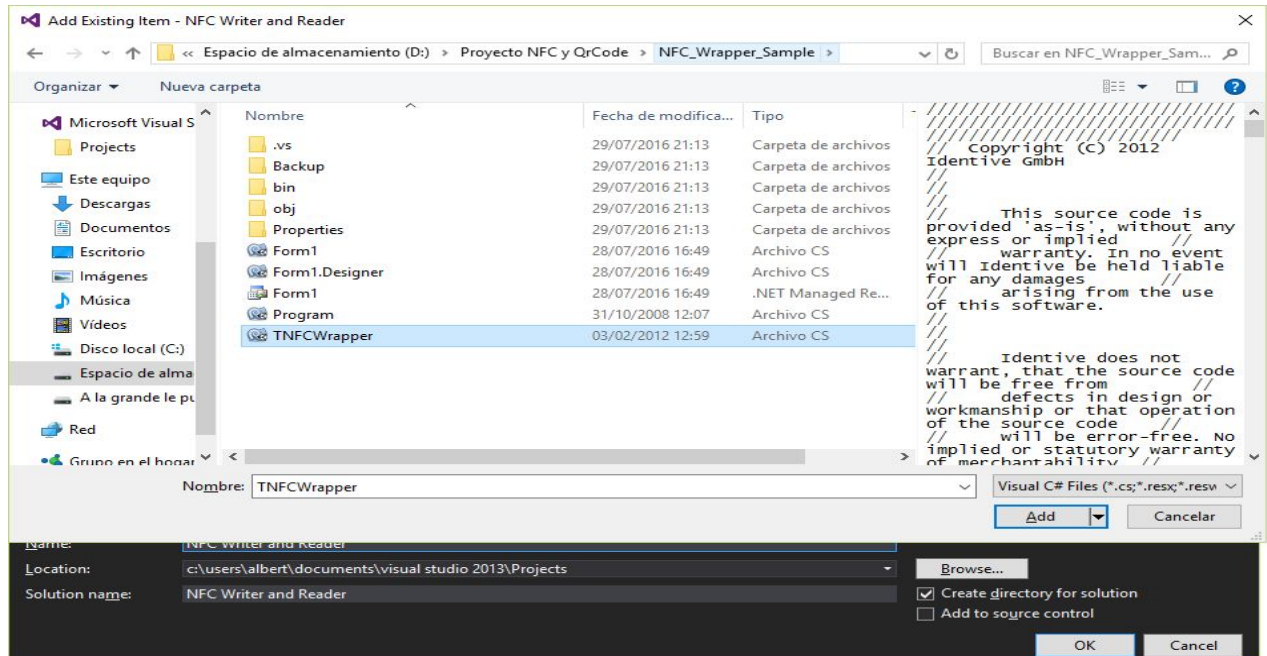
1. Iniciamos Visual studio



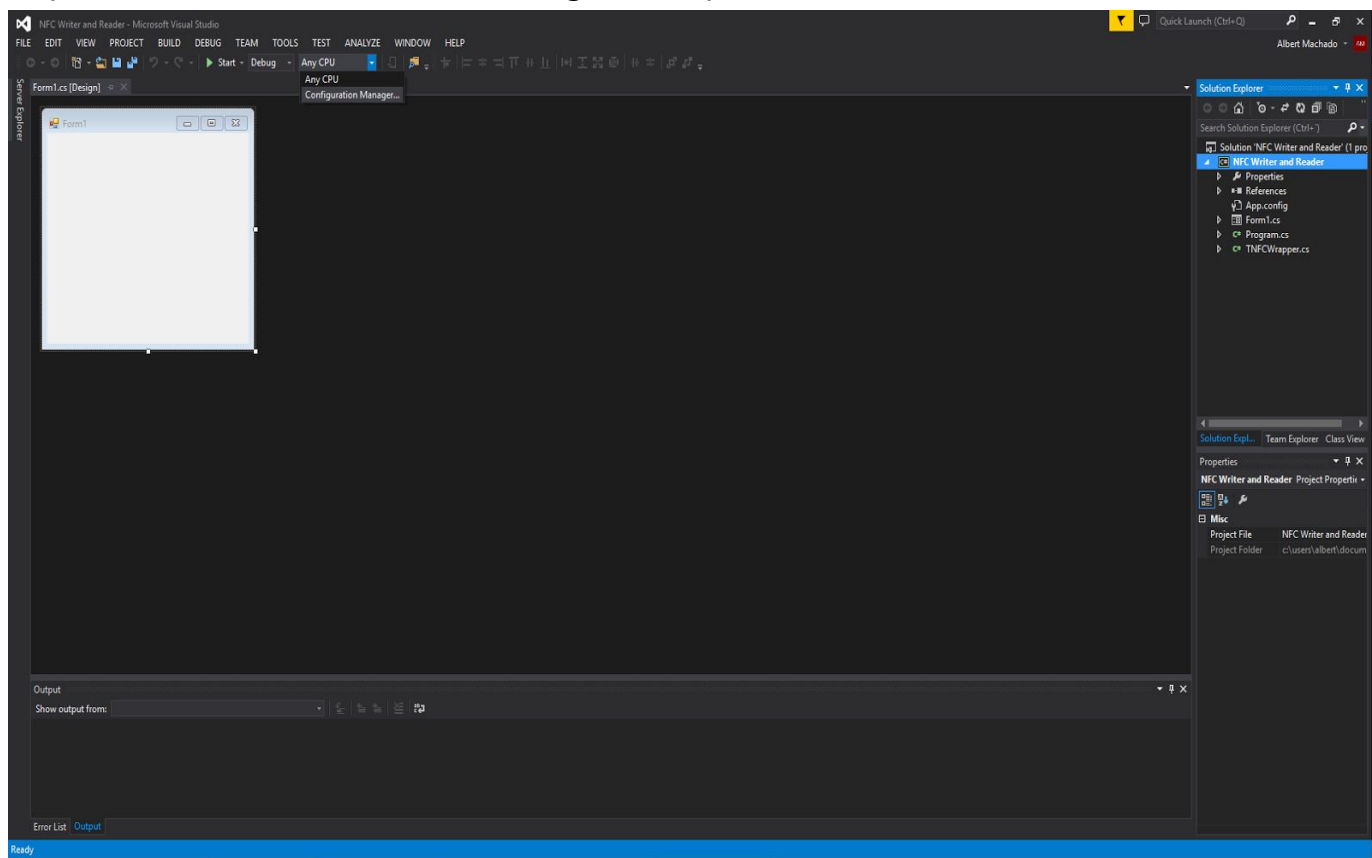
2. Usando la combinación de teclas **Ctrl + Shift + N** creamos un nuevo proyecto, escogemos la opción *Windows Forms Application*, nuestro directorio de trabajo, nombre del proyecto y por último apretamos OK.



3. Pulsamos la combinación **Shift + alt + A** Para agregar un ítem existente en este caso buscamos en nuestros directorios el archivo TNFC Wrapper, que será proporcionado.



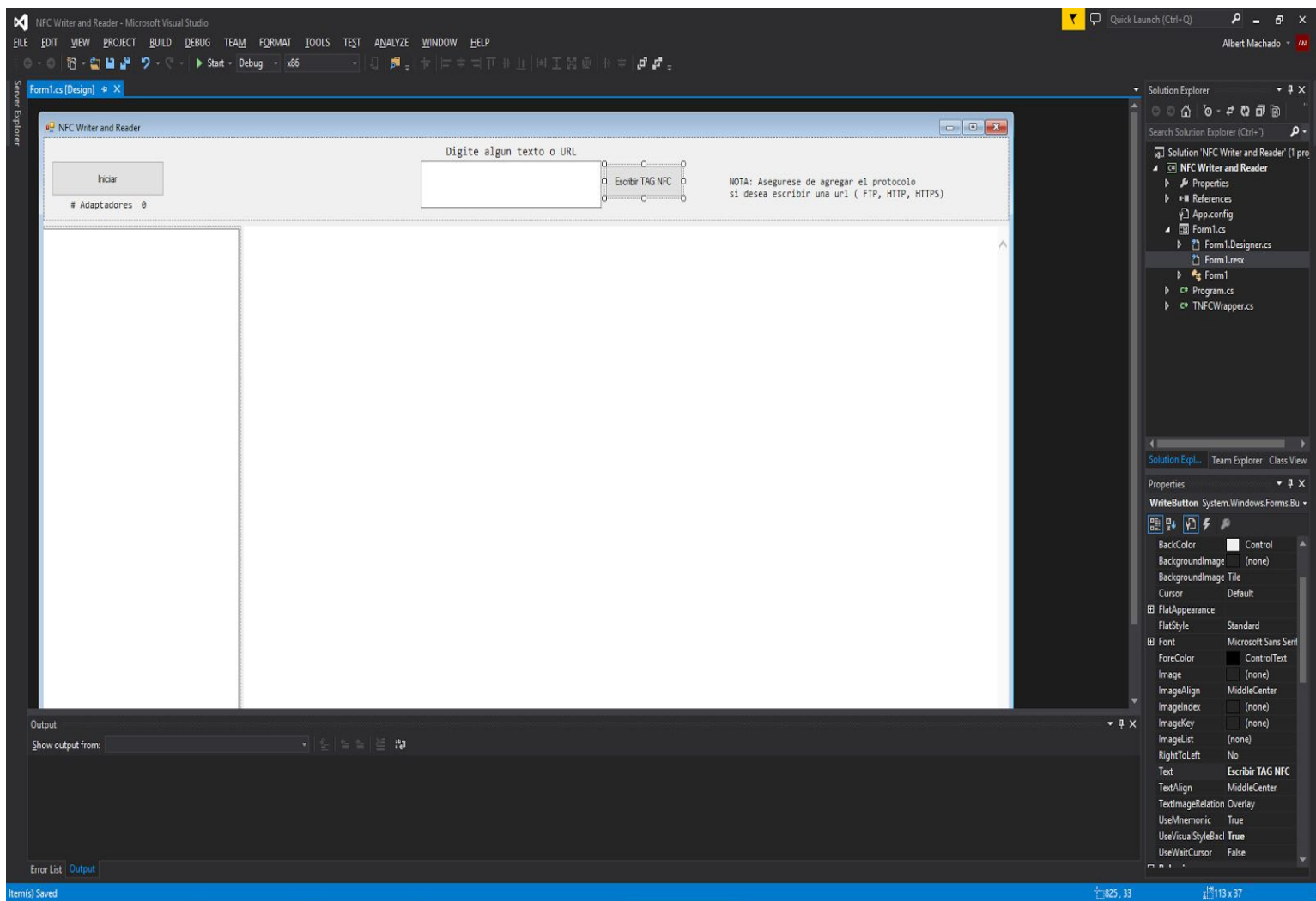
4. Cambiamos la plataforma en la que correremos el programa por un arquitectura x86, haciendo el siguiente procedimiento:



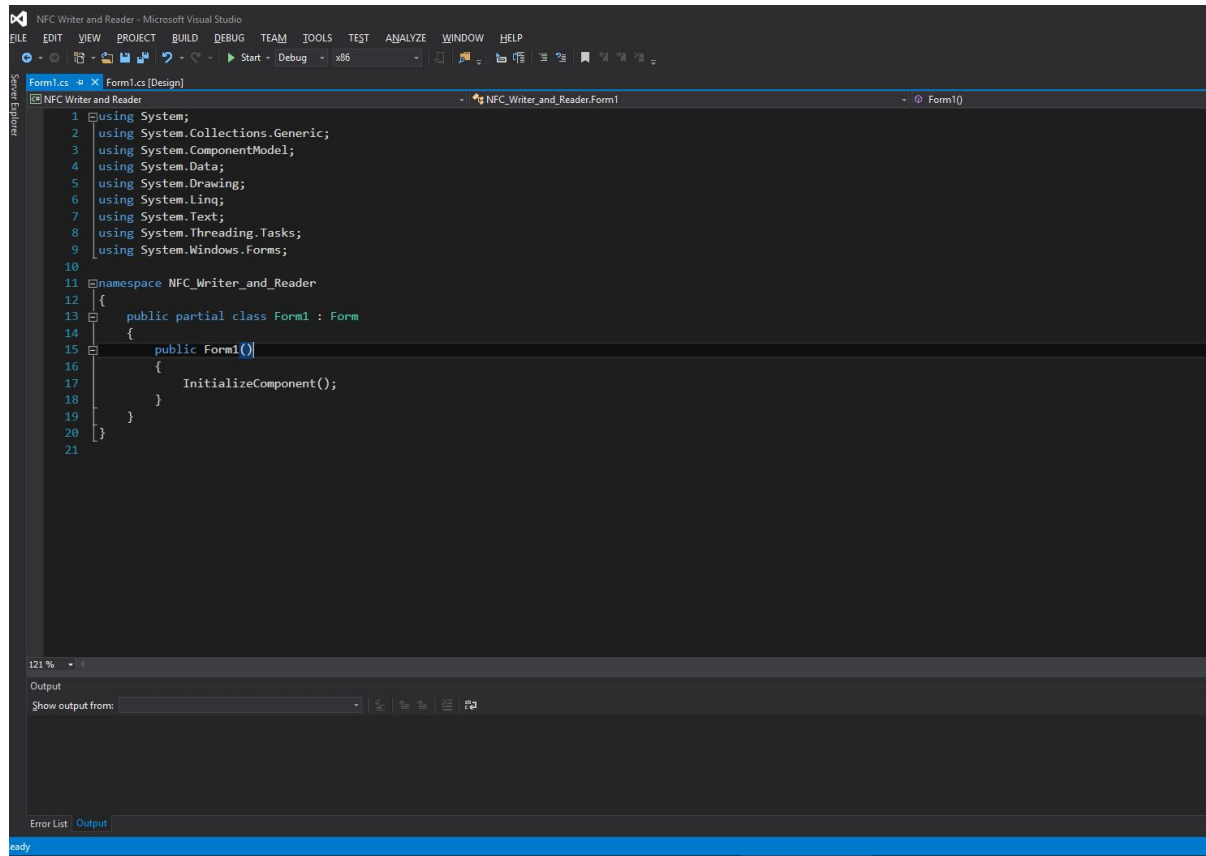
5. Realizamos el diseño de nuestra ventana:

Agregamos un título, el tamaño de la ventana principal y los elementos necesarios:

- 1 Panel que llamaremos *panel1* donde pondremos la cabecera
- 1 splitcontainer
- 1 botón que llamaremos *StartButton*
- 1 botón que llamaremos *WriteButton*, que por defecto estara deshabilitado
- Labels que contendrán mensajes indicativos tendrán el nombre por defecto.
- 1 TextBox que llamaremos *Texto*
- 1 TextBox que irá localizado en la parte izquierda del splitcontainer lo llamaremos *TextBox2*..
- 1 WebBrowser que irá en la parte derecha del splitcontainer y tendrá su nombre por defecto

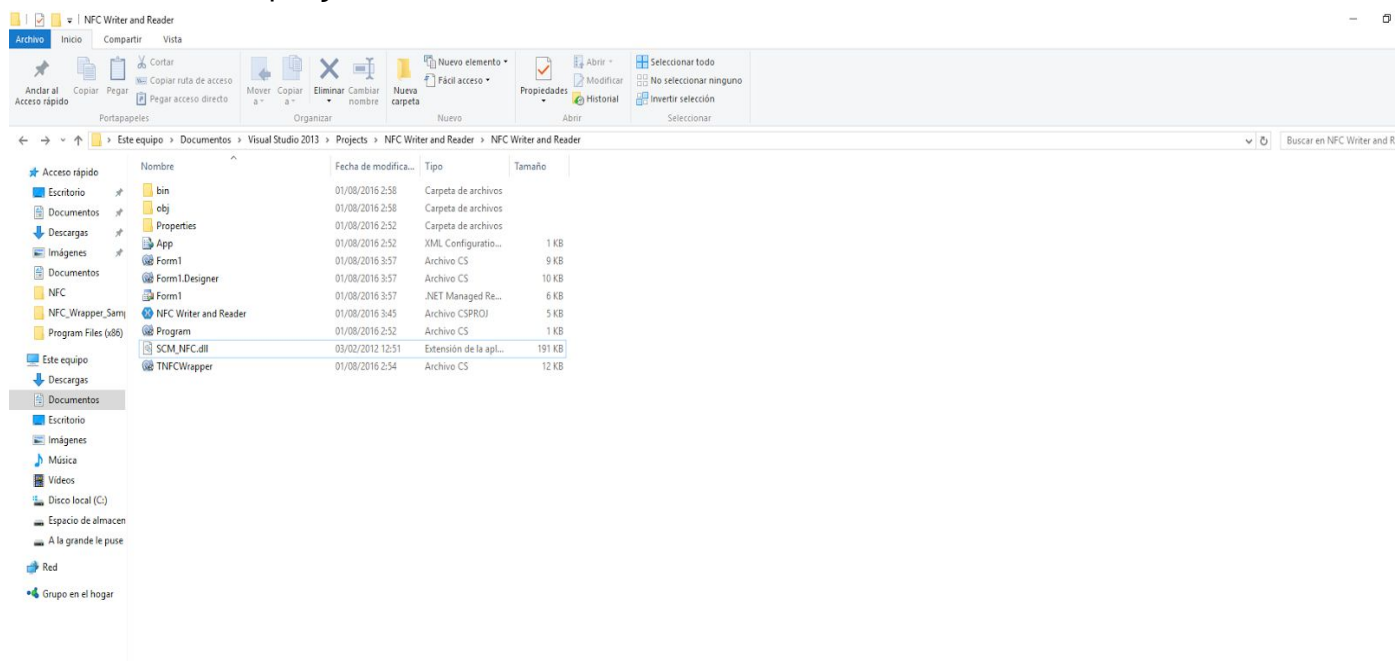


6. Luego nos vamos al código de la *Form1* y veremos el código por defecto



```
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 NFC_Writer_and_Reader
12 {
13     public partial class Form1 : Form
14     {
15         public Form1()
16         {
17             InitializeComponent();
18         }
19     }
20 }
21
```

7. Agregamos la dll SCM_NFC.dll que será proporcionada, a nuestro directorio del proyecto.



8. Agregamos el siguiente código en *Form1.cs* PD: *Tener cuidado con el nombre de la namespace, ya que si el proyecto no es nombrado igual que en esta guía, tendrá que agregar el código por secciones para evitar errores*

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Text.RegularExpressions; // UTILIZADO PARA LAS
EXPRESIONES REGULARES
using NFC_Wrapper_Sample;
namespace NFC_Writer_and_Reader
{
    public partial class Form1 : Form
    {
        [System.Runtime.InteropServices.DllImport("kernel32.dll",
EntryPoint = "LoadLibraryA")]
        static extern int LoadLibrary(string lpLibFileName); //
CARGAMOS LA LIBRERIA

        TNFCWrapper NFCWrapper;

        public Form1()
        {
            InitializeComponent();
        }
        string Mensaje = null; //VARIABLE MENSAJE GLOBAL

        public static bool ValidateUrl(string url) // METODO CON
EXPRESION REGULAR QUE VALIDA LAS URL
        {
            if (url == null || url == "") return false;

            Regex oRegExp = new
Regex(@"(http|ftp|https):\/\/([\\w-]+\\.)+([\\w- ./?%&=]*)?",
RegexOptions.IgnoreCase);
            return oRegExp.Match(url).Success;
        }

        private void LogMessage(string Msg) // FORMATO DE MENSAJES
        {
            textBox2.Text += Msg;
```

```

        textBox2.Text += Environment.NewLine;
        textBox2.Refresh();
        textBox2.SelectionStart = textBox2.Text.Length;
        textBox2.ScrollToCaret();
    }

[System.Security.Permissions.PermissionSet(System.Security.Permissions.SecurityAction.Demand, Name="FullTrust")] // PERMISOS
protected override void WndProc(ref Message aMessage) // TIPOS DE MENSAJES Y EVENTOS
{
    if (aMessage.Msg == TNFCWrapper.WM_NFC_NOTIFY)
    {
        string s = "";
        Int32 wParam = aMessage.WParam.ToInt32();
        switch (wParam)
        {
            case TNFCWrapper.NFC_NDEF_FOUND:
                s = "NFC_NDEF_FOUND Size = " + aMessage.LParam.ToString();
                break;
            case TNFCWrapper.NFC_DEVICE_CHANGED:
                s = "Dispositivo NFC Cargado";
                break;
            case TNFCWrapper.NFC_UNKNOWN_SERVICE:
                s = "Servicio NFC desconocido";
                break;
            case TNFCWrapper.NFC_CONNECTED:
                s = "Tag NFC Conectado";
                break;
            case TNFCWrapper.NFC_DISCONNECTED:
                s = "NFC Desconectado";
                break;
            case TNFCWrapper.NFC_IDLE:
                s = "NFC_IDLE";
                break;
            default:
                s = "Desconocido";
                break;
        }
        LogMessage(s);

        if (wParam == TNFCWrapper.NFC_DEVICE_CHANGED)
            DeviceCountLabel.Text = aMessage.LParam.ToString();
        if (wParam == TNFCWrapper.NFC_NDEF_FOUND)
            ReadNDEF();
    }
}

```

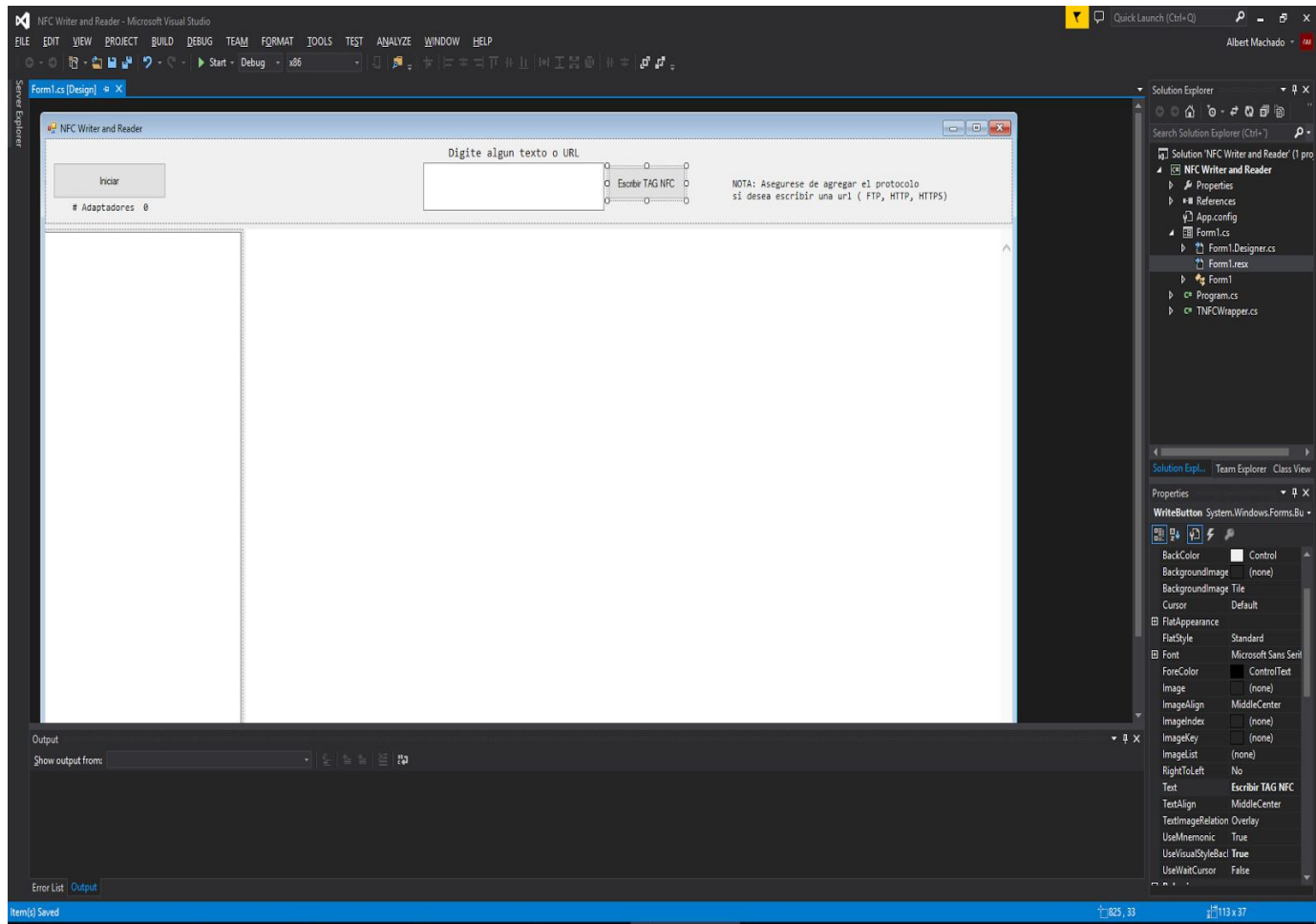


```

        base.WndProc(ref aMessage);
    }

```

9. En el diseñador damos doble click en el boton *Iniciar* para cargar el evento **StartButton_Click**



10. Agregamos el código que corresponde al evento y nos quedaria asi:

```

private void StartButton_Click(object sender, EventArgs e)
{
    // Verificar que exista la libreria
    if (LoadLibrary("SCM_NFC.dll") == 0)
    {
        LogMessage("NFC Wrapper no encontrada");
        LogMessage("Asegurate que el archivo SCM_NFC.dll
este presente");
        return;
    }
}

```



```

    }
    LogMessage("SCM_NFC.DLL Cargada exitosamente.");
    StartButton.Enabled = false;
    WriteButton.Enabled = true; // SE HABILITA EL BOTON
    PARA ESCRIBIR EN LA TARJETA

    //Init NFC Wrapper
    NFCWrapper = new TNFCWrapper();
    TNFCWrapper.Initialize((UInt32)Handle.ToInt32());

    //DECIRLE A NFC WRAPPER QUE EMPIECE CON LA LECTURA
    TNFCWrapper.StartListening();
    LogMessage("Por favor coloca una tag NFC");
}

```

11. Debajo agregamos el siguiente método

```

private void ReadNDEF()
{
    UInt32 DeviceCount = 0;
    UInt32 MessageCount = 0;
    UInt32 NextMessageSize = 0;
    UInt32 Result;

    if (NFCWrapper == null) return;

    // Get information about the message queue
    Result = TNFCWrapper.GetNDEFQueueInfo(ref DeviceCount,
    ref MessageCount, ref NextMessageSize);
    LogMessage("GetNDEFQueueInfo: " +
    NFCWrapper.NFCWrapperErrorToString(Result));
    if (Result != TNFCWrapper.ERR_SUCCESS) return;

    LogMessage(" DeviceCount      = " +
    DeviceCount.ToString());
    LogMessage(" MessageCount      = " +
    MessageCount.ToString());
    LogMessage(" NextMessageSize = " +
    NextMessageSize.ToString());

    //Resize the NDEF buffer accordingly to the size of
    the next message in the queue
    byte[] NDEF = new byte[NextMessageSize];
    UInt32 NDEFSize = NextMessageSize;
    TNFCAddress NFCAddress = new TNFCAddress();
}

```

```

        TMessageInfo MessageInfo = new TMessageInfo();

        //Read the NDEF message from the message queue
        Result = TNFCWrapper.ReadNDEF(ref NFCAddress, ref
MessageInfo, ref NDEF[0], ref NDEFSize);
        LogMessage("ReadNDEF: " +
NFCWrapper.NFCWrapperErrorToString(Result));
        if (Result != TNFCWrapper.ERR_SUCCESS) return;

        // CONVERTIR A XML
        string XML = "";
        string XML1 = null;
        string XML2 = null;
        Result = NFCWrapper.NDEF2XML(ref NDEF[0], NDEFSize,
ref XML);
        LogMessage("NDEF2XML: " +
NFCWrapper.NFCWrapperErrorToString(Result));
        if (Result != TNFCWrapper.ERR_SUCCESS) return;

        // print NDEF as XML
        textBox2.Clear();
        LogMessage(XML);

        // PRIMER FILTRADO del string donde, tomaremos linea
por linea el string XML y tomamos como delimitador los saltos de
linea
        // es alli donde se encuentra la linea del mensaje o
url. este es el resultado: <NDEF_URI:URI>Mensaje</NDEF_URI:URI>
        XML1 = XML.Split('\n')[12];
        // SEGUNDO FILTRADO del string anterior donde,
tomaremos linea por linea el string XML1, y tomamos como
delimitador ">"
        // este es el resultado en la posicion 1:
<NDEF_URI:URI>Mensaje
        XML2 = XML1.Split('<')[1];
        // ULTIMO FILTRADO del string anterior donde,
tomaremos linea por linea el string XML2, y tomamos como
delimitador ">"
        // este es el resultado en la posicion 1 : Mensaje
        Mensaje = XML2.Split('>')[1];

        if (ValidateUrl(Mensaje) == true) // VALIDAMOS QUE SI
MENSAJE ES UNA URL ENTONCES
        {
            webBrowser1.Navigate(Mensaje); // ENVIAR URL AL
NAVEGADOR Y EJECUTARSE
        }
        else

```

```

        {
            MessageBox.Show(Mensaje, "Mensaje"); // SINO ES
            UNA URL ENTONCES ENVIAR ALERTA CON EL MENSAJE
        }
        LogMessage(" ");
    }

```

12. Finalmente hacemos el mismo procedimiento del paso 9 en este caso con el botón que tiene como texto “Escribir tag NFC” y dentro del evento agregaremos el siguiente código que hará que nos quede así:

```

private void WriteButton_Click(object sender, EventArgs e)
{
    if (NFCWrapper == null) return;
    UInt32 Result;
    //Create Smart Poster NDEF Message
    // INFORMACION NECESARIO PARA ENVIARLE A
    TNFCWRAPPER.CS COMO PARAMETRO

    string URI = Texto.Text.ToString();
    string Comment = "";
    string Language = "en-US";
    string TargetType = "";
    UInt32 Size = 0;
    byte Action = 0;
    UInt32 NDEFSize = 1000;
    byte[] NDEF = new byte[NDEFSize];
    Result = TNFCWrapper.CreateNDEFSp(URI, Comment,
    Language, ref Action, ref Size, TargetType, ref NDEF[0], ref
    NDEFSize);
    LogMessage("CreateNDEFSp: " +
    NFCWrapper.NFCWrapperErrorToString(Result));
    if (Result != TNFCWrapper.ERR_SUCCESS) return;

    // convert NDEF into XML
    string XML = "";
    Result = NFCWrapper.NDEF2XML(ref NDEF[0], NDEFSize,
    ref XML);
    LogMessage("NDEF2XML: " +
    NFCWrapper.NFCWrapperErrorToString(Result));
    if (Result != TNFCWrapper.ERR_SUCCESS) return;

    // print NDEF as XML

```

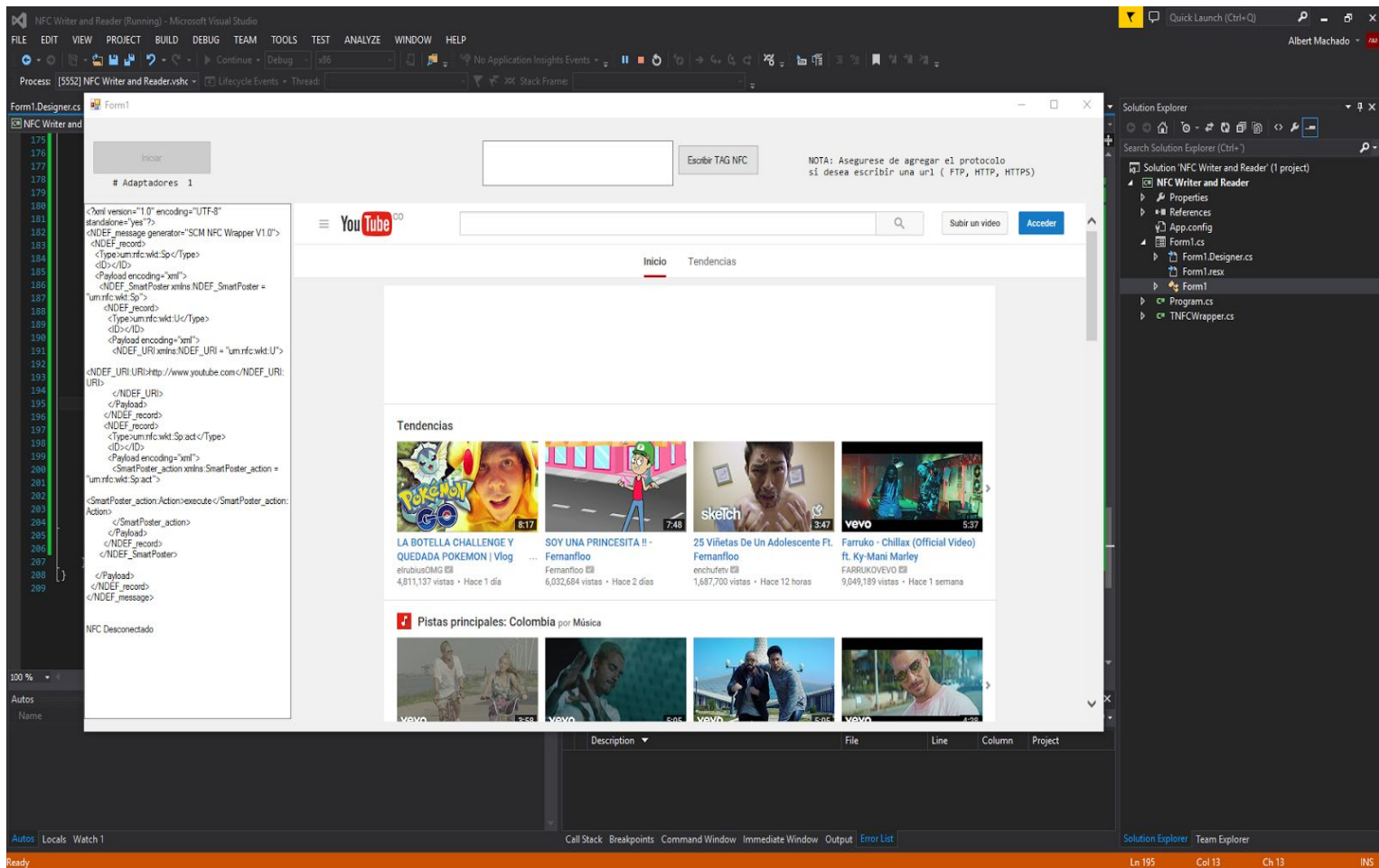
```

        LogMessage(XML);
        LogMessage(" ");

        // ESCRIBIMOS EL NDEF A LA ETIQUETA
        LogMessage("Ponga una NFC Tag en los proximos 5 segundos...");
        Texto.Clear();
        TNFCAddress NFCAddress = new TNFCAddress();
        TMessageInfo MessageInfo = new TMessageInfo();
        Result = TNFCWrapper.WriteNDEF(ref NFCAddress, ref MessageInfo, ref NDEF[0], ref NDEFSize, false, true, 5);
        LogMessage("NDEF2XML: " +
        NFCWrapper.NFCWrapperErrorToString(Result));
        LogMessage("");
    }

```

13. Agregamos los corchetes necesarios, construimos, ejecutamos y testeamos:



NFC Writer and Reader (Running) - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP

Process: [5552] NFC Writer and Reader.vshc Lifecycle Events Thread: XXX Stack Frames

Form1.Designer.cs

NFC Writer and

Form1

175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209

Inicio

Escribir TAG NFC

NOTA: Asegurese de agregar el protocolo si desea escribir una url (FTP, HTTP, HTTPS)

Adaptadores 1

```
<?xml version="1.0" encoding="UTF-8"
standalone="yes" ?>
<NDEF_message generator="SCM NFC Wrapper V1.0">
  <NDEF_record>
    <Type>urn:nfc:wkt:Sp</Type>
    <ID></ID>
    <Payload encoding="xml">
      <NDEF_SmartPoster xmlns:NDEF_SmartPoster =
"urn:nfc:wkt:Sp">
        <NDEF_record>
          <Type>urn:nfc:wkt:U</Type>
          <ID></ID>
          <Payload encoding="xml">
            <NDEF_URI xmlns:NDEF_URI = "urn:nfc:wkt:U">
              <NDEF_URI URI>Holaaaaa</NDEF_URI URI>
            </NDEF_URI>
          </Payload>
        </NDEF_record>
        <Type>urn:nfc:wkt:Sp:act</Type>
        <ID></ID>
        <Payload encoding="xml">
          <SmartPoster_action xmlns:SmartPoster_action =
"urn:nfc:wkt:Sp:act">
            <SmartPoster_action:Action>execute</SmartPoster_action:
Action>
            <SmartPoster_action>
              <Payload>
                <NDEF_record>
                  <NDEF_SmartPoster>
                    <Payload>
                      <NDEF_record>
                        <NDEF_message>
                          NFC Desconectado
                        </NDEF_message>
                      </NDEF_record>
                    </Payload>
                  </NDEF_SmartPoster>
                </NDEF_record>
              </Payload>
            </SmartPoster_action>
          </SmartPoster_action>
        </Payload>
      </NDEF_SmartPoster>
    </NDEF_record>
  </NDEF_message>
</NDEF_message>
```

Mensaje

Holaaaaa

Aceptar

Autos

Name

Description	File	Line	Column	Project
-------------	------	------	--------	---------

Autos Locals Watch 1

Call Stack Breakpoints Command Window Immediate Window Output Error List

Solution Explorer Team Explorer

Ready

Ln 195 Col 13 Ch 13 INS