Universidad Rafael Landívar

Facultad de Ingeniería

Licenciatura en Sistemas

Programación Avanzada

**Proyecto 1**

**“Red Social”**

Marlon Roches

Carné: 12509-18

Guatemala, 16 de octubre del 2018

**INDICE**

[Introducción 3](#_Toc513224648)

[Objetivos 4](#_Toc513224649)

[Análisis 5](#_Toc513224650)

[Diseño 6](#_Toc513224651)

[Diagrama de Flujo 9](#_Toc513224652)

[Seudocódigo 10](#_Toc513224654)

[Conclusiones 27](#_Toc513224655)

[Bibliografía 28](#_Toc513224656)

# Introducción

En el presente proyecto, “Red Social”, consiste en una aplicación de escritorio que te permite visualizar tus datos desde un archivo .txt, basándose en la lectura de estos, puedes ver toda la información desde un formulario padre el cual esta desarrollado para ver varias ventanas desde su interior.

Antes de entrar a la aplicación, debes tener cargados tus datos, ya que, si no, la aplicación no tendrá que mostrar.

El proyecto a su vez consiste en:

Una aplicación que simula una red social con sus aspectos principales, newsfeed, en el cual se muestran los post e imágenes de nuestros amigos, contactos, nos permite visualizar los datos de nuestros contactos como el estado, si eta bloqueado o no, etc. Messeger, es un visualizador de chats con el cual podemos ver las conversaciones que tenemos con nuestro amigos. Login, es el primer form que se muestra y valida que seas el usuario que ingresa, de lo contrario no te dejara entrar.

# Objetivos

* Cumplir con las condiciones que nos pide.
* Crear un simulador de red social basado en archivos de texto.
* Manejar la puntuación conforme al nivel y al tiempo al que se enfrente.
* Un nivel de controles dinámicos, el cual no se limita a estar ya definido antes de ingresar al programa.

# Análisis

* **Entradas**:

1. Nombre
2. Contraseña
3. Archivos De Texto

* **Procesos:**

1. Lectura de archivos de texto, manejos de pilas, colas y listas enlazadas. Lectura de vectores dinámicos.

* **Salidas:**

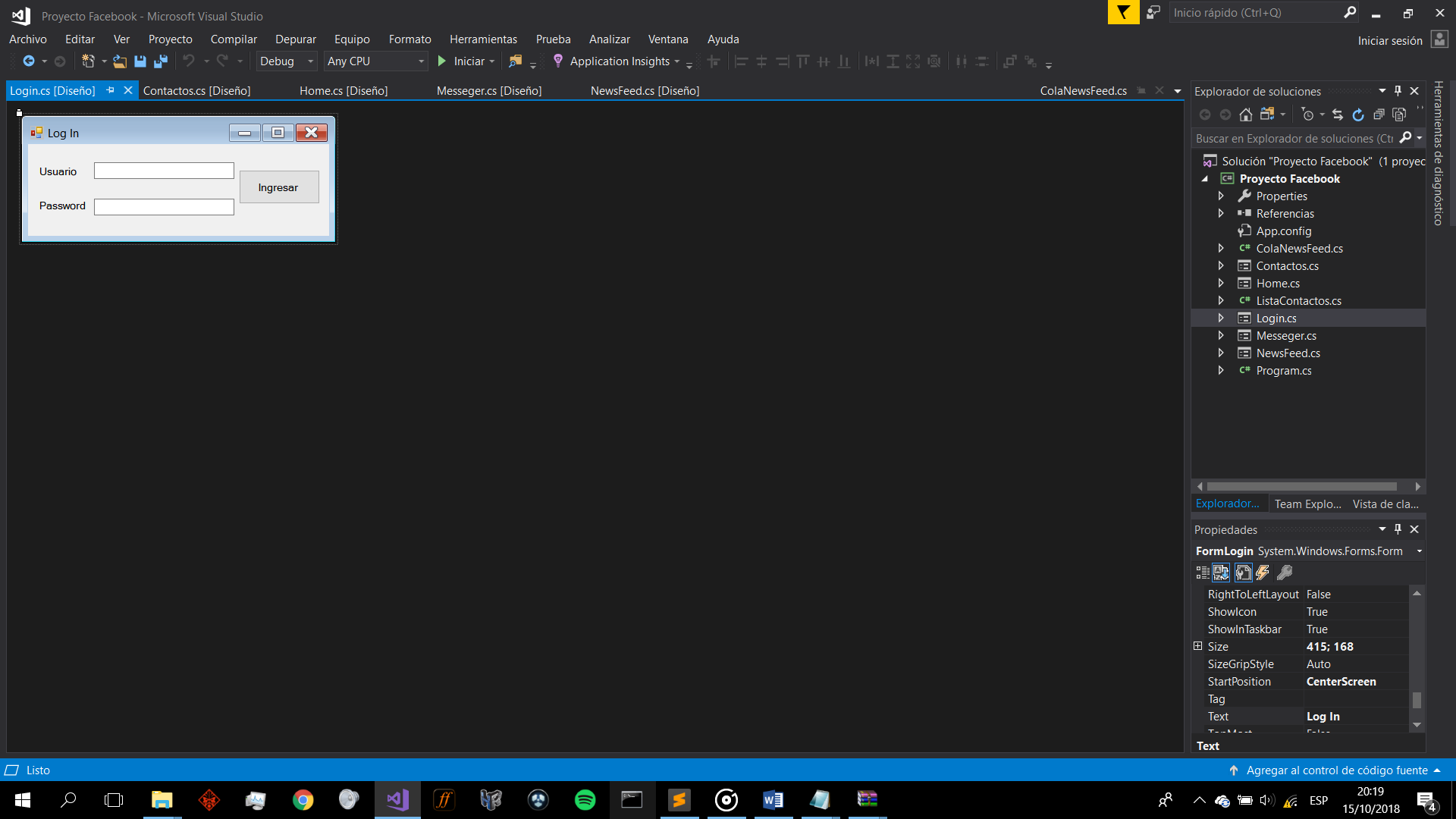
1. Newsfeed
2. Contactos
3. Messenger

* **Validaciones:**

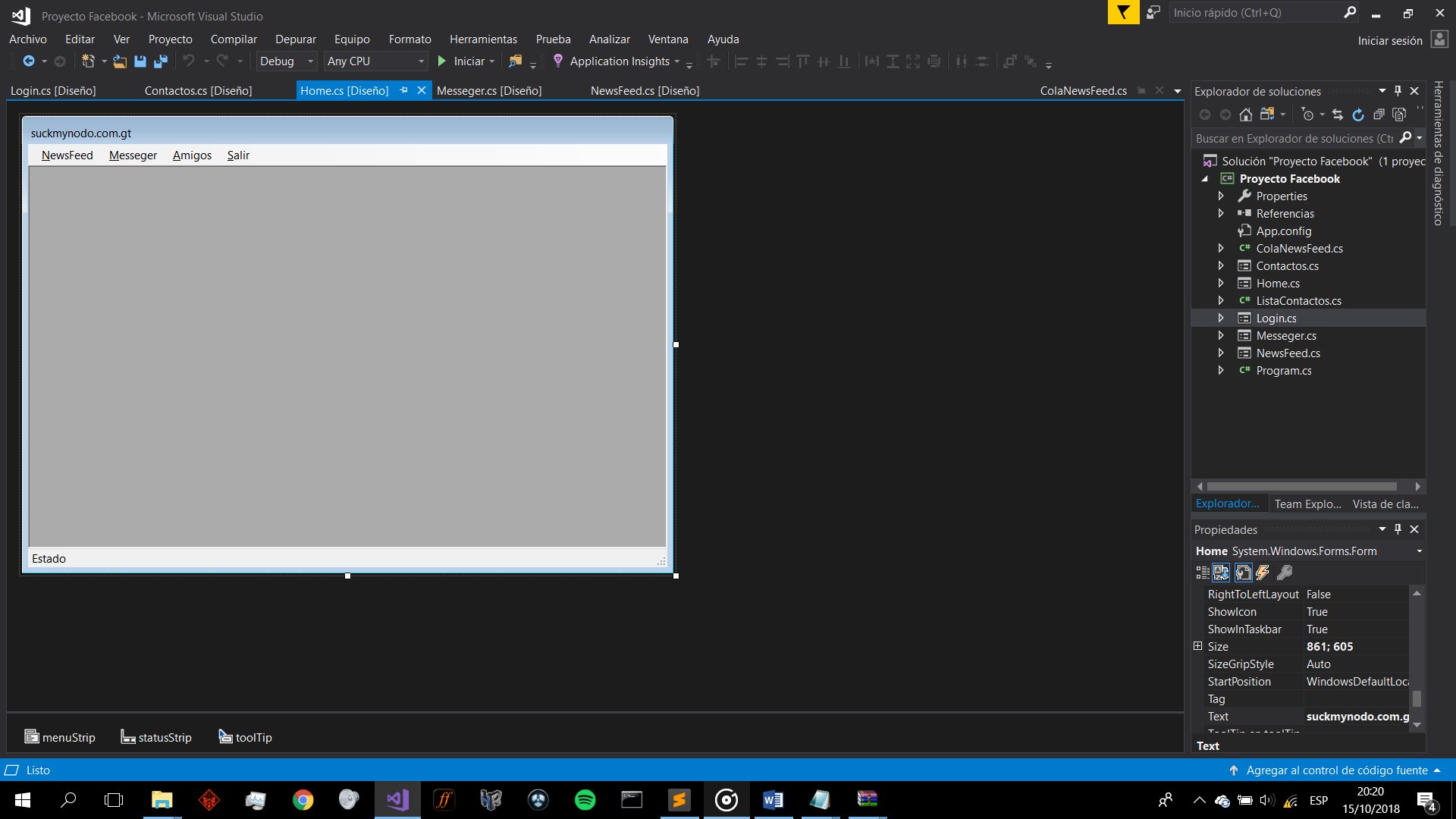
1. El Login.
2. Ver si el contacto esta bloqueado o no.

# Diseño

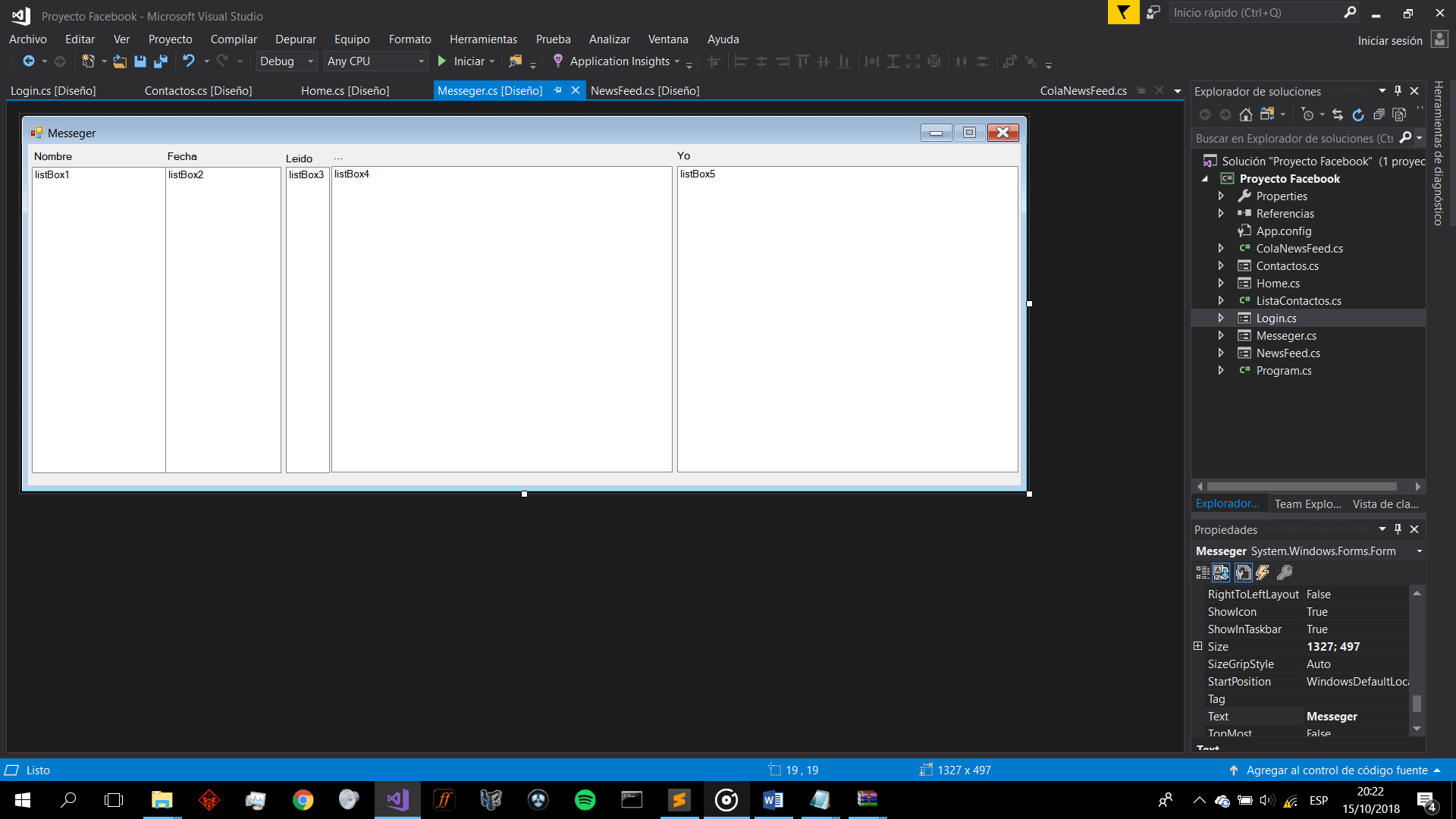
Login



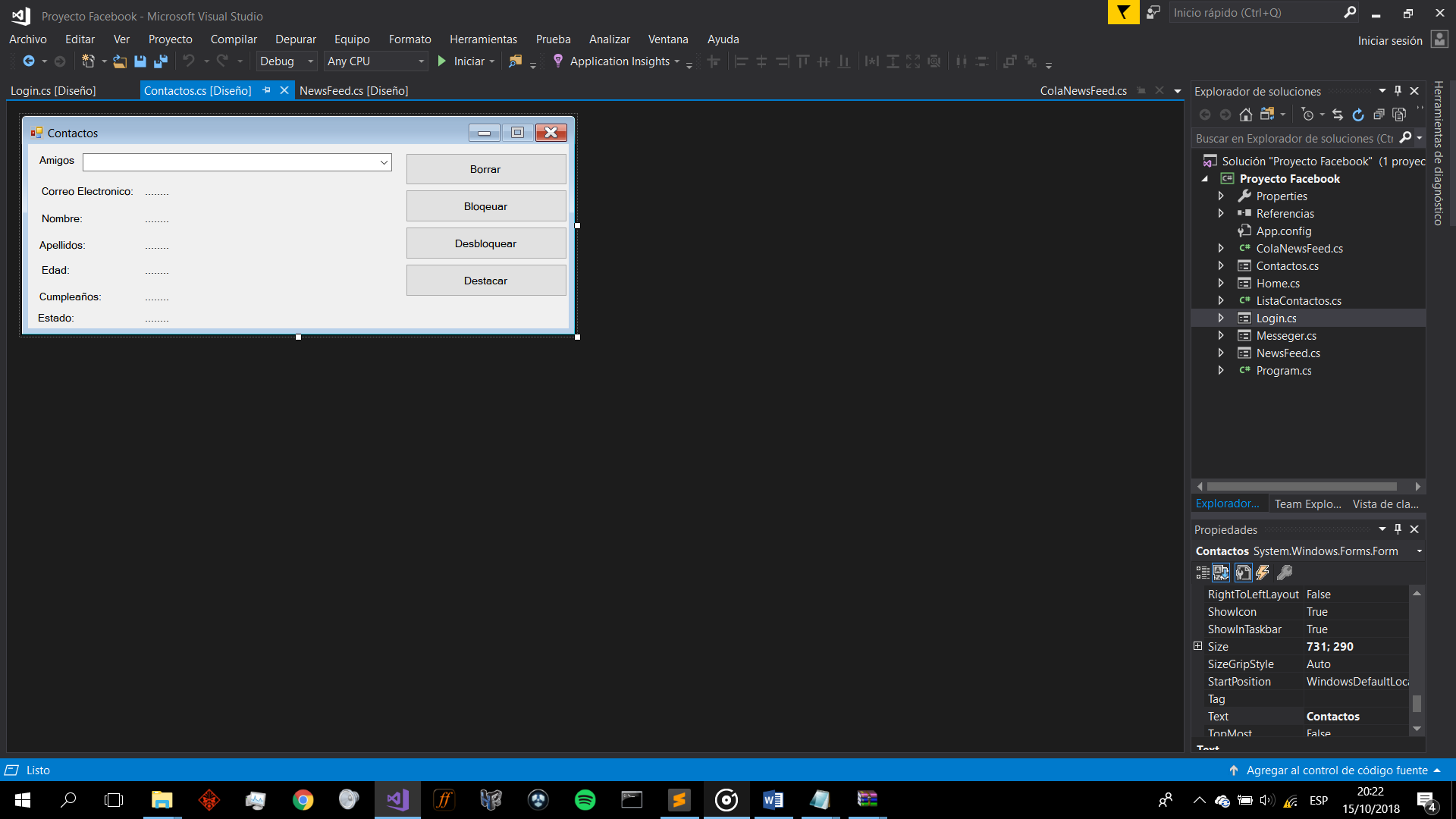
Home



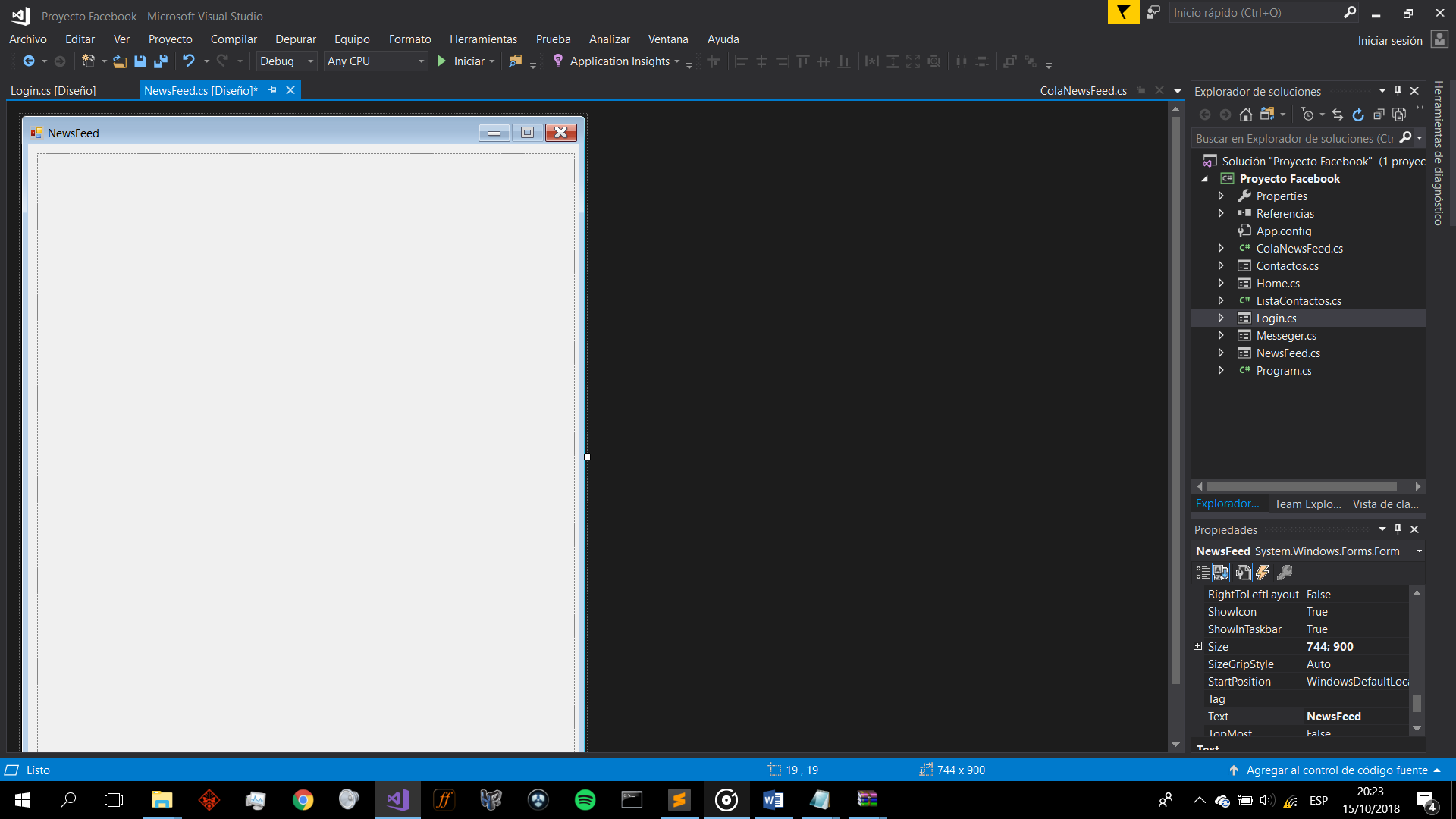
Messenger



Contactos

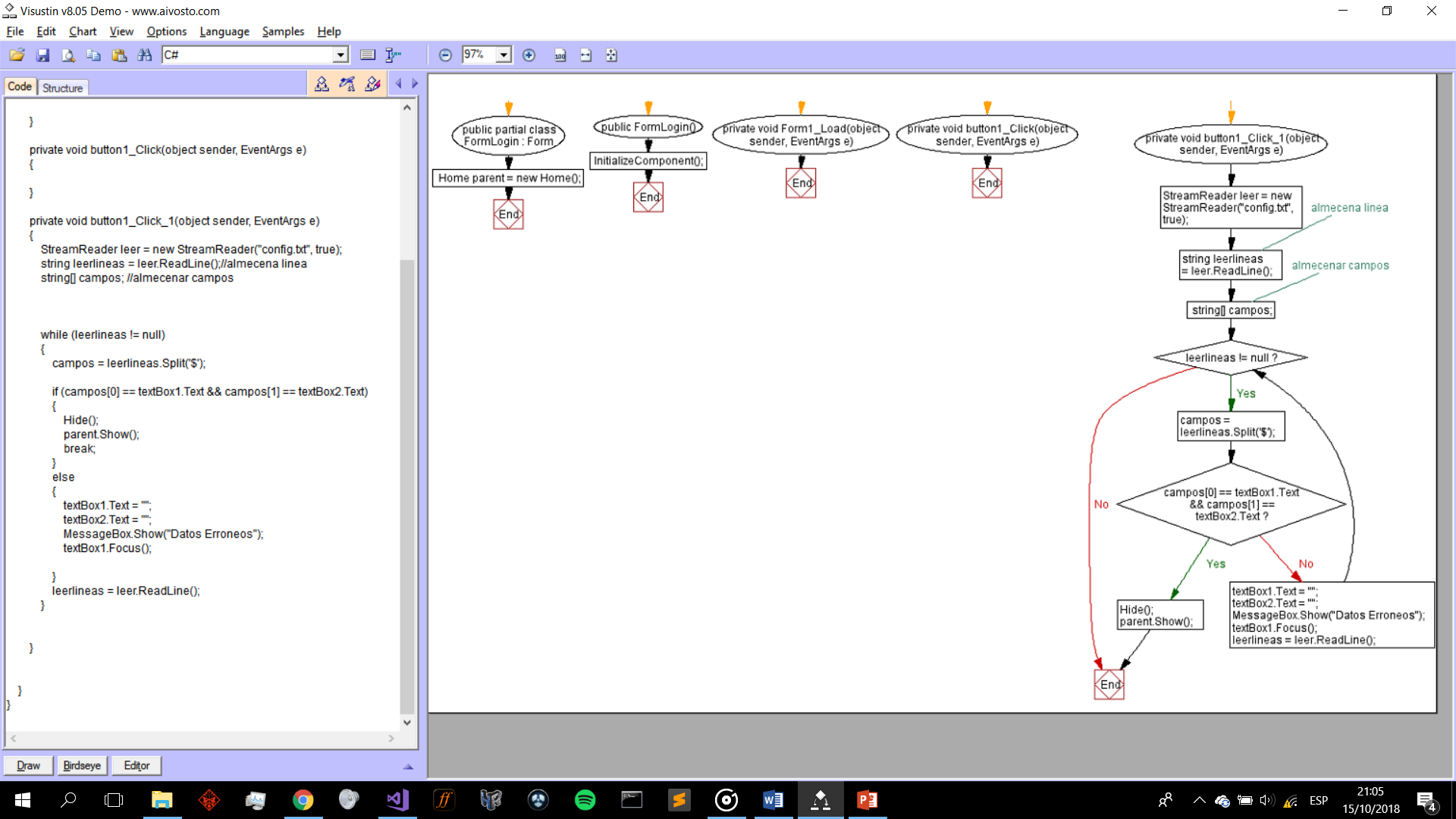


Newsfeed

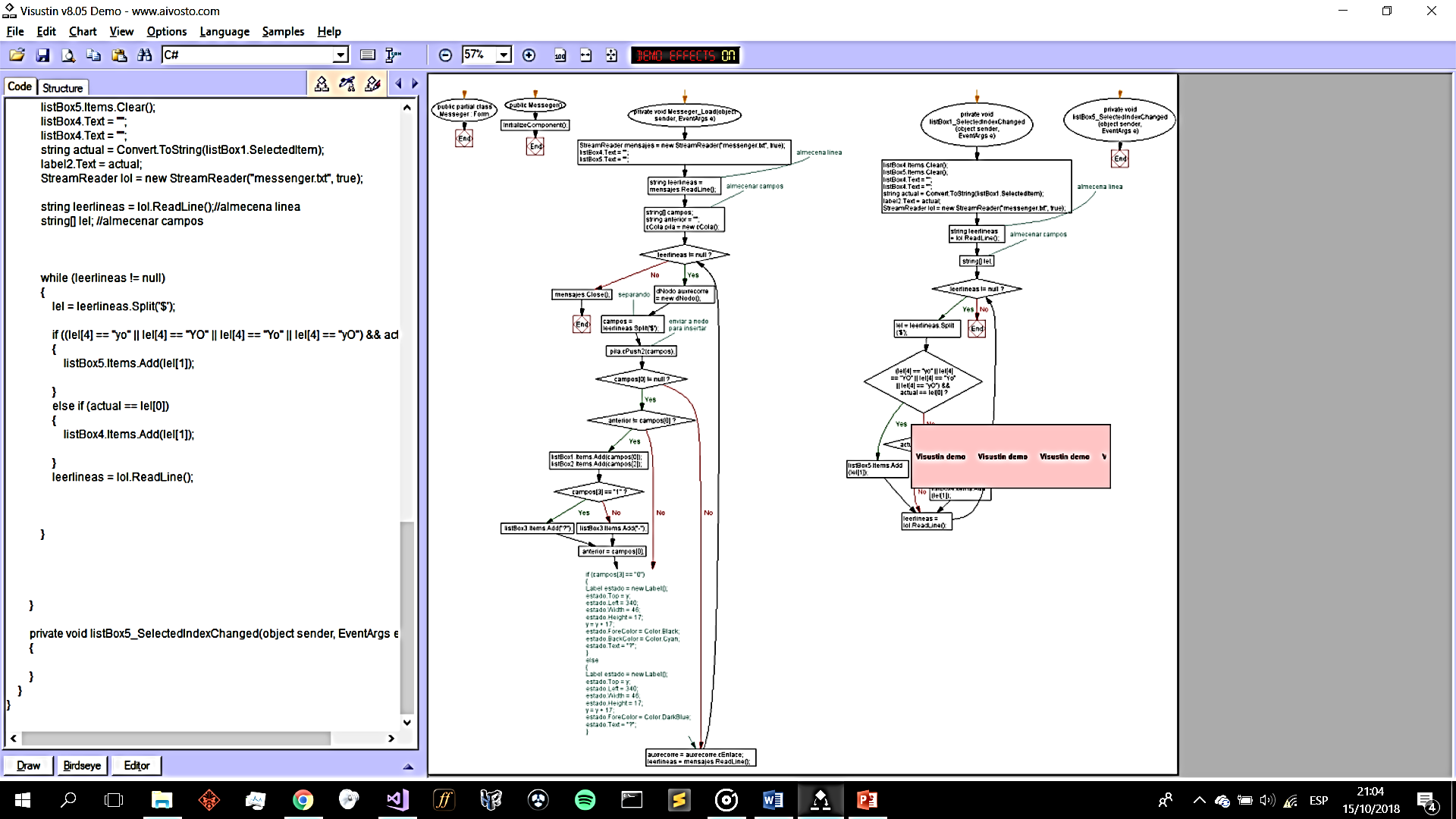


# Diagrama de Flujo

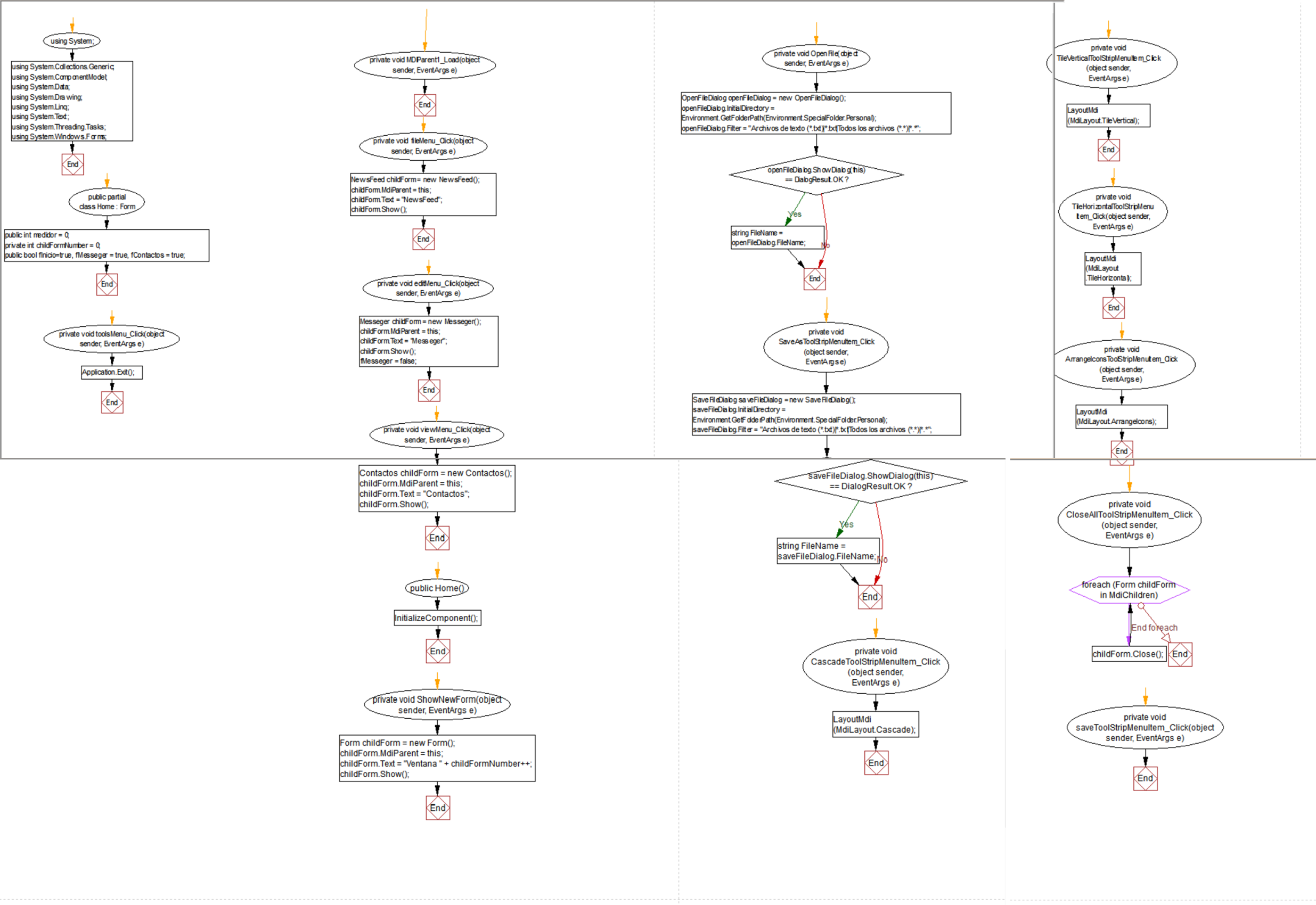
**LOGIN**

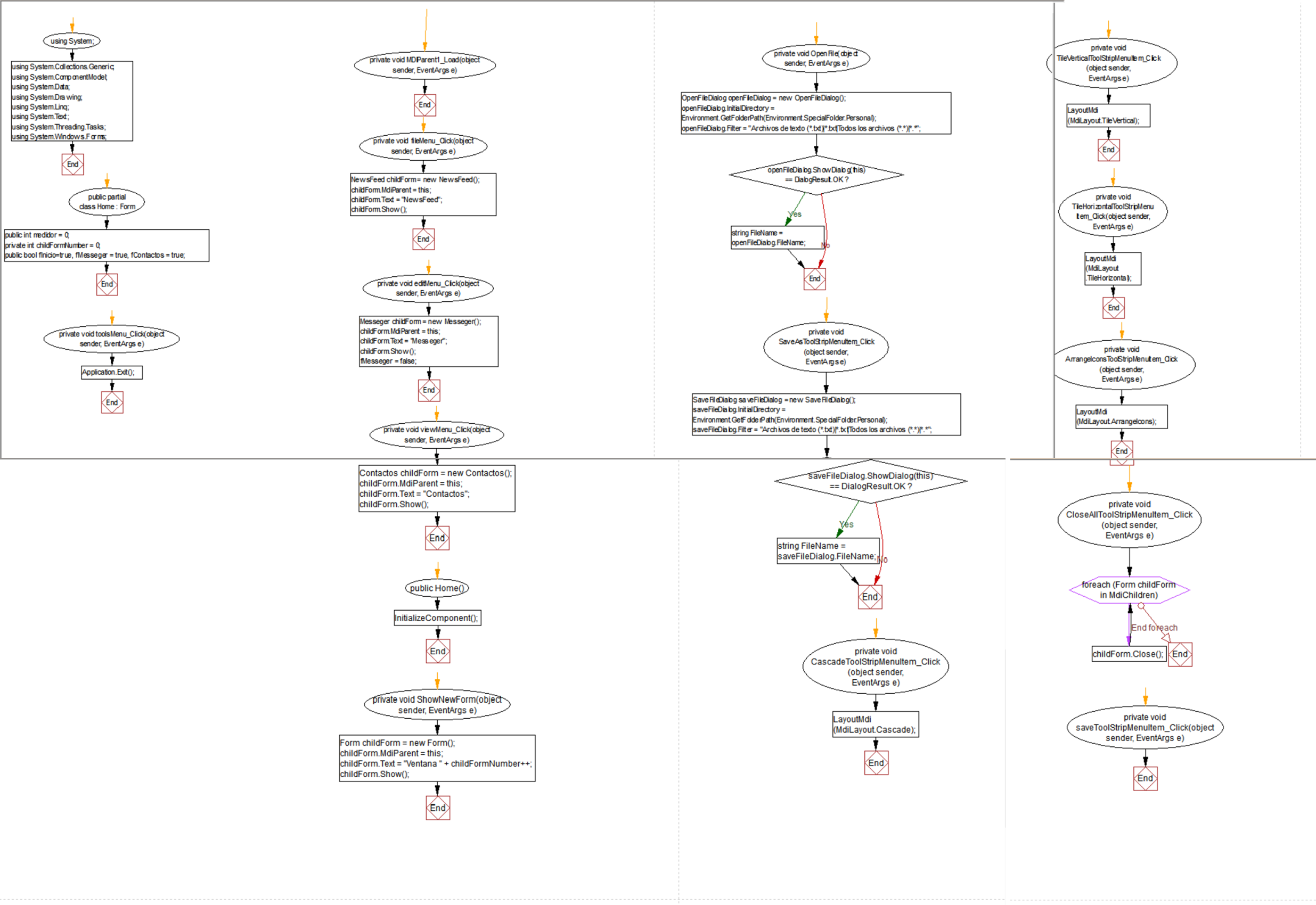
****

**MESSEGER**

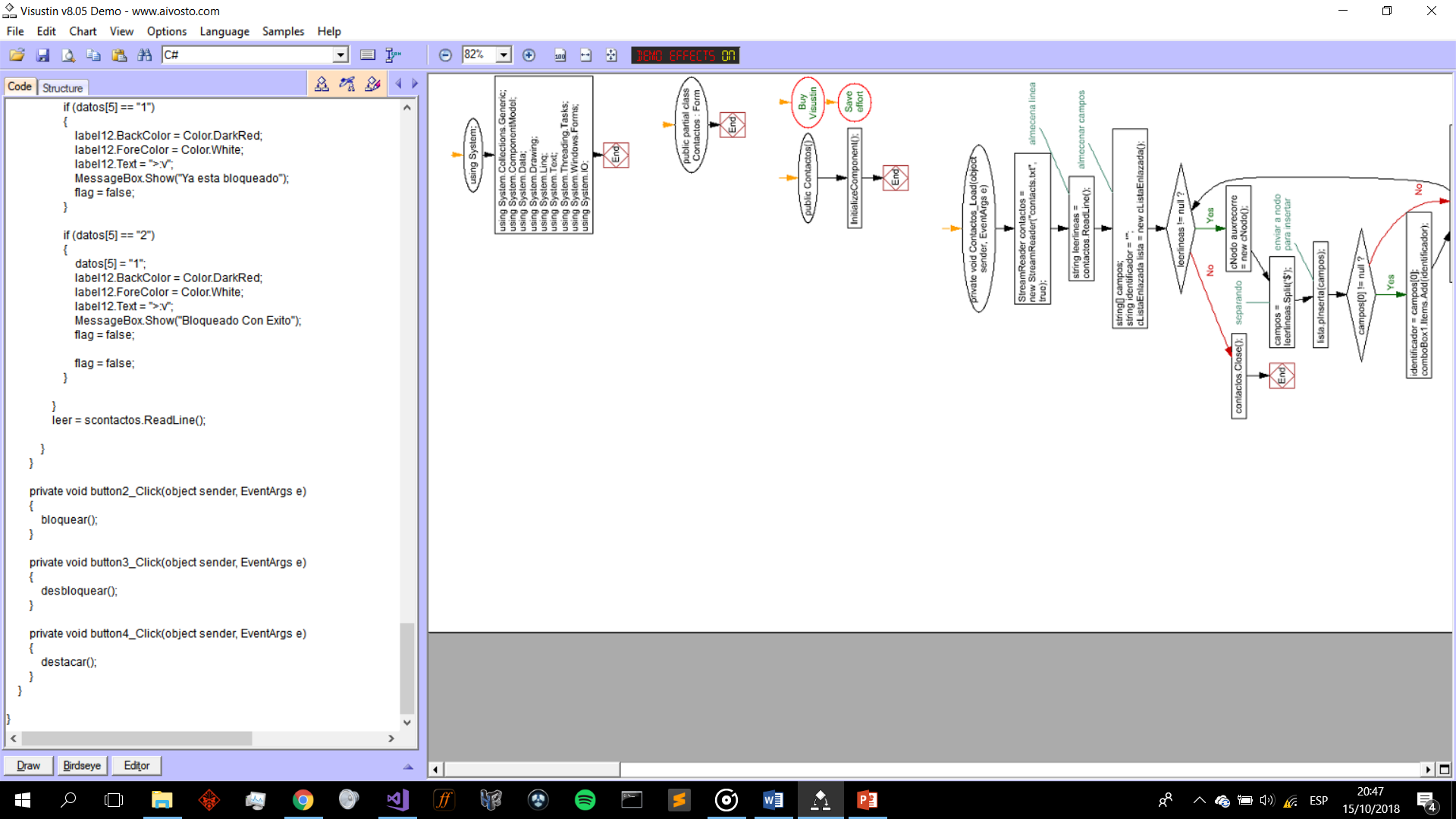
****

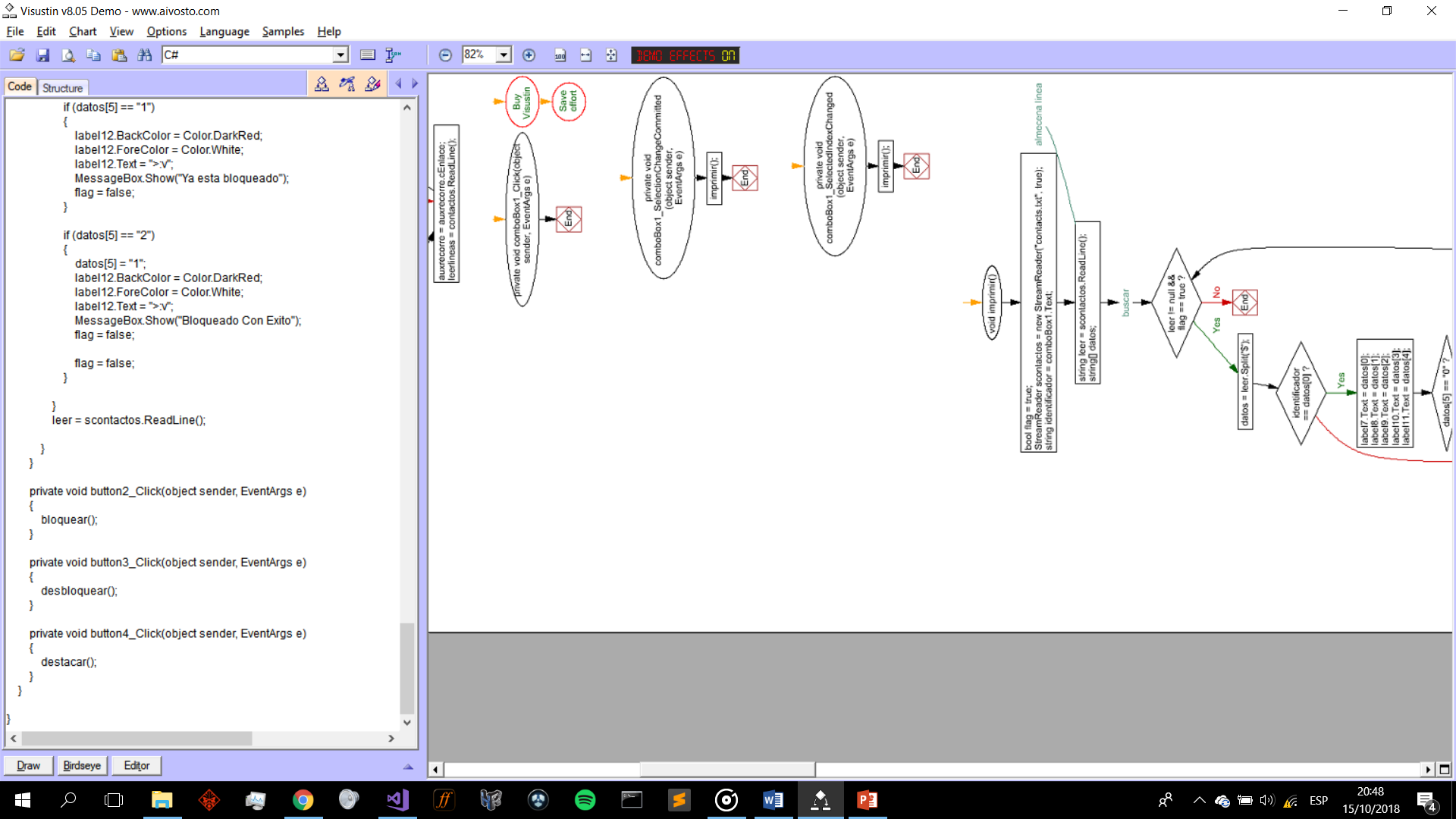
**HOME**

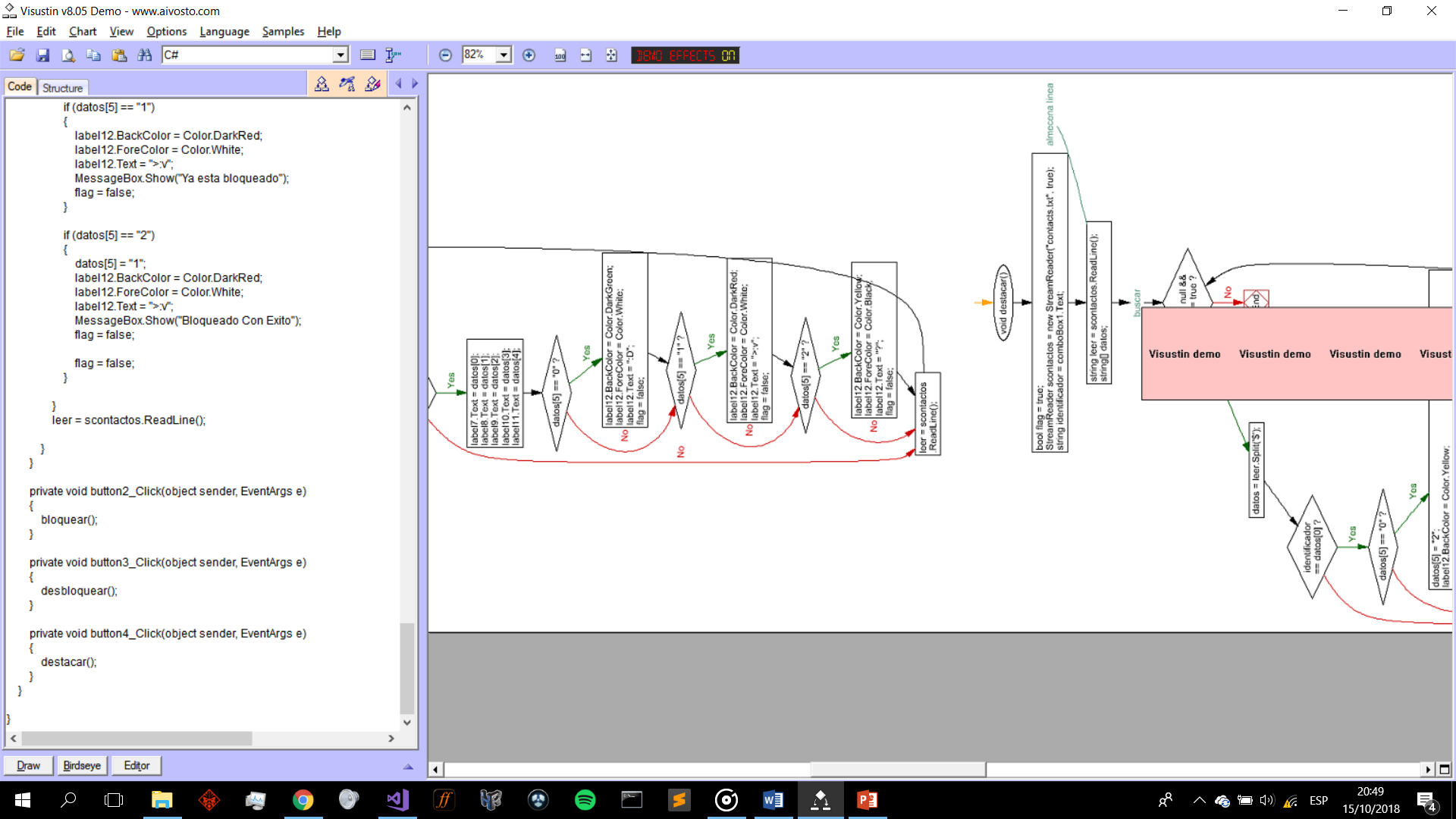
****

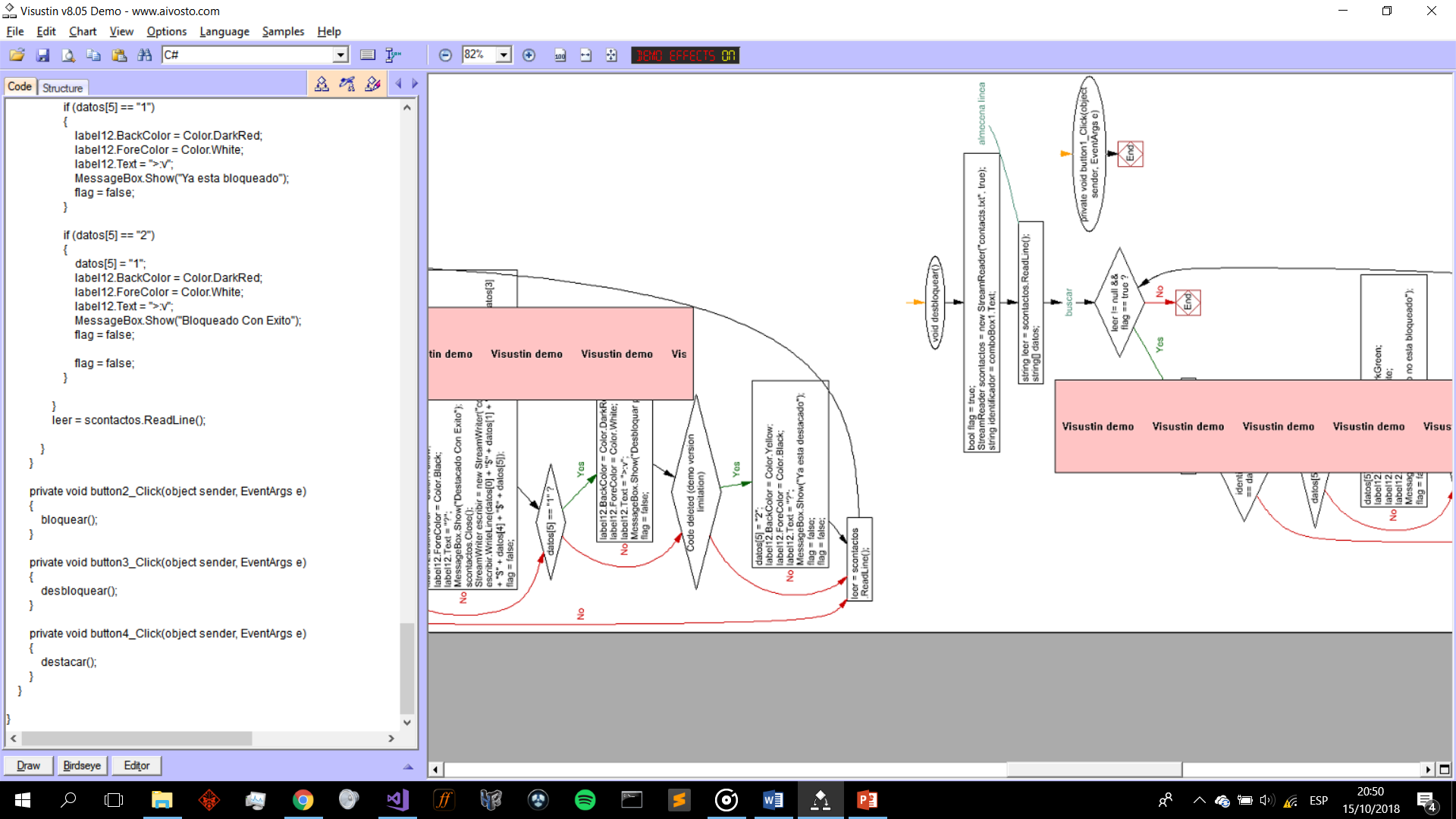
****

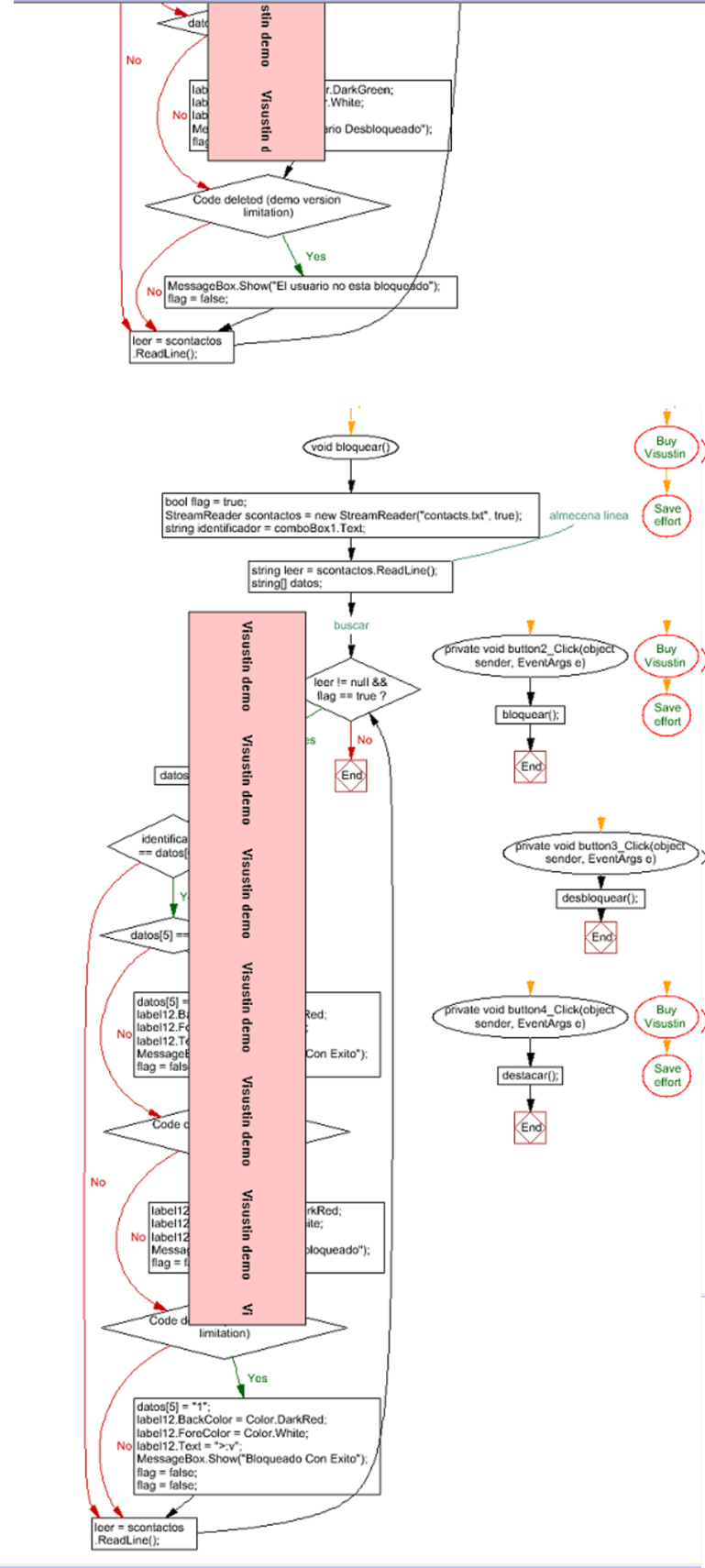
**CONTCTOS**

****

****

****

****

****

# Pseudocódigo

*LOGIN*

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.IO;

namespace Proyecto\_Facebook

{

public partial class FormLogin : Form

{

Home parent = new Home();

public FormLogin()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

}

private void button1\_Click\_1(object sender, EventArgs e)

{

StreamReader leer = new StreamReader("config.txt", true);

string leerlineas = leer.ReadLine();//almecena linea

string[] campos; //almecenar campos

while (leerlineas != null)

{

campos = leerlineas.Split('$');

if (campos[0] == textBox1.Text && campos[1] == textBox2.Text)

{

Hide();

parent.Show();

break;

}

else

{

textBox1.Text = "";

textBox2.Text = "";

MessageBox.Show("Datos Erroneos");

textBox1.Focus();

}

leerlineas = leer.ReadLine();

}

}

}

}

*HOME*

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;

namespace Proyecto\_Facebook

{

public partial class Home : Form

{

public int medidor = 0;

private int childFormNumber = 0;

public bool fInicio=true, fMesseger = true, fContactos = true;

private void toolsMenu\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void MDIParent1\_Load(object sender, EventArgs e)

{

}

private void fileMenu\_Click(object sender, EventArgs e)

{

NewsFeed childForm = new NewsFeed();

childForm.MdiParent = this;

childForm.Text = "NewsFeed";

childForm.Show();

}

private void editMenu\_Click(object sender, EventArgs e)

{

Messeger childForm = new Messeger();

childForm.MdiParent = this;

childForm.Text = "Messeger";

childForm.Show();

fMesseger = false;

}

private void viewMenu\_Click(object sender, EventArgs e)

{

Contactos childForm = new Contactos();

childForm.MdiParent = this;

childForm.Text = "Contactos";

childForm.Show();

}

public Home()

{

InitializeComponent();

}

private void ShowNewForm(object sender, EventArgs e)

{

Form childForm = new Form();

childForm.MdiParent = this;

childForm.Text = "Ventana " + childFormNumber++;

childForm.Show();

}

private void OpenFile(object sender, EventArgs e)

{

OpenFileDialog openFileDialog = new OpenFileDialog();

openFileDialog.InitialDirectory = Environment.GetFolderPath(Environment.SpecialFolder.Personal);

openFileDialog.Filter = "Archivos de texto (\*.txt)|\*.txt|Todos los archivos (\*.\*)|\*.\*";

if (openFileDialog.ShowDialog(this) == DialogResult.OK)

{

string FileName = openFileDialog.FileName;

}

}

private void SaveAsToolStripMenuItem\_Click(object sender, EventArgs e)

{

SaveFileDialog saveFileDialog = new SaveFileDialog();

saveFileDialog.InitialDirectory = Environment.GetFolderPath(Environment.SpecialFolder.Personal);

saveFileDialog.Filter = "Archivos de texto (\*.txt)|\*.txt|Todos los archivos (\*.\*)|\*.\*";

if (saveFileDialog.ShowDialog(this) == DialogResult.OK)

{

string FileName = saveFileDialog.FileName;

}

}

private void CascadeToolStripMenuItem\_Click(object sender, EventArgs e)

{

LayoutMdi(MdiLayout.Cascade);

}

private void TileVerticalToolStripMenuItem\_Click(object sender, EventArgs e)

{

LayoutMdi(MdiLayout.TileVertical);

}

private void TileHorizontalToolStripMenuItem\_Click(object sender, EventArgs e)

{

LayoutMdi(MdiLayout.TileHorizontal);

}

private void ArrangeIconsToolStripMenuItem\_Click(object sender, EventArgs e)

{

LayoutMdi(MdiLayout.ArrangeIcons);

}

private void CloseAllToolStripMenuItem\_Click(object sender, EventArgs e)

{

foreach (Form childForm in MdiChildren)

{

childForm.Close();

}

}

private void saveToolStripMenuItem\_Click(object sender, EventArgs e)

{

}

}

}

*CONTACTOS*

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.IO;

namespace Proyecto\_Facebook

{

public partial class Contactos : Form

{

public Contactos()

{

InitializeComponent();

}

private void Contactos\_Load(object sender, EventArgs e)

{

StreamReader contactos = new StreamReader("contacts.txt", true);

string leerlineas = contactos.ReadLine();//almecena linea

string[] campos; //almecenar campos

string identificador = "";

cListaEnlazada lista = new cListaEnlazada();

while (leerlineas != null)

{

cNodo auxrecorre = new cNodo();

campos = leerlineas.Split('$');//separando

//enviar a nodo para insertar

lista.pInserta(campos);

if (campos[0] != null)

{

identificador = campos[0];

comboBox1.Items.Add(identificador);

}

auxrecorre = auxrecorre.cEnlace;

leerlineas = contactos.ReadLine();

}

contactos.Close();

}

private void comboBox1\_Click(object sender, EventArgs e)

{

}

private void comboBox1\_SelectionChangeCommitted(object sender, EventArgs e)

{

imprimir();

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

imprimir();

}

void imprimir()

{

bool flag = true;

StreamReader scontactos = new StreamReader("contacts.txt", true);

string identificador = comboBox1.Text;

string leer = scontactos.ReadLine();//almecena linea

string[] datos;

//buscar

while (leer != null && flag == true)

{

datos = leer.Split('$');

if (identificador == datos[0])

{

label7.Text = datos[0];

label8.Text = datos[1];

label9.Text = datos[2];

label10.Text = datos[3];

label11.Text = datos[4];

if (datos[5] == "0")

{

label12.BackColor = Color.DarkGreen;

label12.ForeColor = Color.White;

label12.Text = ":D";

flag = false;

}

if (datos[5] == "1")

{

label12.BackColor = Color.DarkRed;

label12.ForeColor = Color.White;

label12.Text = ">:v";

flag = false;

}

if (datos[5] == "2")

{

label12.BackColor = Color.Yellow;

label12.ForeColor = Color.Black;

label12.Text = "♦";

flag = false;

}

}

leer = scontactos.ReadLine();

}

}

void destacar()

{

bool flag = true;

StreamReader scontactos = new StreamReader("contacts.txt", true);

string identificador = comboBox1.Text;

string leer = scontactos.ReadLine();//almecena linea

string[] datos;

//buscar

while (leer != null && flag == true)

{

datos = leer.Split('$');

if (identificador == datos[0])

{

if (datos[5] == "0")

{

datos[5] = "2";

label12.BackColor = Color.Yellow;

label12.ForeColor = Color.Black;

label12.Text = "♦";

MessageBox.Show("Destacado Con Exito");

scontactos.Close();

StreamWriter escribir = new StreamWriter("contacts.txt", true);

escribir.WriteLine(datos[0] + "$" + datos[1] + "$" + datos[2] + "$" + datos[3] + "$" + datos[4] + "$" + datos[5]);

flag = false;

}

if (datos[5] == "1")

{

label12.BackColor = Color.DarkRed;

label12.ForeColor = Color.White;

label12.Text = ">:v";

MessageBox.Show("Desbloquar primero");

flag = false;

}

if (datos[5] == "2")

{

datos[5] = "2";

label12.BackColor = Color.Yellow;

label12.ForeColor = Color.Black;

label12.Text = "♦";

MessageBox.Show("Ya esta destacado");

flag = false;

flag = false;

}

}

leer = scontactos.ReadLine();

}

}

void desbloquear()

{

bool flag = true;

StreamReader scontactos = new StreamReader("contacts.txt", true);

string identificador = comboBox1.Text;

string leer = scontactos.ReadLine();//almecena linea

string[] datos;

//buscar

while (leer != null && flag == true)

{

datos = leer.Split('$');

if (identificador == datos[0])

{

if (datos[5] == "0")

{

datos[5] = "1";

label12.BackColor = Color.DarkGreen;

label12.ForeColor = Color.White;

label12.Text = ":D";

MessageBox.Show("El usuario no esta bloqueado");

flag = false;

}

if (datos[5] == "1")

{

label12.BackColor = Color.DarkGreen;

label12.ForeColor = Color.White;

label12.Text = ":D";

MessageBox.Show("Usuario Desbloqueado");

flag = false;

}

if (datos[5] == "2")

{

MessageBox.Show("El usuario no esta bloqueado");

flag = false;

}

}

leer = scontactos.ReadLine();

}

}

private void button1\_Click(object sender, EventArgs e)

{

}

void bloquear()

{

bool flag = true;

StreamReader scontactos = new StreamReader("contacts.txt", true);

string identificador = comboBox1.Text;

string leer = scontactos.ReadLine();//almecena linea

string[] datos;

//buscar

while (leer != null && flag == true)

{

datos = leer.Split('$');

if (identificador == datos[0])

{

if (datos[5] == "0")

{

datos[5] = "1";

label12.BackColor = Color.DarkRed;

label12.ForeColor = Color.White;

label12.Text = ">:v";

MessageBox.Show("Bloqueado Con Exito");

flag = false;

}

if (datos[5] == "1")

{

label12.BackColor = Color.DarkRed;

label12.ForeColor = Color.White;

label12.Text = ">:v";

MessageBox.Show("Ya esta bloqueado");

flag = false;

}

if (datos[5] == "2")

{

datos[5] = "1";

label12.BackColor = Color.DarkRed;

label12.ForeColor = Color.White;

label12.Text = ">:v";

MessageBox.Show("Bloqueado Con Exito");

flag = false;

flag = false;

}

}

leer = scontactos.ReadLine();

}

}

private void button2\_Click(object sender, EventArgs e)

{

bloquear();

}

private void button3\_Click(object sender, EventArgs e)

{

desbloquear();

}

private void button4\_Click(object sender, EventArgs e)

{

destacar();

}

}

}

*MESSENGER*

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.IO;

namespace Proyecto\_Facebook

{

public partial class Messeger : Form

{

public Messeger()

{

InitializeComponent();

}

private void Messeger\_Load(object sender, EventArgs e)

{

StreamReader mensajes = new StreamReader("messenger.txt", true);

listBox4.Text = "";

listBox5.Text = "";

string leerlineas = mensajes.ReadLine();//almecena linea

string[] campos; //almecenar campos

string anterior = "";

cCola pila = new cCola();

while (leerlineas != null)

{

dNodo auxrecorre = new dNodo();

campos = leerlineas.Split('$');//separando

//enviar a nodo para insertar

pila.cPush2(campos);

if (campos[0] != null)

{

if (anterior != campos[0])

{

listBox1.Items.Add(campos[0]);

listBox2.Items.Add(campos[2]);

if (campos[3] == "1")

{

listBox3.Items.Add("♦");

}

else

{

listBox3.Items.Add("-");

}

anterior = campos[0];

}

//if (campos[3] == "0")

//{

// Label estado = new Label();

// estado.Top = y;

// estado.Left = 340;

// estado.Width = 46;

// estado.Height = 17;

// y = y + 17;

// estado.ForeColor = Color.Black;

// estado.BackColor = Color.Cyan;

// estado.Text = "♦";

//}

//else

//{

// Label estado = new Label();

// estado.Top = y;

// estado.Left = 340;

// estado.Width = 46;

// estado.Height = 17;

// y = y + 17;

// estado.ForeColor = Color.DarkBlue;

// estado.Text = "♦";

//}

}

auxrecorre = auxrecorre.cEnlace;

leerlineas = mensajes.ReadLine();

}

mensajes.Close();

}

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

listBox4.Items.Clear();

listBox5.Items.Clear();

listBox4.Text = "";

listBox4.Text = "";

string actual = Convert.ToString(listBox1.SelectedItem);

label2.Text = actual;

StreamReader lol = new StreamReader("messenger.txt", true);

string leerlineas = lol.ReadLine();//almecena linea

string[] lel; //almecenar campos

while (leerlineas != null)

{

lel = leerlineas.Split('$');

if ((lel[4] == "yo" || lel[4] == "YO" || lel[4] == "Yo" || lel[4] == "yO") && actual == lel[0])

{

listBox5.Items.Add(lel[1]);

}

else if (actual == lel[0])

{

listBox4.Items.Add(lel[1]);

}

leerlineas = lol.ReadLine();

}

}

private void listBox5\_SelectedIndexChanged(object sender, EventArgs e)

{

}

}

}

*NEWSFEED*

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.IO;

namespace Proyecto\_Facebook

{

public partial class NewsFeed : Form

{

Home home = new Home();

cCola lista = new cCola();

int x = 10;

int y = 30;

int contador = 1;

int panelalto = 426;

public NewsFeed()

{

InitializeComponent();

}

private void NewsFeed\_Load(object sender, EventArgs e)

{

StreamReader leer = new StreamReader("newsfeed.txt", true);

string leerlineas = leer.ReadLine();//almecena linea

string[] campos; //almecenar campos

while (leerlineas != null)

{

cNodo auxrecorre = new cNodo();

campos = leerlineas.Split('$');//separando

//enviar a nodo para insertar

lista.cPush(campos);

if (campos[3] != null)

{

Label correo = new Label();

Label tipo = new Label();

PictureBox foto = new PictureBox();

RichTextBox txt = new RichTextBox();

//correo

correo.Height = 15;

correo.Width = 100;

correo.Left = x;

correo.Top = y;

correo.Text = campos[0];

panel1.Controls.Add(correo);

//tipo

tipo.Height = 15;

tipo.Width = 50;

tipo.Text = campos[1] + ":";

tipo.Left = x;

tipo.Top = y + 20;

panel1.Controls.Add(tipo);

//imagen

if (campos[1] == "Historia" || campos[1] == "historia" || campos[1] == "HISTORIA")

{

PictureBox imagen = new PictureBox();

//string ruta;

imagen.Width = 200;

imagen.Height = 100;

imagen.Top = y;

imagen.Left = x + 310;

imagen.Enabled = false;

imagen.BackColor = Color.Red;

imagen.Name = "imagen" + contador;

imagen.SizeMode = PictureBoxSizeMode.StretchImage;

imagen.ImageLocation = "imagen" + contador + ".jpg";

panel1.Controls.Add(imagen);

//imagen.Image = ";

}

//texto

txt.Text = campos[3];

txt.Width = 200;

txt.Height = 100;

txt.Top = y;

txt.Left = x + 110;

txt.Enabled = false;

panel1.Controls.Add(txt);

txt.Name = "Txt" + contador;

contador++;

y = y + 105;

}

auxrecorre = auxrecorre.cEnlace;

leerlineas = leer.ReadLine();

panel1.Height = panelalto + 100;

}

}

private void NewsFeed\_FormClosing(object sender, FormClosingEventArgs e)

{

home.fInicio = false;

}

private void NewsFeed\_FormClosed(object sender, FormClosedEventArgs e)

{

home.fInicio = false;

}

private void button1\_Click(object sender, EventArgs e)

{

}

private void vScrollBar1\_Scroll(object sender, ScrollEventArgs e)

{

//panely = vScrollBar1.Value;

//panely = panel1.Location.Y;

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

}

}

# Conclusiones

1. Se lograron satisfacer las instrucciones necesarias para la red social.
2. Todo fue programado a base de nuestros conocimientos aprendidos en clase tanto en clase teórica como práctica.
3. Se determino la correcta funcionalidad de la librería System.IO; y sus objetos StreamReader.
4. Se lograron cumplir con las condiciones dadas.
5. Se logró mostrar y diferenciar entres muchos parámetros de una red social, como lo chats y los Newsfeed.

# Bibliografía

Hernandez, M. (s.f.). *Curso de Programación orientada a objetos*. Obtenido de https://codigofacilito.com/cursos/c\_sharp

Rosa, V. d. (24 de Marzo de 2012). *Foros de Programadores*. Obtenido de https://es.ccm.net/forum/affich-1145256-como-programar-el-boton-imprimir-en-c-2010

Tuttini, L. (26 de Julio de 2011). *Mover un PictureBox*. Obtenido de https://social.msdn.microsoft.com/Forums/es-ES/576c134d-7e74-4b06-b0a5-17c8a6106ac4/mover-un-picturebox-en-el-eje-y?forum=vcses