CÓDIGO CLIENTE

using System;

using System.Net.Sockets;

using System.Text;

using System.Threading;

using System.Windows.Forms;

namespace Cliente

{

/// <summary>

/// Aplicacion cliente de chat

/// </summary>

public partial class frmCliente : Form

{

#region [ Variables ]

/// <summary>

/// La clase TcpClient proporciona métodos sencillos para conectar, enviar y recibir flujos de datos a través de una red en modo de bloqueo sincrónico.

/// </summary>

TcpClient cliente;

/// <summary>

/// La clase NetworkStream proporciona métodos para enviar y recibir datos a través de sockets de Stream en modo de bloqueo

/// </summary>

NetworkStream streamServidor;

/// <summary>

/// Mensaje que se envia a la pantalla del chat

/// </summary>

string mensajeChat;

/// <summary>

/// Variable que almacena el id del cliente

/// </summary>

public string idNombre;

public string sala;

#endregion

#region [ Constructor ]

public frmCliente()

{

InitializeComponent();

}

#endregion

#region [ Eventos ]

/// <summary>

/// Conecta al cliente con el servidor de chat

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

idNombre = txtNombre.Text;

sala = txtSala.Text;

metodoConectar();

}

/// <summary>

/// Envía un mensaje al servidor

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

try

{

btnConectar.Enabled = true;

//Incializamos el NetworkStream

streamServidor = cliente.GetStream();

//si el usuario esta duplicado actualio el nombre

txtNombre.Text = idNombre ;

//Transformamos el string en un arreglo de bytes para poder ser enviado mediante el NetworkStream

Byte[] datos = Encoding.ASCII.GetBytes(txtMensaje.Text + "$" + sala + " " + txtNombre.Text);

//Enviamos los datos al servidor

streamServidor.Write(datos, 0, datos.Length);

streamServidor.Flush();

txtMensaje.Text = "";

txtMensaje.Focus();

}

catch (Exception ex)

{

txtChat.Text = ex.ToString();

}

}

/// <summary>

/// Desconecta nuestra sesion.

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

if (cliente != null && cliente.Connected)

{

cliente.Close();

btnConectar.Enabled = true;

btnDesconectar.Enabled = false;

btnEnviar.Enabled = false;

this.Hide();

}

}

#endregion

#region [ Metodos ]

/// <summary>

/// Cicla indefinidamente el proceso de espera de mensajes por parte del servidor

/// </summary>

private void Chat()

{

try

{

while (true)

{

streamServidor = cliente.GetStream();

//int buffSize = 0;

byte[] bytes = new byte[256];

//Leemos el mensaje enviado por el servidor

streamServidor.Read(bytes, 0, bytes.Length);

//Y lo enviamos a la pantalla del chat

mensajeChat = Encoding.ASCII.GetString(bytes);

if (mensajeChat.Substring(0,21) == "/\*usuario duplicado\*/")

{

idNombre = txtNombre.Text +"1";

MessageBox.Show("El usuario que ingresó ya existe" );

}

if (mensajeChat.Substring(0,sala.Length)==sala)

{

Mensaje();

}

}

}catch(Exception e)

{}

}

/// <summary>

/// Envia el mensaje al TextBox del Chat

/// </summary>

private void Mensaje()

{

if (this.InvokeRequired)

this.Invoke(new MethodInvoker(Mensaje));

else

txtChat.Text = txtChat.Text + Environment.NewLine + " -> " + " Sala: " + mensajeChat;

}

/// <sumary>

/// metod0 de conexion con el servidor

/// </sumary>

private void metodoConectar()

{

try

{

this.Text = string.Format("Sala: {0} Cliente: {1}",txtSala.Text, txtNombre.Text);

}

catch

{ }

try

{

mensajeChat = "Solicitando conexión al servidor";

Mensaje();

//Abrimos la conexion con el servidor

cliente = new TcpClient(txtIP.Text, int.Parse(txtPuerto.Text));

//Incializamos el stream

streamServidor = cliente.GetStream();

//Transformamos el string en un arreglo de bytes para poder ser enviado mediante el NetworkStream

Byte[] datos = System.Text.Encoding.ASCII.GetBytes(sala +" "+ txtNombre.Text + "$");

//Enviamos los datos al servidor

streamServidor.Write(datos, 0, datos.Length);

streamServidor.Flush();

//Ciclamos el proceso de escuchar respuestas del servidor, en esta caso cada que un cliente envía un mensaje.

Thread ctThread = new Thread(Chat);

ctThread.Start();

btnConectar.Enabled = false;

btnDesconectar.Enabled = true;

btnEnviar.Enabled = true;

}

catch (Exception ex)

{

if (MessageBox.Show("Ha ocurrido un error intente de nuevo ") == System.Windows.Forms.DialogResult.Yes)

btnConectar\_Click(null, null);

else

this.Close();

}

}

#endregion

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

{

try

{

if (streamServidor != null)

{

Byte[] datos = Encoding.ASCII.GetBytes("/\*Listar Usuarios\*/" + "$" + sala + " " + txtNombre.Text);

//Enviamos los datos al servidor

streamServidor.Write(datos, 0, datos.Length);

streamServidor.Flush();

}

}

catch(Exception ex)

{}

}

}

}