



INFORME DE GUÍA PRÁCTICA

I. PORTADA

Tema: Informe de Práctica: Comunicación de Aplicaciones por Sockets.
Unidad de Organización Curricular: BÁSICA
Nivel y Paralelo: Tercero - A
Alumnos participantes: Raza Guerrero Benjamín Alejandro
Asignatura: Programación Avanzada
Docente: Ing. Rubén Caiza, Mg.

II. INFORME DE GUÍA PRÁCTICA

1.1 Objetivo General:

Desarrollar una aplicación de comunicación entre dos equipos mediante sockets TCP en un entorno gráfico Windows Forms, incorporando mecanismos de cifrado para garantizar la seguridad en la transmisión de mensajes.

Específicos:

- Establecer una conexión punto a punto mediante sockets TCP para el envío de mensajes.
- Aplicar técnicas de encriptación y desencriptación (AES) para asegurar la confidencialidad de los datos.
- Diseñar una interfaz gráfica intuitiva que facilite la interacción entre el usuario y la aplicación.
- Validar el correcto funcionamiento de la aplicación en un entorno de red local simulado.

1.2 Modalidad

Presencial.

1.3 Tiempo de duración Presenciales: 6

No presenciales: 0

1.4 Instrucciones

En esta práctica se implementó una solución de comunicación entre aplicaciones utilizando el lenguaje C# y el entorno Visual Studio 2022. Se desarrolló una interfaz gráfica con Windows Forms que permite a los usuarios interactuar con el sistema de envío y recepción de mensajes, garantizando la confidencialidad de los datos mediante el uso de cifrado AES. La prueba se realizó en un entorno local simulando dos equipos, utilizando IP 127.0.0.2 para diferenciar las instancias.

1.5 Listado de equipos, materiales y recursos

TAC (Tecnologías para el Aprendizaje y Conocimiento) empleados en la guía práctica:

- ☒ Plataformas educativas
- ☐ Simuladores y laboratorios virtuales
- ☒ Aplicaciones educativas
- ☒ Recursos audiovisuales
- ☐ Gamificación



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
CICLO ACADÉMICO: MARZO – JULIO 2025



☒ Inteligencia Artificial

Otros (Especifique): _____

1.6 Actividades desarrolladas

Durante el desarrollo de la práctica se utilizó Visual Studio 2022 Community Edition para crear una aplicación Windows Forms en C#. Se diseñó un formulario que permite introducir la IP (127.0.0.2), el puerto, el mensaje a enviar y que cuenta con botones para iniciar el servidor, conectarse como cliente y enviar mensajes.

En la codificación se utilizaron las clases TcpListener y TcpClient del espacio de nombres System.Net.Sockets para establecer la conexión, y System.Security.Cryptography para realizar el cifrado AES de los mensajes. Se emplearon hilos (Thread) para gestionar la comunicación asincrónica entre cliente y servidor, permitiendo que ambas instancias funcionen simultáneamente.

La prueba fue ejecutada en un solo equipo simulando ambos roles (cliente y servidor), garantizando la transmisión segura y la visualización correcta de los mensajes mediante un área de log.

1.7 Resultados obtenidos

La aplicación funcionó correctamente, permitiendo establecer comunicación entre dos instancias locales utilizando la IP 127.0.0.2. Los mensajes se enviaron y recibieron con éxito, mostrando en el log tanto el texto enviado como el recibido, ya descifrado.

Se evidenció que la implementación del cifrado AES cumplía su propósito, ya que los datos en tránsito no podían leerse directamente. También se verificó la estabilidad de la conexión, la recepción continua de mensajes y la correcta separación de roles entre servidor y cliente. La experiencia permitió comprender la práctica real de la programación de sockets y la integración de seguridad en aplicaciones de escritorio.

1.7.1 Form1.cs

Descripción: El archivo Form1.cs contiene la lógica principal de la aplicación. Aquí se define el comportamiento de los botones para iniciar el servidor, conectarse como cliente y enviar mensajes. También se manejan las funciones de encriptación y descifrado de mensajes usando AES, así como la gestión de hilos para la recepción continua de datos.

a. Diseño del Formulario

El diseño incluye campos de entrada para la IP y el puerto, una caja de texto para el mensaje, un área de log para mostrar la conversación, y botones funcionales para conectarse, iniciar el servidor y enviar mensajes. La interfaz fue desarrollada con un enfoque práctico, permitiendo ejecutar rápidamente las funciones deseadas desde un mismo formulario.

b. Codificación del Código

```
using System;  
using System.Net.Sockets;  
using System.Net;  
using System.Text;  
using System.Security.Cryptography;  
using System.Windows.Forms;  
using System.Threading;
```



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
CICLO ACADÉMICO: MARZO – JULIO 2025



```
namespace ChatTcpEncriptado
{
    public partial class Form1 : Form
    {
        TcpListener listener;
        TcpClient client;
        NetworkStream stream;
        Thread recibirThread;
        static string claveSecreta = "claveSegura16bts";

        public Form1()
        {
            InitializeComponent();
        }

        private void btnIniciarServidor_Click(object sender, EventArgs e)
        {
            int puerto = int.Parse(txtPuerto.Text);
            listener = new TcpListener(IPAddress.Any, puerto);
            listener.Start();
            txtLog.AppendText("Servidor iniciado en puerto " + puerto + "\r\n");

            Thread serverThread = new Thread(() =>
            {
                client = listener.AcceptTcpClient();
                stream = client.GetStream();
                this.Invoke((MethodInvoker)() => txtLog.AppendText("Cliente
conectado.\r\n"));

                byte[] buffer = new byte[1024];
                int bytesLeidos;
                while ((bytesLeidos = stream.Read(buffer, 0, buffer.Length)) > 0)
                {
                    string mensajeEncriptado = Encoding.UTF8.GetString(buffer, 0, bytesLeidos);
                    string mensaje = Desencriptar(mensajeEncriptado);
                    this.Invoke((MethodInvoker)() => txtLog.AppendText("Recibido: " + mensaje
+ "\r\n"));
                }
            });
            serverThread.IsBackground = true;
            serverThread.Start();
        }

        private void btnConectar_Click(object sender, EventArgs e)
        {
            try
            {
                client = new TcpClient();
                client.Connect(txtIP.Text, int.Parse(txtPuerto.Text));
                stream = client.GetStream();
                txtLog.AppendText("Conectado al servidor.\r\n");
            }
            catch { }
        }
    }
}
```



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
CICLO ACADÉMICO: MARZO – JULIO 2025



```
recibirThread = new Thread(() =>
{
    byte[] buffer = new byte[1024];
    int bytesLeidos;
    while ((bytesLeidos = stream.Read(buffer, 0, buffer.Length)) > 0)
    {
        string mensajeEncriptado = Encoding.UTF8.GetString(buffer, 0,
bytesLeidos);
        string mensaje = Desencriptar(mensajeEncriptado);
        this.Invoke((MethodInvoker)() => txtLog.AppendText("Recibido: " +
mensaje + "\r\n"));
    }
});
recibirThread.IsBackground = true;
recibirThread.Start();
}
catch (Exception ex)
{
    MessageBox.Show("Error al enviar: " + ex.Message);
}
}

private void btnEnviar_Click(object sender, EventArgs e)
{
    if (stream != null && client.Connected && stream.CanWrite)
    {
        string mensaje = txtMensaje.Text;
        string mensajeEncriptado = Encriptar(mensaje);
        byte[] datos = Encoding.UTF8.GetBytes(mensajeEncriptado);
        stream.Write(datos, 0, datos.Length);
        txtLog.AppendText("Enviado: " + mensaje + "\r\n");
        txtMensaje.Clear();
    }
}

private string Encriptar(string texto)
{
    using (Aes aes = Aes.Create())
    {
        byte[] clave = Encoding.UTF8.GetBytes(claveSecreta);

        // Asegura que la clave tenga 16 bytes (AES-128)
        if (clave.Length != 16)
            throw new ArgumentException("La clave debe tener exactamente 16 caracteres
para AES-128.");

        aes.Key = clave;
        aes.IV = new byte[16]; // Vector de inicialización fijo (puedes randomizar si
deseas más seguridad)

        ICryptoTransform encryptor = aes.CreateEncryptor();
        byte[] inputBuffer = Encoding.UTF8.GetBytes(texto);
```



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
CICLO ACADÉMICO: MARZO – JULIO 2025



```
byte[] output = encryptor.TransformFinalBlock(inputBuffer, 0,
inputBuffer.Length);
return Convert.ToBase64String(output);
}
}

private string Desencriptar(string textoEncriptado)
{
    using (Aes aes = Aes.Create())
    {
        aes.Key = Encoding.UTF8.GetBytes(claveSecreta);
        aes.IV = new byte[16];
        ICryptoTransform decryptor = aes.CreateDecryptor(aes.Key, aes.IV);
        byte[] inputBuffer = Convert.FromBase64String(textoEncriptado);
        byte[] output = decryptor.TransformFinalBlock(inputBuffer, 0,
inputBuffer.Length);
        return Encoding.UTF8.GetString(output);
    }
}

[STAThread]
static void Main()
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);

    // Lanzar dos ventanas automáticamente
    Thread t1 = new Thread(() => Application.Run(new Form1()));
    Thread t2 = new Thread(() => Application.Run(new Form1()));
    t1.SetApartmentState(ApartmentState.STA);
    t2.SetApartmentState(ApartmentState.STA);
    t1.Start();
    t2.Start();
}
}
```

1.7.2 Form1.Designer.cs

Descripción: El archivo Form1.Designer.cs define visualmente los componentes del formulario en tiempo de diseño. Es donde se configuran las propiedades gráficas, como la ubicación y tamaño de los controles (TextBox, Button, etc.) y sus textos por defecto, como la IP inicial 127.0.0.2.

a. Diseño del Formulario

Se organizó una interfaz sencilla y funcional. A la izquierda se ubican los campos de entrada (IP, puerto), y al centro un área de texto para los mensajes. En la parte inferior, un botón permite el envío, mientras que otros dos botones permiten iniciar el servidor o conectarse como cliente. El TextBox de log ocupa una posición destacada para monitorear toda la conversación cifrada.

b. Codificación del código

```
namespace ChatTcpEncriptado
```



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
CICLO ACADÉMICO: MARZO – JULIO 2025



```
{
partial class Form1
{
    private System.ComponentModel.IContainer components = null;
    private System.Windows.Forms.TextBox txtIP;
    private System.Windows.Forms.TextBox txtPuerto;
    private System.Windows.Forms.TextBox txtMensaje;
    private System.Windows.Forms.Button btnConectar;
    private System.Windows.Forms.Button btnIniciarServidor;
    private System.Windows.Forms.Button btnEnviar;
    private System.Windows.Forms.TextBox txtLog;

    protected override void Dispose(bool disposing)
    {
        if (disposing && (components != null))
        {
            components.Dispose();
        }
        base.Dispose(disposing);
    }

    private void InitializeComponent()
    {
        this.txtIP = new System.Windows.Forms.TextBox();
        this.txtPuerto = new System.Windows.Forms.TextBox();
        this.txtMensaje = new System.Windows.Forms.TextBox();
        this.btnConectar = new System.Windows.Forms.Button();
        this.btnIniciarServidor = new System.Windows.Forms.Button();
        this.btnEnviar = new System.Windows.Forms.Button();
        this.txtLog = new System.Windows.Forms.TextBox();
        this.SuspendLayout();
        //
        // txtIP
        //
        this.txtIP.Location = new System.Drawing.Point(12, 12);
        this.txtIP.Name = "txtIP";
        this.txtIP.Size = new System.Drawing.Size(150, 20);
        this.txtIP.TabIndex = 0;
        this.txtIP.Text = "127.0.0.1";
        //
        // txtPuerto
        //
        this.txtPuerto.Location = new System.Drawing.Point(168, 12);
        this.txtPuerto.Name = "txtPuerto";
        this.txtPuerto.Size = new System.Drawing.Size(100, 20);
        this.txtPuerto.TabIndex = 1;
        this.txtPuerto.Text = "5000";
        //
        // txtMensaje
        //
        this.txtMensaje.Location = new System.Drawing.Point(12, 240);
        this.txtMensaje.Name = "txtMensaje";
        this.txtMensaje.Size = new System.Drawing.Size(350, 20);
    }
}
```



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
CICLO ACADÉMICO: MARZO – JULIO 2025



```
this.txtMensaje.TabIndex = 2;
//
// btnConectar
//
this.btnConectar.Location = new System.Drawing.Point(274, 10);
this.btnConectar.Name = "btnConectar";
this.btnConectar.Size = new System.Drawing.Size(88, 23);
this.btnConectar.TabIndex = 3;
this.btnConectar.Text = "Conectar";
this.btnConectar.UseVisualStyleBackColor = true;
this.btnConectar.Click += new System.EventHandler(this.btnConectar_Click);
//
// btnIniciarServidor
//
this.btnIniciarServidor.Location = new System.Drawing.Point(368, 10);
this.btnIniciarServidor.Name = "btnIniciarServidor";
this.btnIniciarServidor.Size = new System.Drawing.Size(110, 23);
this.btnIniciarServidor.TabIndex = 4;
this.btnIniciarServidor.Text = "Iniciar Servidor";
this.btnIniciarServidor.UseVisualStyleBackColor = true;
this.btnIniciarServidor.Click += new
System.EventHandler(this.btnIniciarServidor_Click);
//
// btnEnviar
//
this.btnEnviar.Location = new System.Drawing.Point(368, 237);
this.btnEnviar.Name = "btnEnviar";
this.btnEnviar.Size = new System.Drawing.Size(110, 23);
this.btnEnviar.TabIndex = 5;
this.btnEnviar.Text = "Enviar";
this.btnEnviar.UseVisualStyleBackColor = true;
this.btnEnviar.Click += new System.EventHandler(this.btnEnviar_Click);
//
// txtLog
//
this.txtLog.Location = new System.Drawing.Point(12, 40);
this.txtLog.Multiline = true;
this.txtLog.Name = "txtLog";
this.txtLog.ScrollBars = System.Windows.Forms.ScrollBars.Vertical;
this.txtLog.Size = new System.Drawing.Size(466, 190);
this.txtLog.TabIndex = 6;
//
// Form1
//
this.ClientSize = new System.Drawing.Size(490, 270);
this.Controls.Add(this.txtLog);
this.Controls.Add(this.btnEnviar);
this.Controls.Add(this.btnIniciarServidor);
this.Controls.Add(this.btnConectar);
this.Controls.Add(this.txtMensaje);
this.Controls.Add(this.txtPuerto);
this.Controls.Add(this.txtIP);
this.Name = "Form1";
```




UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
CICLO ACADÉMICO: MARZO – JULIO 2025



```
this.Text = "Chat TCP Encriptado";  
this.ResumeLayout(false);  
this.PerformLayout();  
}  
}  
}
```

1.8 Habilidades blandas empleadas en la práctica

- ☒ Liderazgo
- ☒ Trabajo en equipo
- ☒ Comunicación asertiva
- ☒ La empatía
- ☒ Pensamiento crítico
- ☒ Flexibilidad
- ☒ La resolución de conflictos
- ☒ Adaptabilidad
- ☒ Responsabilidad

1.9 Conclusiones

La presente práctica permitió adquirir experiencia directa en el desarrollo de aplicaciones que utilizan sockets TCP para la comunicación entre equipos. Mediante la implementación de una interfaz gráfica en Windows Forms, fue posible observar cómo los conceptos de programación avanzada pueden aplicarse en soluciones reales orientadas a la transmisión de datos.

Uno de los principales logros fue integrar el cifrado AES en la capa de transporte de mensajes, lo cual refuerza la importancia de considerar la seguridad como parte integral del ciclo de desarrollo de software. La correcta encriptación y desencriptación de los mensajes evidenció que los datos pueden ser protegidos de manera eficiente sin comprometer la funcionalidad de la aplicación.

Además, se logró entender el funcionamiento interno del modelo cliente-servidor mediante el uso de hilos para mantener la comunicación asíncrona. Esto demostró cómo se pueden gestionar múltiples procesos simultáneamente sin bloquear la interfaz de usuario, un conocimiento fundamental para el diseño de sistemas más complejos y escalables.

También se destacó la relevancia de trabajar con pruebas en un entorno local controlado, lo cual facilita la depuración y permite validar el correcto funcionamiento antes de llevar el sistema a un entorno de red real. La práctica fomentó el desarrollo de habilidades tanto técnicas como blandas, como la resolución de problemas, pensamiento crítico, trabajo en equipo y adaptación a nuevas herramientas.

En resumen, la actividad fortaleció el conocimiento en programación avanzada, específicamente en lo relacionado a la comunicación de red, la seguridad de la información y la estructuración de aplicaciones de escritorio robustas. Este tipo de prácticas sienta una base sólida para afrontar proyectos más ambiciosos en el ámbito profesional.

1.10 Recomendaciones

- Realizar pruebas en entornos reales con diferentes dispositivos para validar la robustez de la aplicación fuera del entorno local.



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
CICLO ACADÉMICO: MARZO – JULIO 2025



- Añadir mecanismos de autenticación para aumentar la seguridad en el intercambio de mensajes.
- Optimizar la gestión de excepciones y errores de conexión para evitar cierres inesperados.
- Documentar exhaustivamente el código con comentarios que faciliten el mantenimiento.
- Implementar mejoras visuales en la interfaz para una experiencia de usuario más amigable.
- Versionar el proyecto en plataformas como GitHub para mantener un control del desarrollo y facilitar la colaboración en equipo.

1.11 Referencias bibliográficas

[1] Microsoft, “TcpListener Class,” *Microsoft Learn*, [Online]. Available: <https://learn.microsoft.com/en-us/dotnet/api/system.net.sockets.tcplistener>

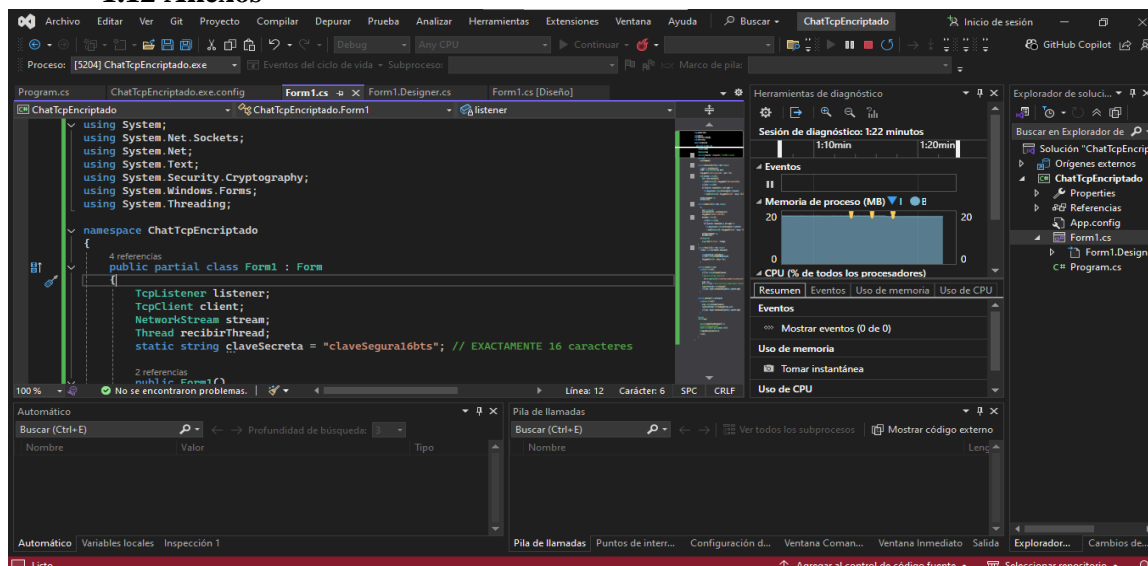
[2] Microsoft, “TcpClient Class,” *Microsoft Learn*, [Online]. Available: <https://learn.microsoft.com/en-us/dotnet/api/system.net.sockets.tcpclient>

[3] Microsoft, “Aes Class,” *Microsoft Learn*, [Online]. Available: <https://learn.microsoft.com/en-us/dotnet/api/system.security.cryptography.aes>

[4] J. Richter, *CLR via C#*, 4th ed., Microsoft Press, 2012.

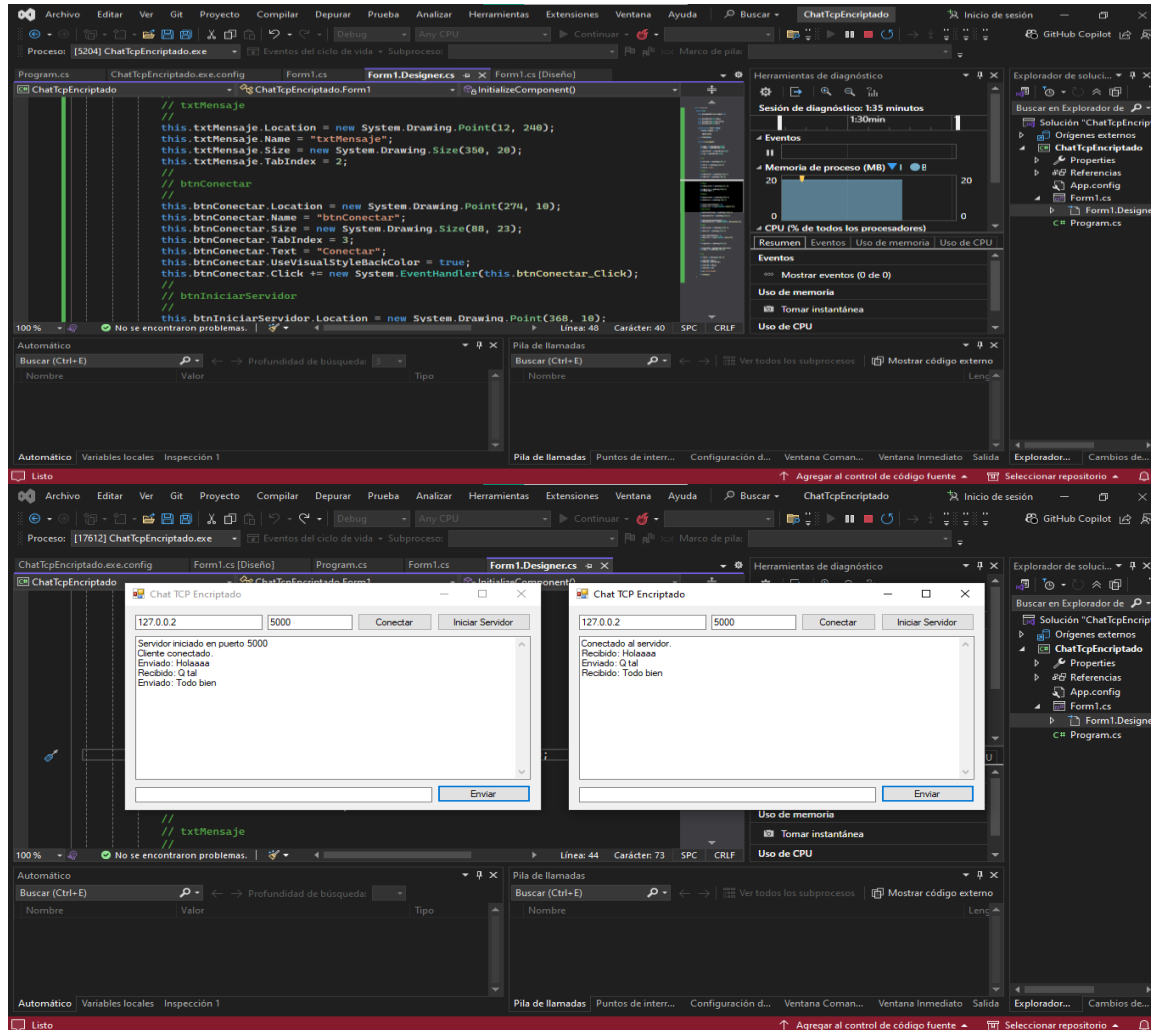
[5] A. Troelsen and P. Japikse, *Pro C# 10 with .NET 6*, 11th ed., Apress, 2022.

1.12 Anexos

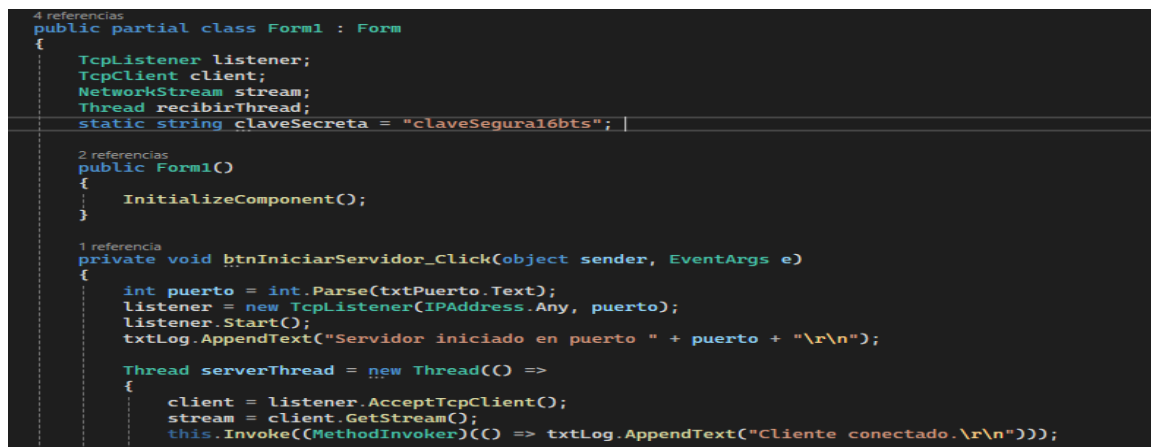




UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
CICLO ACADÉMICO: MARZO – JULIO 2025



Parte de Sockets:



LINK GITHUB:

<https://github.com/Grias-emory/SG---APE-5.git>