

# Universidad Mariano Gálvez De Guatemala



**Catedrático: Ing. Orlando Xeron**

**Curso: Sistemas Operativos I**

**Tema:**  
**Proyecto Final**

**Alumna: Merzi Caniz**

**Carnet: 2290-20-10279**

**Fecha de Entrega: 28 de octubre del 2023**

## Introducción

El presente documento contiene la información que necesita el usuario acerca del proyecto que se trabajo a lo largo de estos días en el curso de Sistemas Operativos el cual este interfaz grafico ayudara tanto, al usuario a interactuar dentro de una base llamada comandos Windows que son:

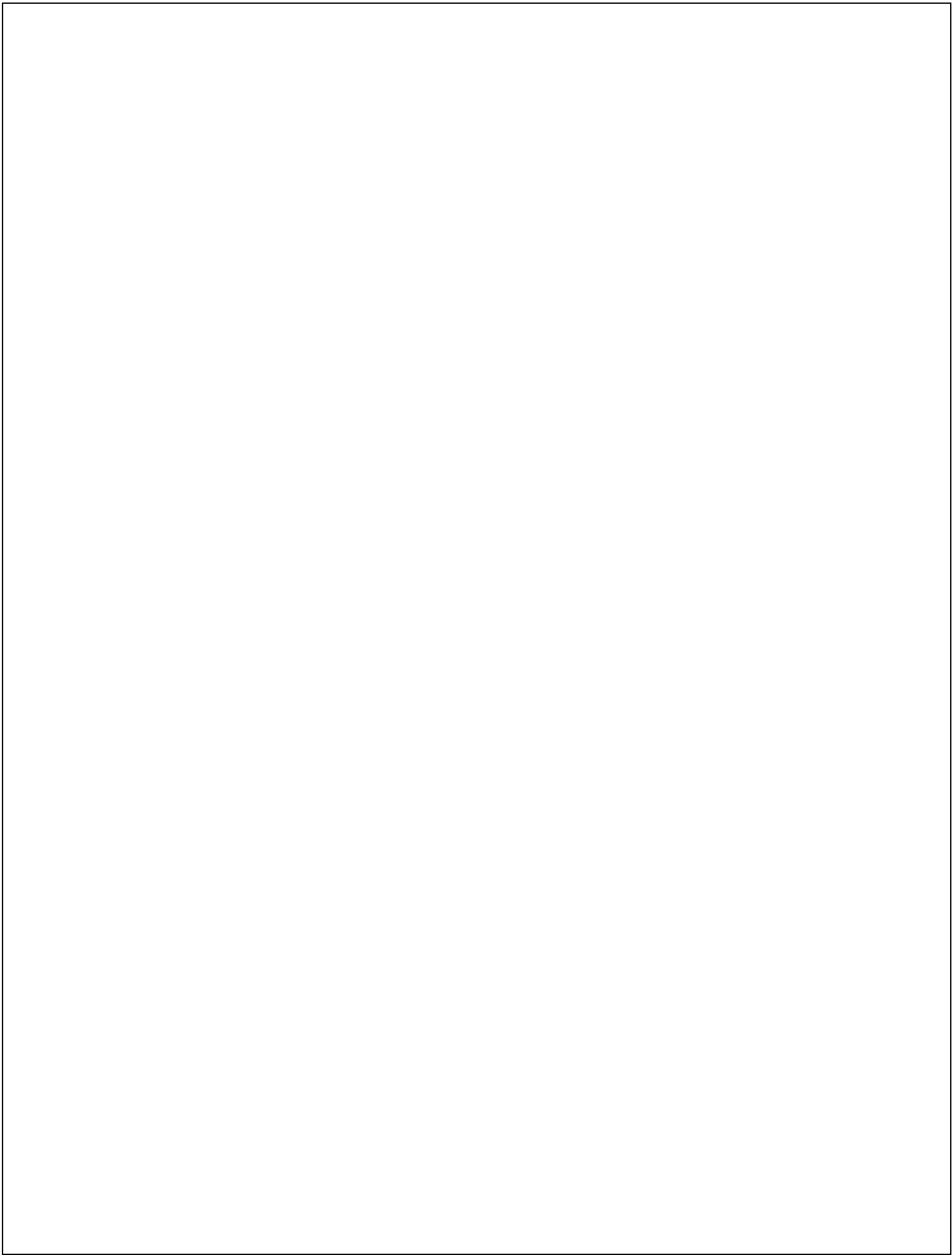
- ✓ Netstat
- ✓ Winver
- ✓ AppWiz
- ✓ DxDiag
- ✓ Tasklist
- ✓ Perfom

de igual manera tomando en cuenta lo visto en clase utilizamos las concurrencias e interbloqueo, tomando ejemplos y la definición de cada una

## índice

<b>Detalles de las opciones del Proyecto .....</b>	<b>6</b>
Hardware: .....	6
✓ DirectX .....	6
✓ Software:.....	6
✓ Appwiz.Cpl:.....	6
✓ Winver.....	6
✓ Tasklist: .....	6
✓ Perfmon: .....	6
✓ Netstat: .....	6
<b>Lenguaje Utilizado: .....</b>	<b>6</b>
Lenguaje C# .....	7
Librerías adicionales .....	7
Guna Library:.....	7
Análisis de Costo Beneficio del proyecto .....	7
<b>Librerías de iconos utilizados en el proyecto .....</b>	<b>7</b>
<b>Generalidades del Lenguaje C#.....</b>	<b>8</b>
<b>Características de C#.....</b>	<b>8</b>
Tipos de datos .....	9
Tipo de datos de enteros .....	9
<b>¿Qué puedo hacer con el lenguaje c#?.....</b>	<b>10</b>
Unity .....	10
Herramientas Adicionales .....	10
Infragistics .....	10
<b>TABLA COMPARATIVA .....</b>	<b>10</b>
<b>ULTRAGRID .....</b>	<b>11</b>
<b>Telerik.....</b>	<b>12</b>
<b>Formularios.....</b>	<b>12</b>
Parte del menú .....	13
Información de la versión de Windows .....	13
Información de la acción del comando DirectX.....	14
AppWiz .....	15
Tasklist: .....	16

<b>Netstat .....</b>	<b>17</b>
<b>Perfmon .....</b>	<b>18</b>
<b>Concurrencias.....</b>	<b>19</b>
<b>Interbloqueos.....</b>	<b>21</b>
<b>Código de como se llamaron a los comandos .....</b>	<b>23</b>
<b>Método para llamar a Winver: .....</b>	<b>23</b>
<b>Llamada de Netstat .....</b>	<b>25</b>
<b>Llamada de Perfmon .....</b>	<b>25</b>
<b>Conclusiones .....</b>	<b>26</b>
<b>Recomendaciones.....</b>	<b>27</b>
<b>Anexos.....</b>	<b>28</b>
<b>Egrafia .....</b>	<b>30</b>
<b>Enlace de la carpeta con el video, el manual de usuario y el proyecto como tal : .....</b>	<b>30</b>



## Detalles de las opciones del Proyecto

Hardware: aquellos elementos físicos o materiales que constituyen una computadora o un sistema informático. Es decir, son aquellas partes físicas de un sistema operativo tales como sus componentes eléctricos, electrónicos, electromecánicos, mecánicos y cualquier elemento físico que esté involucrado.

- ✓ **DirectX:** DirectX es una colección de API desarrolladas para facilitar las complejas tareas relacionadas con multimedia, especialmente programación de juegos y vídeo, en la plataforma Microsoft Windows. DirectX
- ✓ **Software:** término informático que hace referencia a un programa o conjunto de programas de cómputo, así como datos, procedimientos y pautas que permiten realizar distintas tareas en un sistema informático.
- ✓ **Appwiz.Cpl:** es un acceso directo de comando de ejecución para abrir Agregar o quitar programas o Desinstalar una lista de programas en Microsoft Windows XP, Vista, 7, 8 y 10.
- ✓ **Winver:** Una manera muy sencilla de comprobar el número de versión de Windows que tiene un ordenador cualquiera y saber si está registrado.
- ✓ **Tasklist:** Muestra una lista de los procesos que se están ejecutando actualmente en la computadora local o en una computadora remota. Tasklist reemplaza la herramienta tlist.
- ✓ **Perfmon:** Inicie el Monitor de confiabilidad y rendimiento de Windows en un modo independiente específico.
- ✓ **Netstat:** Muestra las conexiones TCP activas, los puertos en los que escucha la computadora, las estadísticas de Ethernet, la tabla de enrutamiento de IP, las estadísticas de IPv4 (para los protocolos IP, ICMP, TCP y UDP) y las estadísticas de IPv6 (para los protocolos IPv6, ICMPv6, TCP sobre IPv6). y UDP sobre protocolos IPv6). Usado sin parámetros, este comando muestra las conexiones TCP activas.

## Lenguaje Utilizado:

## **Lenguaje C#**

es un lenguaje de programación moderno, basado en objetos y con seguridad de tipos. C# permite a los desarrolladores crear muchos tipos de aplicaciones seguras y sólidas que se ejecutan en .NET. C# tiene sus raíces en la familia de lenguajes C

Para la creación de este proyecto utilizamos el lenguaje C# en su modalidad de Windows Forms

C# proporciona construcciones de lenguaje para admitir directamente estos conceptos, por lo que se trata de un lenguaje natural en el que crear y usar componentes de software, utilizamos los formularios para dar una vista agradable al usuario, de esta manera podemos observar que los códigos van “detrás” del diseño.

## **Librerías adicionales**

Guna Library: Es un framework que nos sirve para crear interfaces más estéticas y modernas.

Esto nos ayuda a ahorrar mucho tiempo de desarrollo de aplicaciones ya que para utilizar las herramientas no es necesario seleccionarlás para luego ubicarlas en el área de trabajo, simplemente debemos arrastrarlas lo que nos ayuda a ahorrar tiempo de diseño.

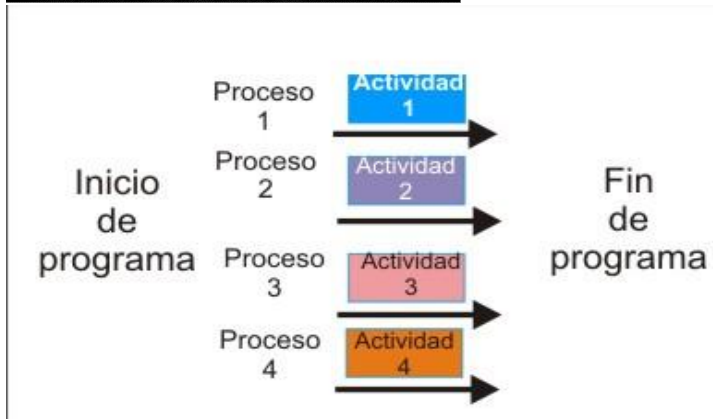
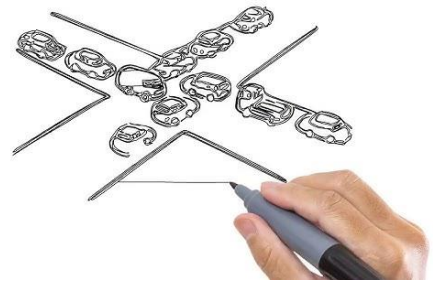
Actualmente está librería solo funciona para C# y VB.Net y es específicamente para aplicaciones con Windows Forms

## **Análisis de Costo Beneficio del proyecto**

Para la realización de este proyecto tenemos varios aspectos muy importantes que considerar, en base al lenguaje de programación que utilizaremos C#:

- ✓ Identificar la versión grafica de Windows instalada
- ✓ Identificar el software y hardware
- ✓ Identificar los procesos que están en ejecución (pausar, matar uno o todos los procesos)
- ✓ Integrar el monitoreo de los recursos de la computadora que está siendo evaluada
- ✓ Integrar las conexiones de red activas (entrantes y salientes)

## **Librerías de iconos utilizados en el proyecto**



## Generalidades del Lenguaje C#

C# (pronunciado "si sharp" en inglés) es un lenguaje de programación moderno, basado en objetos y con seguridad de tipos. C# permite a los desarrolladores crear muchos tipos de aplicaciones seguras y sólidas que se ejecutan en .NET

C# es un lenguaje de programación orientado a componentes, orientado a objetos. C# proporciona construcciones de lenguaje para admitir directamente estos conceptos, por lo que se trata de un lenguaje natural en el que crear y usar componentes de software. Desde su origen, C# ha agregado características para admitir nuevas cargas de trabajo y prácticas de diseño de software emergentes. En el fondo, C# es un lenguaje orientado a objetos.

## Características de C#

- ✓ Multiplataforma, ejecutable en los sistemas más comunes como Windows, MacOS, Linux



- ✓ Sintaxis similar a C, C++, Java y otros
- ✓ Lenguaje de paradigma de programación orientada a objetos, con expresiones de control heredadas de la programación estructurada
- ✓ Incluye algunas características de programación funcional como clojures, aunque es imperativo.
- ✓ Fuertemente tipado (tipado estático)
- ✓ Lenguaje moderno con actualizaciones de mejoras frecuentes
- ✓ Dispone de un nutrido conjunto de librerías
- ✓ Orientado a componentes

## Tipos de datos

C# contiene veinte categorías generales de tipos de datos integrados: tipos de valor y tipos de referencia. El término tipo de valor indica que esos tipos contienen directamente sus valores. Tipos para definir números enteros:

Tipo de datos de enteros				
Tipo	Equivalente BCL	Tamaño	Intervalo	Significado
byte	System.Byte	8-bit (1-byte)	0 a 255	Entero sin signo
sbyte	System.SByte	8-bit (1-byte)	-128 a 127	Entero con signo
short	System.Int16	16-bit (2-byte)	-32.768 a 32.767	Entero corto con signo
ushort	System.UInt16	16-bit (2-byte)	0 a 65.535	Entero corto sin signo
int	System.Int32	32-bit (4-byte)	-2.147.483.648 a 2.147.483.647	Entero medio con signo
uint	System.UInt32	32-bit (4-byte)	0 a 4.294.967.295	Entero medio sin signo
long	System.Int64	64-bit (8-byte)	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807	Entero largo con signo
ulong	System.UInt64	64-bit (8-byte)	0 a 18.446.744.073.709.551.615	Entero largo sin signo

Enteros	
decimal	245, [0..9]+

hexadecimal	0xF5, 0x[0..9, A..F, a..f]+		
entero largo	12L		
entero largo sin signo	654UL		
Coma flotante			
float	23.5F, 23.5f, 1.72E3F, 1.72E3f, 1.72e3F, 1.72e3f		
double	23.5, 23.5D, 23.5d, 1.72E3, 1.72E3D		
decimal	9.95M		
Caracteres			
char	'a', 'Z', '\u0231'		
Cadenas			
String	"Hello, world"; "C:\\Windows\\", @"C:\\Windows\\"		
Secuencias de escape			
Alerta (timbre)	\\a	Nulo	\\0
Retroceso	\\b	Comilla simple	\\'
Avance de página	\\f	Comilla doble	\\"
Nueva línea	\\n	Barra inversa	\\
Retorno de carro	\\r		
Tabulador horizontal	\\t		
Tabulador vertical	\\v		

## ¿Qué puedo hacer con el lenguaje c#?

### Unity

Es un programa para crear videojuegos, muy popular últimamente por todas las posibilidades que tiene. Su fama la tiene porque su motor es multiplataforma, por lo que permite exportar los videojuegos a ordenadores, dispositivos móviles con Android o iOS, y las diferentes videoconsolas de mercado. Además, es posible programar de diversos lenguajes, dentro de los cuales esta C#, gracias a Mono DEVELOP.

### Herramientas Adicionales

#### Infragistics

Infragistics es una colección o paquete de controles que extienden de los controles normales de las herramientas de programación de Windows que permite el uso de mejorados controles que permiten hacer aplicaciones poderosas. Por ello, es importante conocerlas. Infragistics tiene varias herramientas como:

## TABLA COMPARATIVA

CARACTERÍSTICA	ULTRAGRID	DATAGRIDVIEW
----------------	-----------	--------------

<i>Datos</i>	Mostrar datos con o sin un origen de datos.	Mostrar datos con o sin un origen de datos.
<i>Clasificación de datos</i>	Clasificación de datos por las columnas de datos única o múltiple sin necesidad de código.	Orden de datos por filtros manuales mediante código.
<i>Filtros</i>	Filtros de drenaje, la creación de filtros personalizados.	Filtros mediante la programación de código.
<i>Formatos</i>	Formatos personalizados o automáticos ilimitados con el diseñador (cardView o fila diseño)	Formatos y estilos básicos personalizados de forma manual mediante propiedades del control
<i>Datos virtuales</i>	Maximiza el rendimiento y mostrar un número ilimitado de filas y columnas mediante WinDataSource para cargar los datos a la red y desde la red.	
<i>Excel</i>	Exportación a Excel, copiar / pegar desde Excel y el uso fórmulas de Excel.	Exportación desde Excel mediante código y el uso de otros controles, hacia Excel se pueden copiar y pegar la información de los registros

## ULTRAGRID

VENTAJAS	DESVENTAJAS
Múltiples agrupaciones de filtros	Es necesario la compra e instalación de una extensión de la visual 2020, Wingrid o la versión trial, pero es por un tiempo indefinido
Automatización de datos jerárquicos	
Filas, cabeceras y pies de páginas fijas	
Resúmenes de comunas	
Explotación de Microsoft Excel	
Vista de impresión e impresión de la tabla	
Agradas de botones de insertar, borrar, editar automáticos	

## Telerik

Crea, diseña, visualiza y exporta informes .NET con la solución de informes .NET completa, potente y liviana para aplicaciones web y de escritorio

Telerik AD (Телерик АД) es una empresa búlgara que ofrece herramientas de software para web, móviles, desarrollo de aplicaciones de escritorio, herramientas y servicios de suscripción para el desarrollo de aplicaciones multiplataforma. Fundada en 2002 como una empresa enfocada en herramientas de desarrollo .NET, Telerik ahora también vende una plataforma para el desarrollo de aplicaciones web, híbridas y nativas.

## Formularios



## Parte del menú



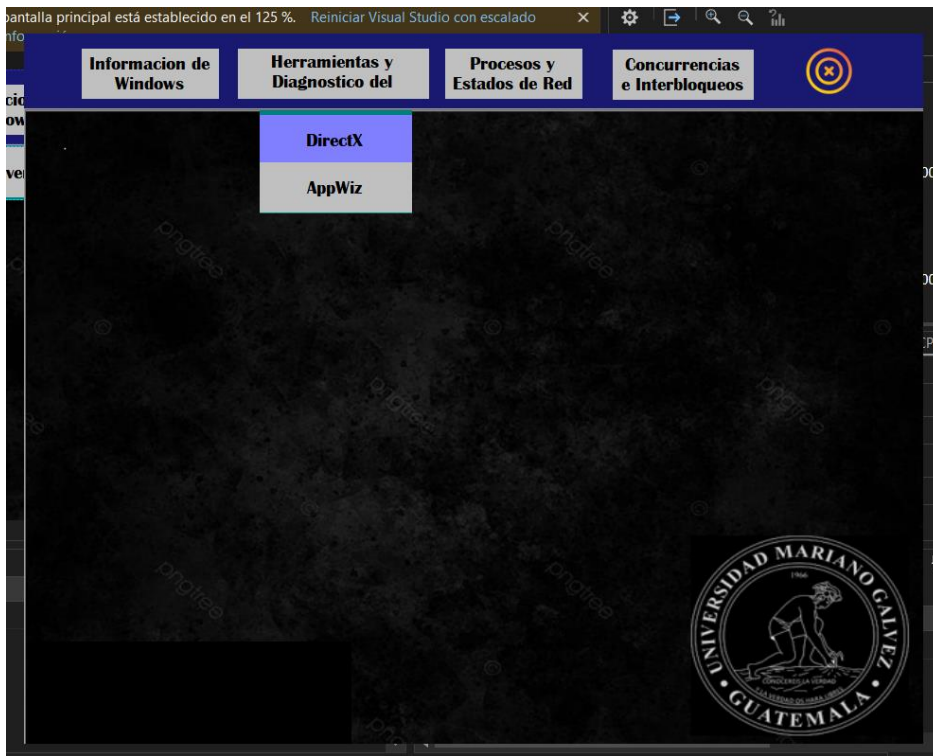
## Información de la versión de Windows

Para poder seleccionar esta información vamos al apartado donde dice Información de Windows y picarle luego de realizar la acción nos desplegará Winver nos dirigimos a picarle y nos desglosará la pantalla siguiente

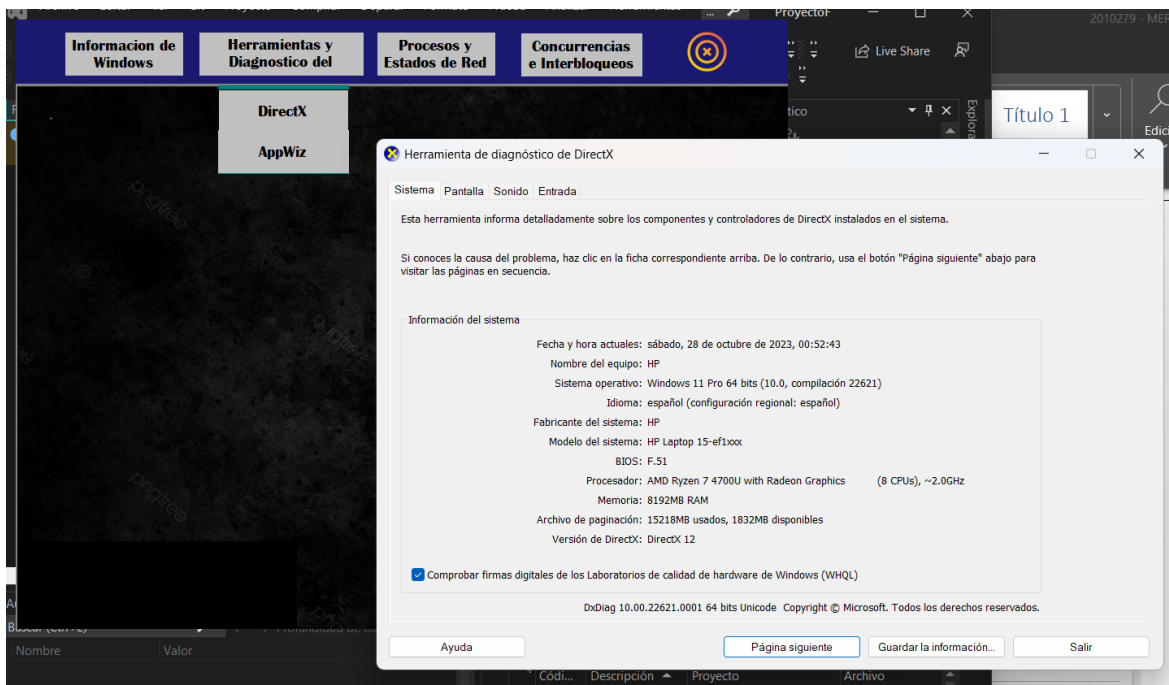


## Información de la acción del comando DirectX

Para poder ver el comando de DirectX vamos al apartado donde dice Herramientas y Diagnóstico al seleccionarlo nos aparecerá estas opciones en la imagen siguiente



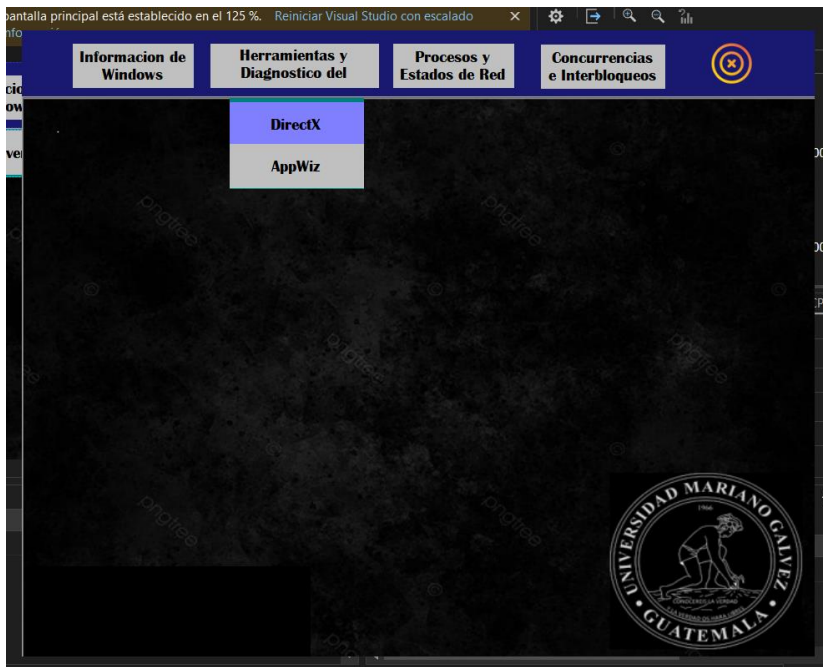
Ya al seleccionar la opción a nuestra elección en este caso será Dxdiag nos mostrará la siguiente pagina a visualizar



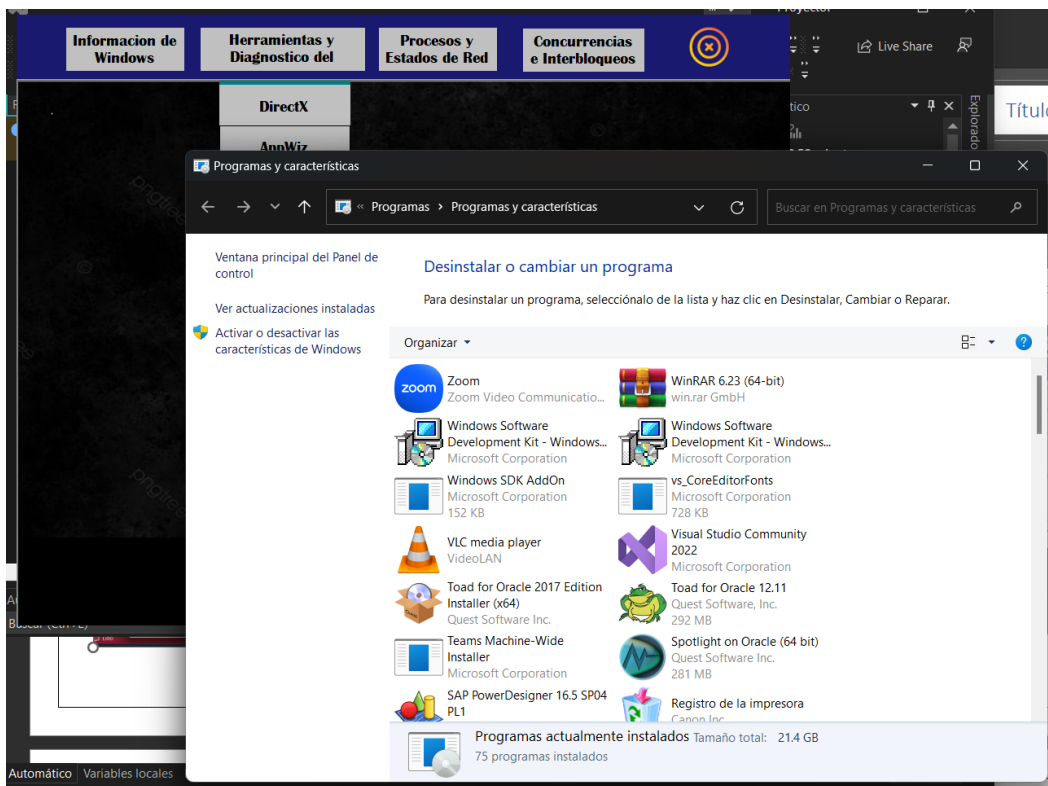


## AppWiz

Para poder ver la opción de AppWiz vamos al apartado donde dice Herramientas y Diagnóstico al seleccionarlo nos aparecerá estas opciones en la imagen siguiente

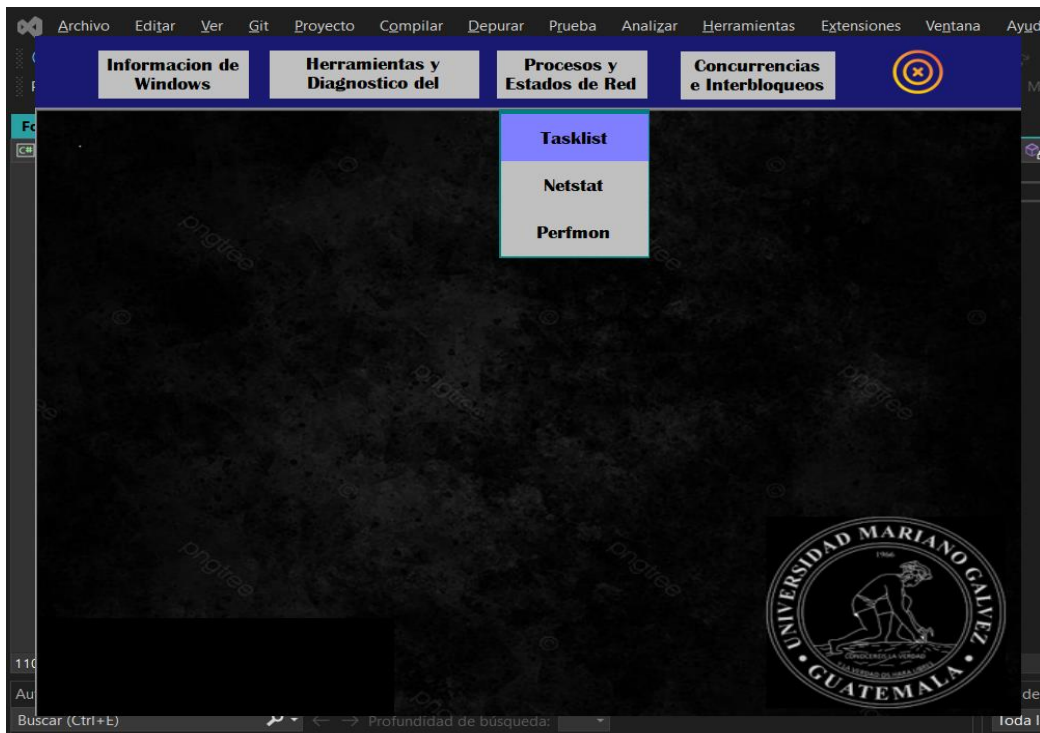


Ya al seleccionar la opción a nuestra elección en este caso será AppWiz nos mostrará la siguiente página a visualizar

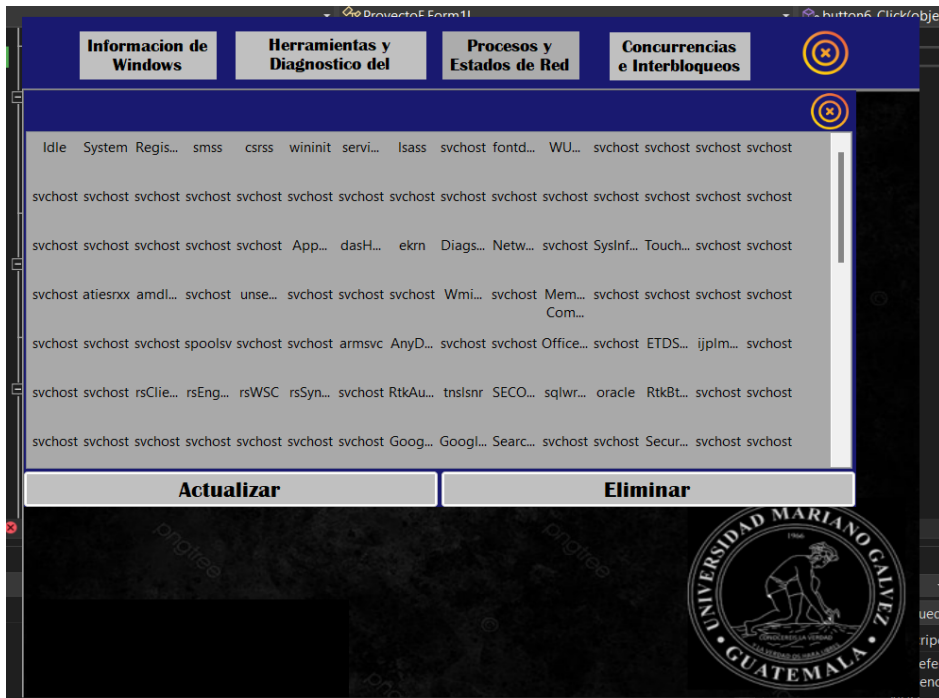


## Tasklist:

Para poder ver la opción de Tasklist vamos al apartado donde dice Procesos y estado de Red al seleccionarlo nos aparecerá estas opciones en la imagen siguiente



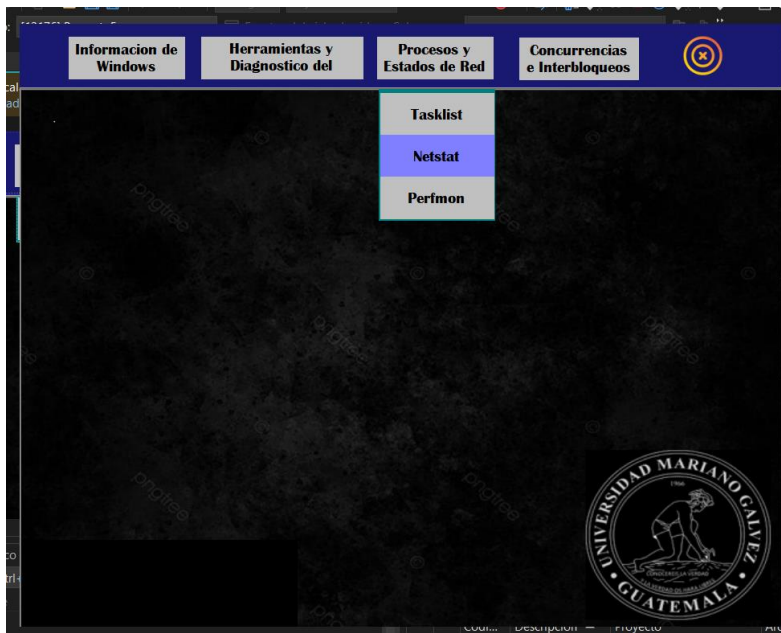
Ya al seleccionar la opción a nuestra elección en este caso será Tasklist nos mostrará la siguiente página a visualizar



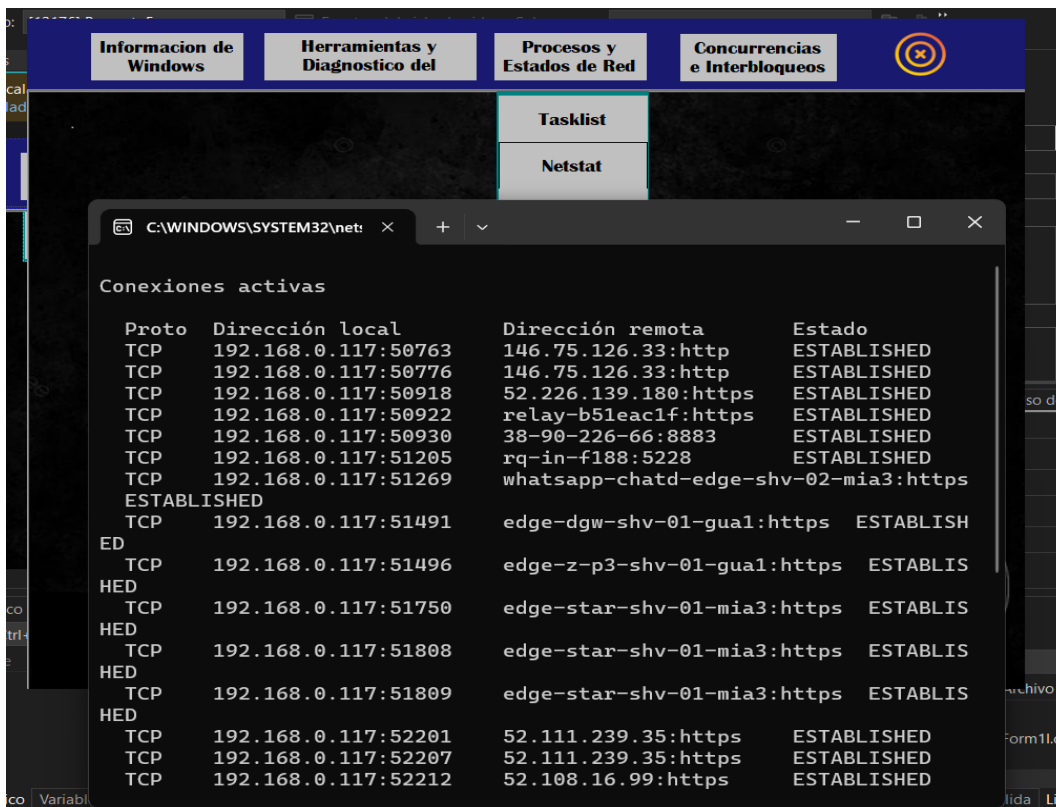


## Netstat

Para poder ver la opción de Netstat vamos al apartado donde dice Procesos y estado de Red al seleccionarlo nos aparecerá estas opciones en la imagen siguiente

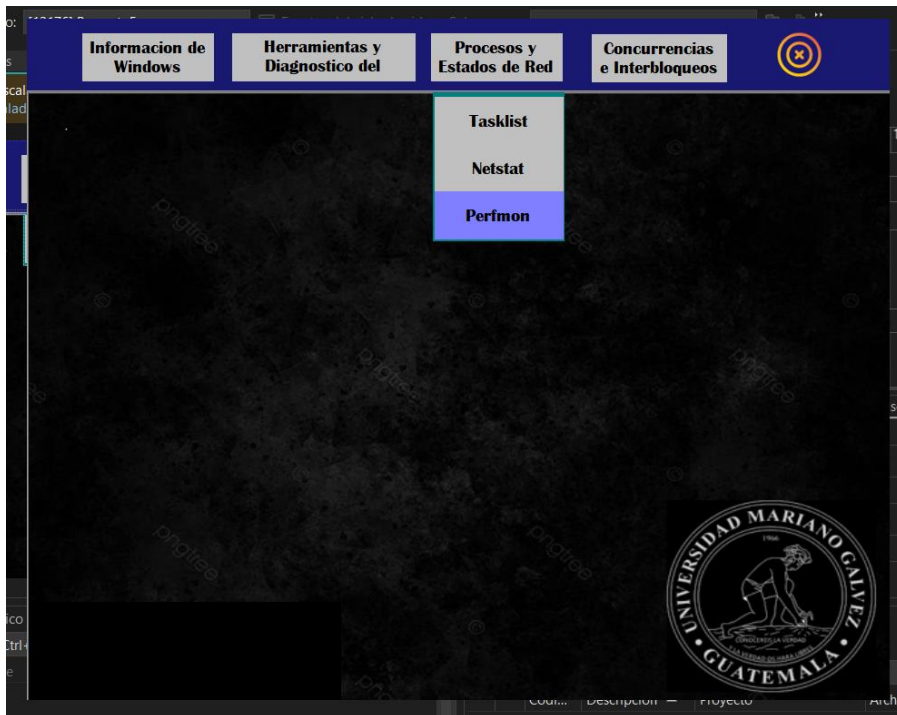


Ya al seleccionar la opción a nuestra elección en este caso será Netstat nos mostrará la siguiente página a visualizar

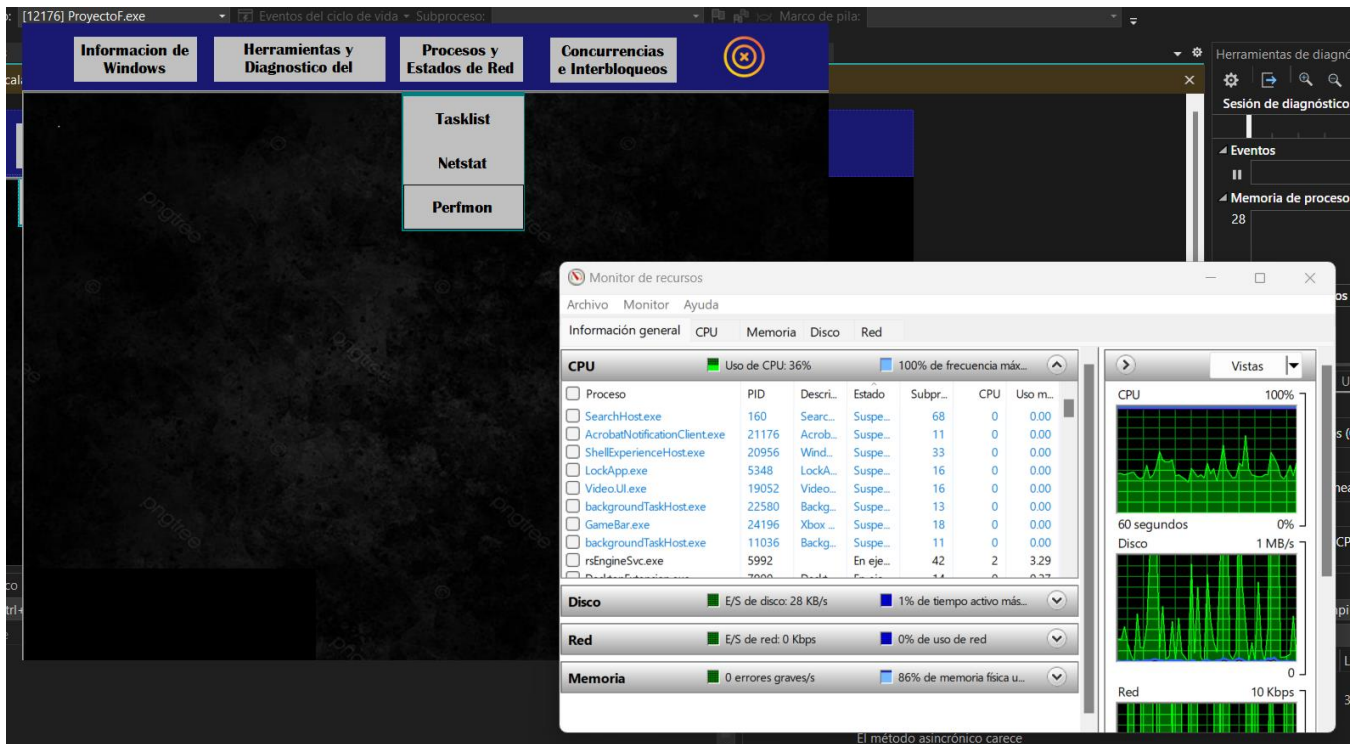


## Perfmon

Para poder ver la opción de Perfmon vamos al apartado donde dice Procesos y estado de Red al seleccionarlo nos aparecerá estas opciones en la imagen siguiente



Ya al seleccionar la opción a nuestra elección en este caso será Perfmon nos mostrará la siguiente página a visualizar



## Concurrencias

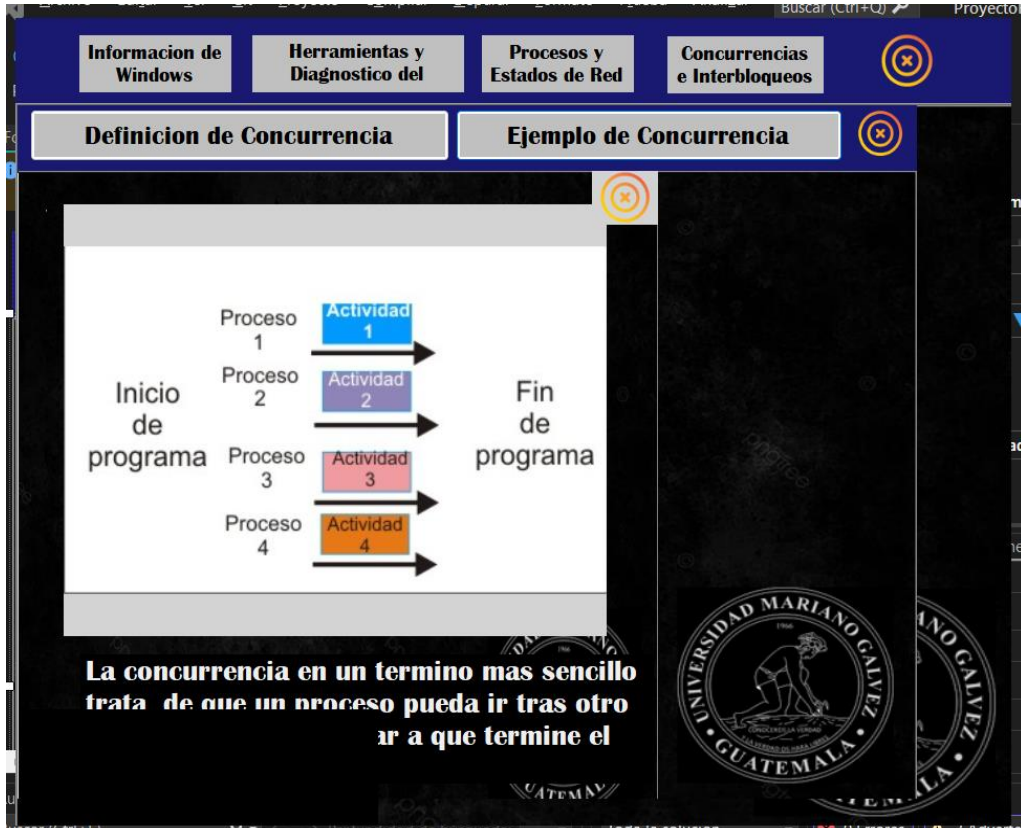
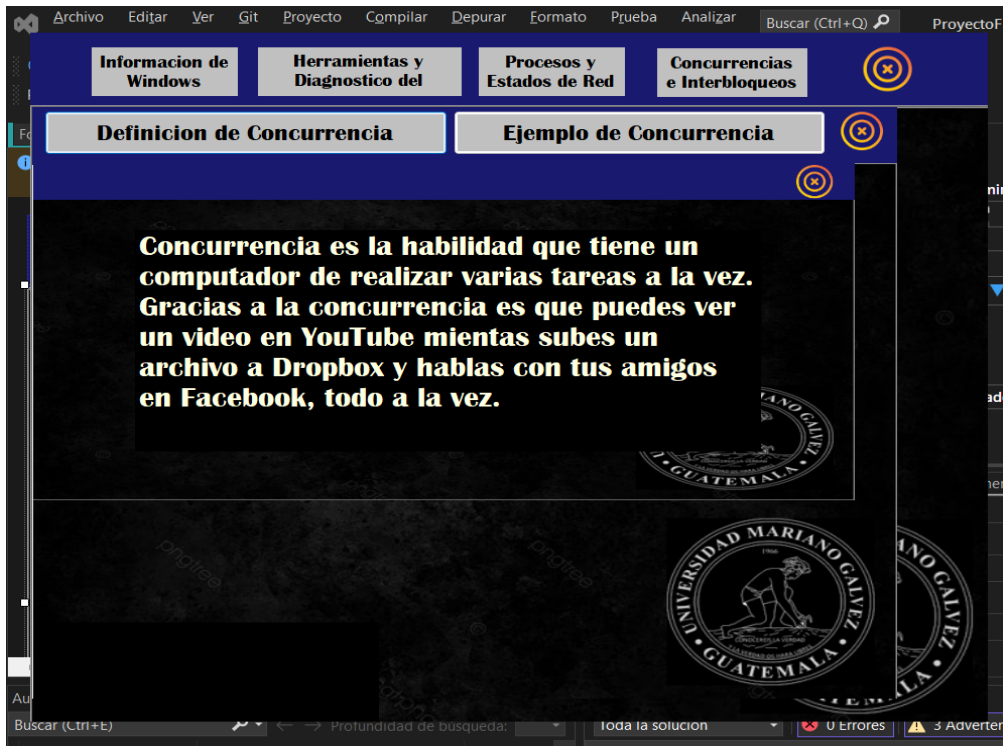
Para poder ver la opción de Concurrencias vamos al apartado donde dice Concurrencias e interbloqueo al seleccionarlo nos aparecerá estas opciones en la imagen siguiente



Ya al seleccionar la opción a nuestra elección en este caso será Concurrencias nos mostrará la siguiente página a visualizar



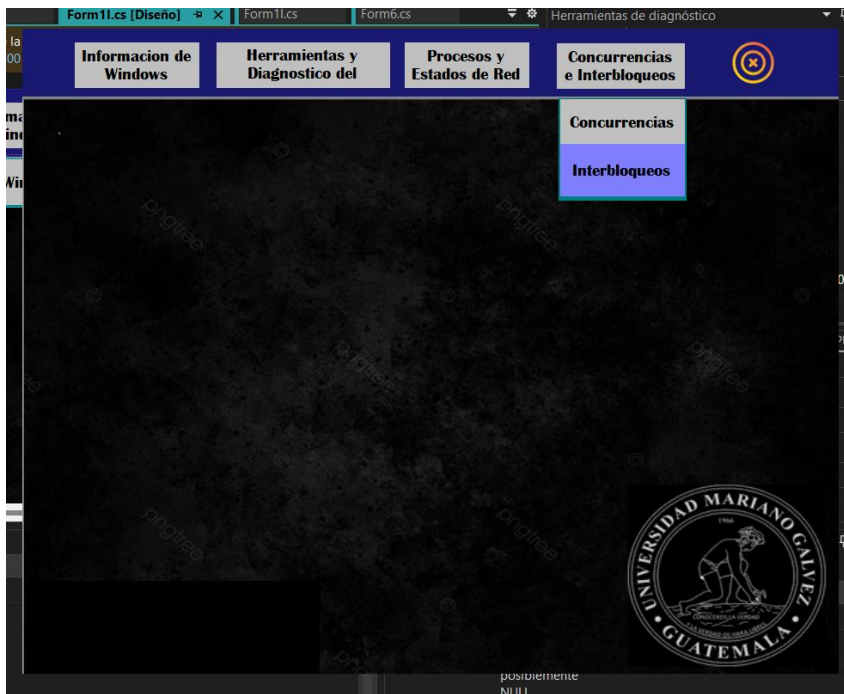
La misma ventana de concurrencias como vemos en la imagen nos aparece la definición como tal y un ejemplo de la misma concurrencia aquí lo podemos visualizar en las ilustraciones





## Interbloqueos

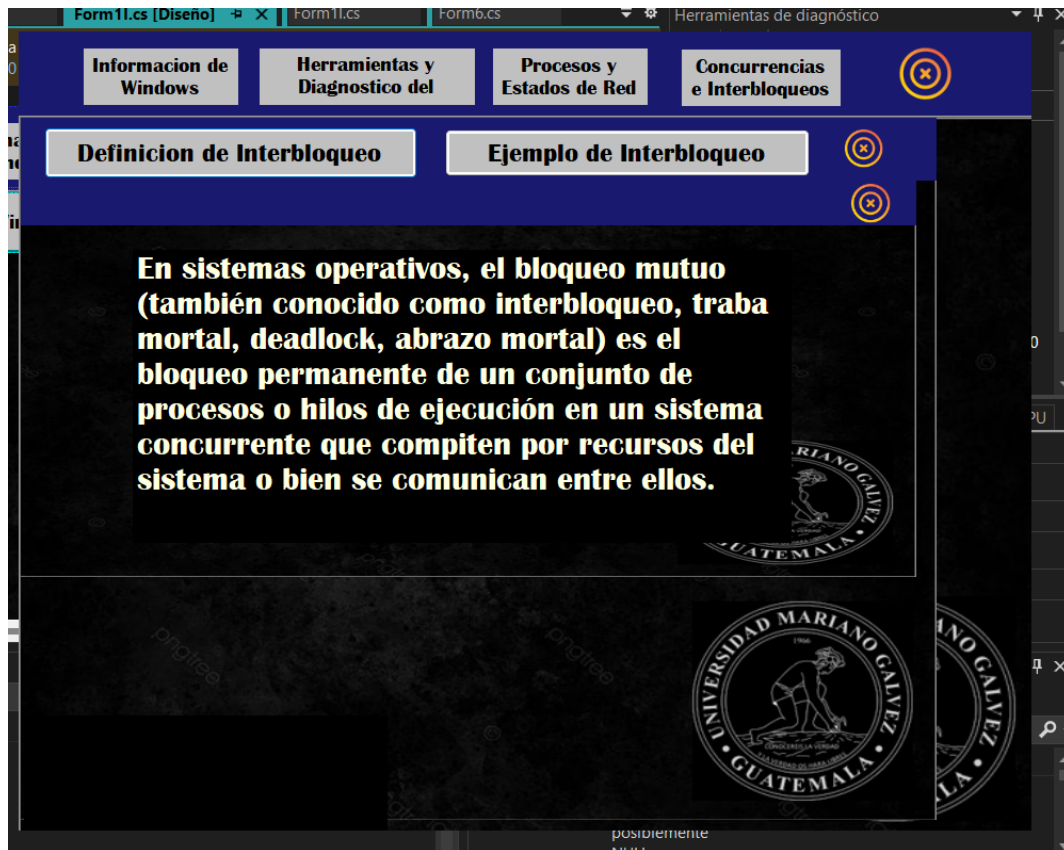
Para poder ver la opción de Interbloqueos vamos al apartada donde dice Concurrencias e interbloqueo al seleccionarlo nos aparecerá estas opciones en la imagen siguiente



Ya al seleccionar la opción a nuestra elección en este caso será Interbloqueo nos mostrará la siguiente página a visualizar



La misma ventana de Interbloqueo como vemos en la imagen nos aparece la definición como tal y un ejemplo de Interbloqueo aquí lo podemos visualizar en las ilustraciones



## Código de como se llamaron a los comandos

Método para llamar a Winver:

```
private void button9_Click(object sender, EventArgs e)
{
    Process p = Process.Start("winver.exe");
    Thread.Sleep(500);
    SetParent(p.MainWindowHandle, pResultado.Handle);
}
```

Llamada a DirectX:

```
private void button6_Click(object sender, EventArgs e)
{
    Process p = Process.Start("DirectX.exe");
    Thread.Sleep(500);
    SetParent(p.MainWindowHandle, pResultado.Handle);
}
```

Llamada a AppWiz:

```
private void button7_Click_1(object sender, EventArgs e)
{
    Process p = Process.Start("control.exe", "appwiz.cpl");
    SetParent(p.MainWindowHandle, pResultado.Handle);
}
```

Llamada a Tasklist:

```
private void button10_Click(object sender, EventArgs e)
{
    Tasklist h = new Tasklist();
    AddFormulario(h);
}
```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace ProyectoF
{
    public partial class Tasklist : Form
    {
        public Tasklist()
        {
            InitializeComponent();
            UpdateProcessList();
            timer1.Enabled = true;
        }
        private void UpdateProcessList()
        {
            listView1.Items.Clear();
            int id = 1;
            foreach (Process p in Process.GetProcesses())
            {
                ListViewItem item = new ListViewItem();
                item = listView1.Items.Add(p.ProcessName);
                item.SubItems.Add(p.Id.ToString());
                item.SubItems.Add(p.WorkingSet64.ToString());
                item.SubItems.Add(p.VirtualMemorySize64.ToString());
                item.SubItems.Add(p.SessionId.ToString());
                id = id + 1;
            }
        }
        private void button1_Click(object sender, EventArgs e)
        {
            UpdateProcessList();
        }
        private void button2_Click(object sender, EventArgs e)
        {
            try
            {
                foreach (Process p in Process.GetProcesses())
                {
                    string arr = listView1.SelectedItems[0].SubItems[0].Text;
                    if (p.ProcessName == arr)

```



```

        {
            p.Kill();
        }
    }

    }
    catch (Exception x)
    {
        MessageBox.Show("no selecciono nignun proceso"
+ x, "error al eliminar", MessageBoxButtons.OK);
    }
}

```

#### Llamada de Netstat

```

private async void button8_Click(object sender, EventArgs e)
{
    Process p = Process.Start("netstat.exe");
    Thread.Sleep(500);
    SetParent(p.MainWindowHandle, pResultado.Handle);

    /*netstat h = new netstat();
    AddFormulario(h);*/
}

```

#### Llamada de Perfmon

```

private void button11_Click_1(object sender, EventArgs e)
{
    Process p = Process.Start("resmon.exe");
    SetParent(p.MainWindowHandle, pResultado.Handle);
}

```

## Conclusiones

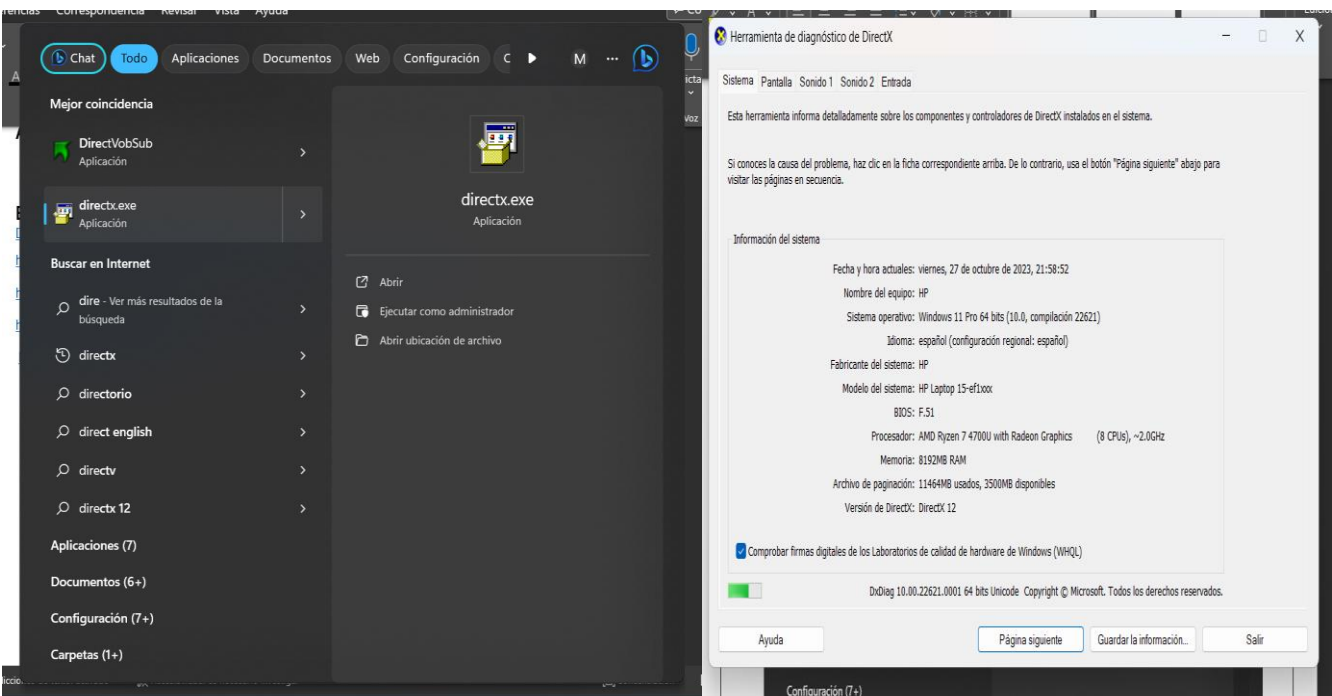
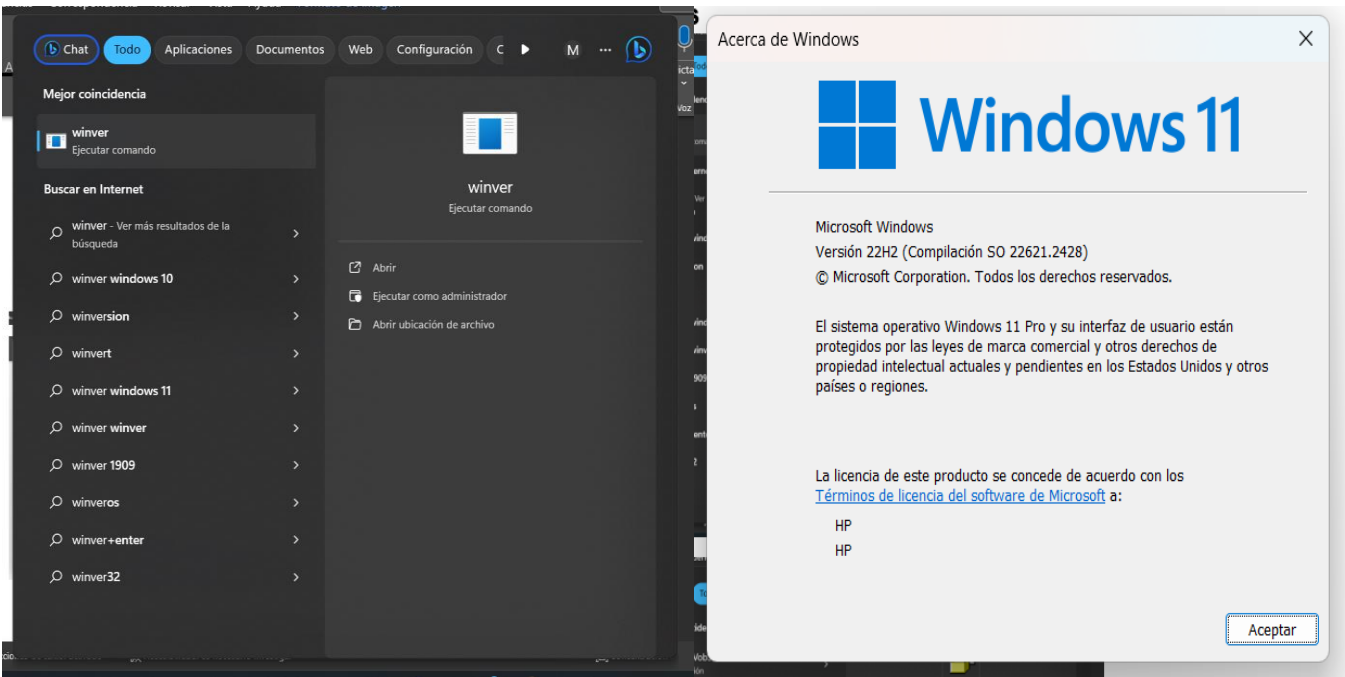
En conclusión, el proyecto requirió muchas faces entre ellas saber programar y poder llamar a los procesos dentro de códigos y como también entre unas de las dificultades que me genero al momento de programar he fucionarlos para que sea eficiente fueron

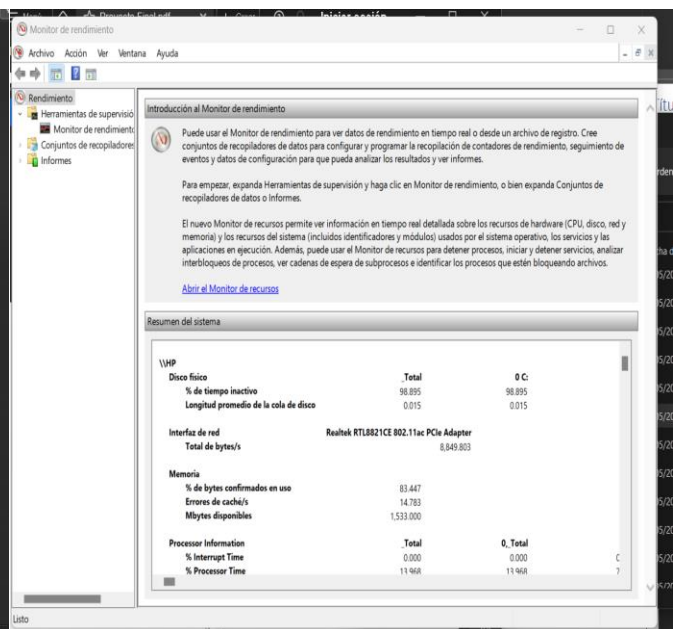
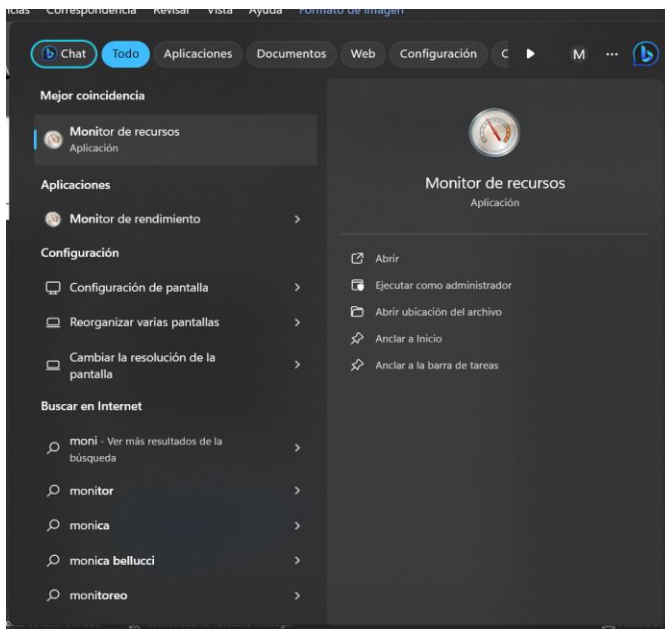
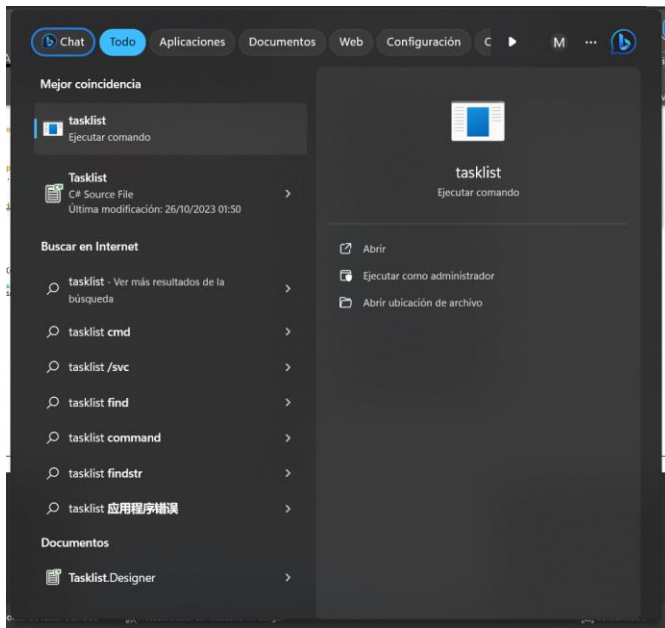
1. Saber como se llamaban y tener bien claro las funciones ya que unos se utilizaron formularios
2. Y había un problema con la llamada de los mismos formularios que ya no logre que encajar dentro de un forms y me genera con pantallas independientes, pero siempre y cuando llama a las ventanas
3. Tenia que investigar bien lo de como llamar y como eran las opciones de Netstat y tasklist que son comandos propiamente de Windows que ni sabia he indague mucho para poder entender mejor

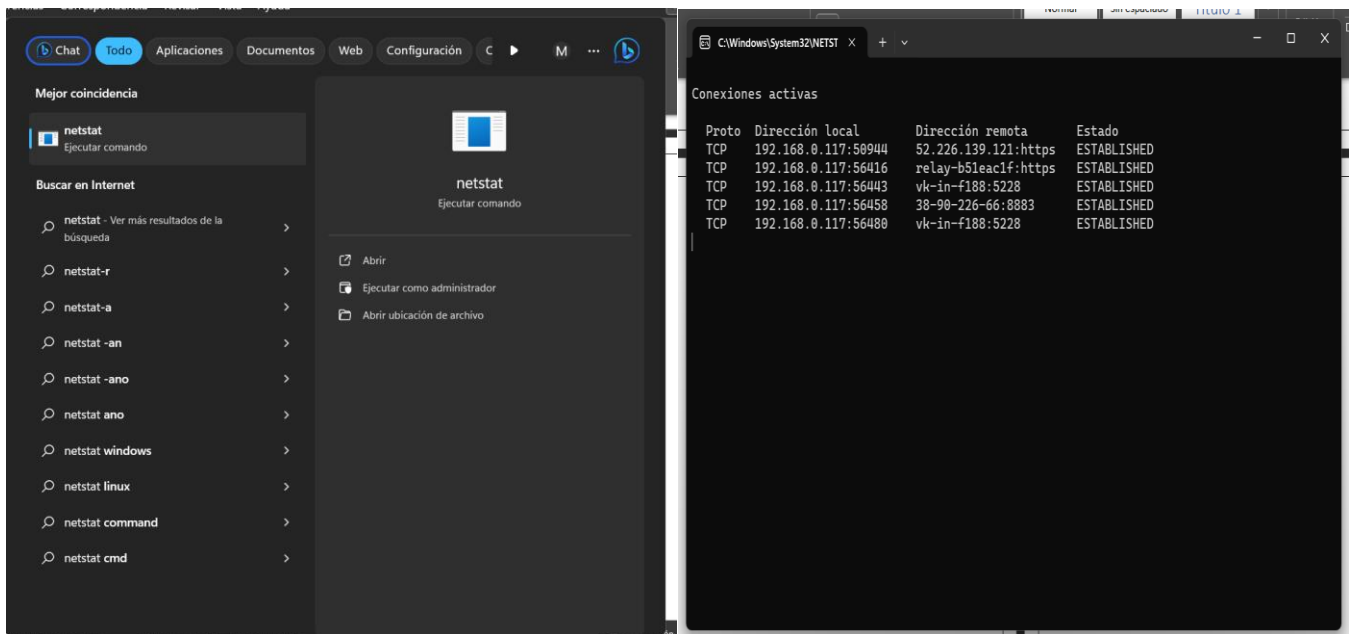
## **Recomendaciones**

1. Tenemos que ver que es el tipo de maquinas como poder utilizarla
2. Saber para que funciona saber el tipo de datos que nos va a mostrar para que nos va a servir
3. Sabes que tipo de es
4. Alguna mejora al proyecto ya sea por el sistema grafico o utilizar el color de la paleta de la empresa en algún caso

# Anexos







## Egrafía

[https://es.wikipedia.org/wiki/C\\_Sharp#:~:text=%22C%23%22%20\(pronunciado%20cii%20sharp,\(ISO%2FIEC%2023270\).](https://es.wikipedia.org/wiki/C_Sharp#:~:text=%22C%23%22%20(pronunciado%20cii%20sharp,(ISO%2FIEC%2023270).)

[Descargar DirectX Gratis en Español \(portalprogramas.com\)](https://portalprogramas.com/)

<https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/tasklist>

[https://es.wikipedia.org/wiki/C\\_Sharp](https://es.wikipedia.org/wiki/C_Sharp)

<http://eve-ingsistemas-u.blogspot.com/2012/04/infragistics.html>

<https://en.wikipedia.org/wiki/Telerik>

Enlace de la carpeta con el video, el manual de usuario y el proyecto como tal :

[https://drive.google.com/drive/folders/1JcVqOc-fMyui8r\\_utEuzvk3NEhOxcx4e](https://drive.google.com/drive/folders/1JcVqOc-fMyui8r_utEuzvk3NEhOxcx4e)