

TECNOLÓGICO NACIONAL DE MÉXICO  
INSTITUTO TECNOLÓGICO DE TLAXIACO

**REPORTE DE VISUAL STUDIO UI.**

**PRESENTAN:**

ISAIAS BRAYAN LOPEZ DOMINGUEZ

MAYRA GONZALEZ LITA

PAOLA ROSARIO AYALA

ARIADNA BELEN BERNABE JUAREZ

**ASIGNATURA:**

TOPICOS AVANZADOS DE PROGRAMACION

**CARRERA:**

INGENIERÍA EN SISTEMAS COMPUTACIONALES

**DOCENTE:**

ING. ROMAN CRUZ JOSE ALFREDO



TLAXIACO, OAXACA, 14 DE FEBRERO DE 2025.

*"Educación, Ciencia y Tecnología, Progreso  
día con día"®*

## ÍNDICE

<b>INTRODUCCIÓN.....</b>	<b>4</b>
<b>OBJETIVO DE LA PRÁCTICA.....</b>	<b>5</b>
<b>MATERIAL. ....</b>	<b>5</b>
<b>PROCEDIMIENTO DEL DISEÑO EN VISUAL STUDIO UI.....</b>	<b>6</b>
<b>RESULTADOS.....</b>	<b>22</b>
<b>CONCLUSIÓN.....</b>	<b>23</b>

## LISTA DE IMÁGENES

Figure 1 Pantalla De Inicio de sesión -----	6
Figure 2 Código de la Pantalla de Inicio de sesión -----	6
Figure 3 Código del Inicio de sesión -----	7
Figure 4 Código del Inicio de sesión -----	7
Figure 5 Código del resultado del registro -----	7
Figure 6 Diseño de Registro -----	8
Figure 7 Código del diseño de registro -----	8
Figure 8 Código de la Interfaz del diseño de Registro -----	9
Figure 9 Código de la Interfaz del diseño de Registro -----	9
Figure 10 Resultado de la Interfaz del Diseño de Registro -----	10
Figure 11 Interfaz de la Pantalla Principal -----	10
Figure 12 Código de la Interfaz de la pantalla de Principal -----	11
Figure 13 Código de la Interfaz de la Pantalla Principal -----	11
Figure 14 Código de la Interfaz de la Pantalla de Inicio -----	12
Figure 15 Código de la Interfaz de la pantalla de Inicio -----	12
Figure 16 Código de la Interfaz de la pantalla de Inicio -----	13
Figure 17 Código de la Interfaz de la pantalla Principal -----	13
Figure 18 Diseño final de la pantalla Principal -----	14
Figure 19 Diseño de la Interfaz de Agendar Cita -----	14
Figure 20 Código de la Interfaz de Agendar Cita -----	15
Figure 21 Código de la Interfaz de Agendar Cita -----	15
Figure 22 Diseño final de la Interfaz Agendar cita -----	16
Figure 23 Diseño de la Interfaz de Ubicación de Módulos -----	16
Figure 24 Código de la Interfaz de Ubicar Módulos -----	17
Figure 25 Código de la Interfaz Ubicar Módulos -----	17
Figure 26 Código de la Interfaz de Ubicar Módulos -----	18
Figure 27 Código de la Interfaz de Ubicar Módulos -----	18
Figure 28 Diseño Final de la Interfaz de Ubica Tu Módulo -----	19
Figure 29 Diseño de la Interfaz de Trámites -----	19
Figure 30 Código de la Interfaz de Trámites -----	20
Figure 31 Código de la Interfaz de Trámites -----	21
Figure 32 Resultado del Diseño de Trámites -----	21

## INTRODUCCIÓN

El objetivo de esta práctica es aplicar los principios de diseño y desarrollo de software en la creación de una aplicación que permita la interacción del usuario con información electoral. Se trabajará con C# (C Sharp) y Windows Forms para desarrollar una interfaz gráfica intuitiva y funcional.

En esta práctica, desarrollaremos el proceso de diseño y creación de una aplicación utilizando Visual Studio, desde la configuración del entorno hasta la implementación de funcionalidades básicas.

Al diseñar una interfaz gráfica con controles interactivos aplicando principios de UI/UX. Luego, se desarrollan funcionalidades para la consulta y gestión de datos del INE (simulados), implementando eventos y programación orientada a objetos.

El desarrollo de esta aplicación permitirá a nosotros los estudiantes fortalecer nuestras habilidades en programación y diseño de interfaces gráficas, al mismo tiempo que comprenderemos cómo se pueden gestionar datos en aplicaciones orientadas a programación. Además, se fomentará la aplicación de buenas prácticas en el desarrollo de software, asegurando la creación de sistemas seguros, eficientes y fáciles de usar.

## **OBJETIVO DE LA PRÁCTICA.**

Los objetivos esperados de esta práctica deben enfocarse en la creación de una aplicación funcional en Visual Studio. A través de este proyecto desarrollaremos competencias en diseño de interfaces y en el manejo adecuado de datos en aplicaciones de escritorio.

## **MATERIAL.**

### **1. Programa Visual Studio UI**

## PROCEDIMIENTO DEL DISEÑO EN VISUAL STUDIO UI. DESCRIPCIÓN DE LAS CADA UNA DE LAS ACTIVIDADES

### ♣ Creación De Login.

Diseño de la pantalla de Inicio de sesión.

Apartado en el cual el usuario se registra o inicia sesión en la aplicación.



Figure 1 Pantalla De Inicio de sesión

### ♣ Código (Comentado)

```
0 referencias
private void Form1_Resize(object sender, EventArgs e)
{
    // Centrar el Label horizontal y verticalmente en el formulario
    Titulo.Left = (this.ClientSize.Width - Correo.Width) / 2;
    Titulo.Top = (this.ClientSize.Height - Correo.Height) / 2;
}

1 referencia
private void panel1_Paint(object sender, PaintEventArgs e)
{
    Contenedor_1.Dock = DockStyle.Fill;
    Titulo.BackColor = ColorTranslator.FromHtml("#FFD6D0");
    this.BackColor = ColorTranslator.FromHtml("#FFD6D0");
}

1 referencia
private void tableLayoutPanel1_Paint(object sender, PaintEventArgs e)
{
    Menu_1.Width = 300;
    Menu_1.Height = 270;
    this.BackColor = ColorTranslator.FromHtml("#FFB3C1");
}

1 referencia
private void panel1_Paint_1(object sender, PaintEventArgs e)
{
    //this.BackColor = ColorTranslator.FromHtml("#FFB3C1");
    panell.Width = 300;
    panell.Height = 270;
}
```

Figure 2 Código de la Pantalla de Inicio de sesión

```
int radius = 40; // Radio de las esquinas
GraphicsPath path = new GraphicsPath();
path.AddArc(0, 0, radius, radius, 180, 90);
path.AddArc(panel.Width - radius, 0, radius, radius, 270, 90);
path.AddArc(panel.Width - radius, panel.Height - radius, radius, radius, 0,
path.AddArc(0, panel.Height - radius, radius, radius, 90, 90);
path.CloseFigure();
panel.Region = new Region(path);
panel.Left = (this.ClientSize.Width - panel.Width) / 2;
panel.Top = (this.ClientSize.Height - panel.Height) / 2;
}

0 referencias
private void textBox1_Paint(object sender, PaintEventArgs e)
{
    // Establecer el padding para el TextBox
    int paddingLeft = 20;
    int paddingTop = 10;
    int paddingRight = 20;
    int paddingBottom = 10;
    // Crear un rectángulo con el padding definido
    Rectangle rect = new Rectangle(
        paddingLeft,
        paddingTop,
        C_correo.ClientSize.Width - paddingLeft - paddingRight,
        C_correo.ClientSize.Height - paddingTop - paddingBottom
    );
    // Dibujar el texto dentro del área con padding
    e.Graphics.FillRectangle(Brushes.White, rect);
}
```

Figure 3 Código del Inicio de sesión

```
e.Graphics.DrawString(C_correo.Text, C_correo.Font, Brushes.Black, rect);
}
1 referencia
private void button1_Click(object sender, EventArgs e)
{
    // Crea una instancia de Form2
    Form2 form2 = new Form2();
    // Muestra el Form2
    form2.Show(); // Abre Form2 de forma no modal (el usuario puede interactuar
}
}
```

Figure 4 Código del Inicio de sesión

## ♣ Resultado del diseño.

Figure 5 Código del resultado del registro

## ♣ Diseño De Interfaz De Registro

En este interfaz de registro cuenta con botones y apartados con respecto a la informacion solicitada.

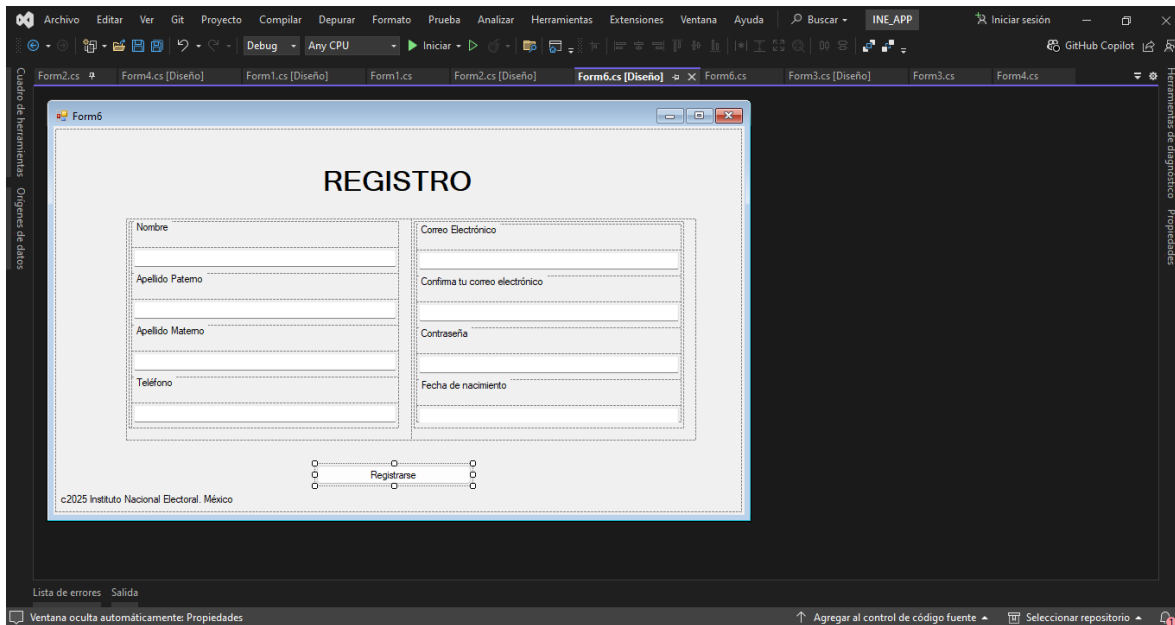


Figure 6 Diseño de Registro

## ♣ Código(Comentado)

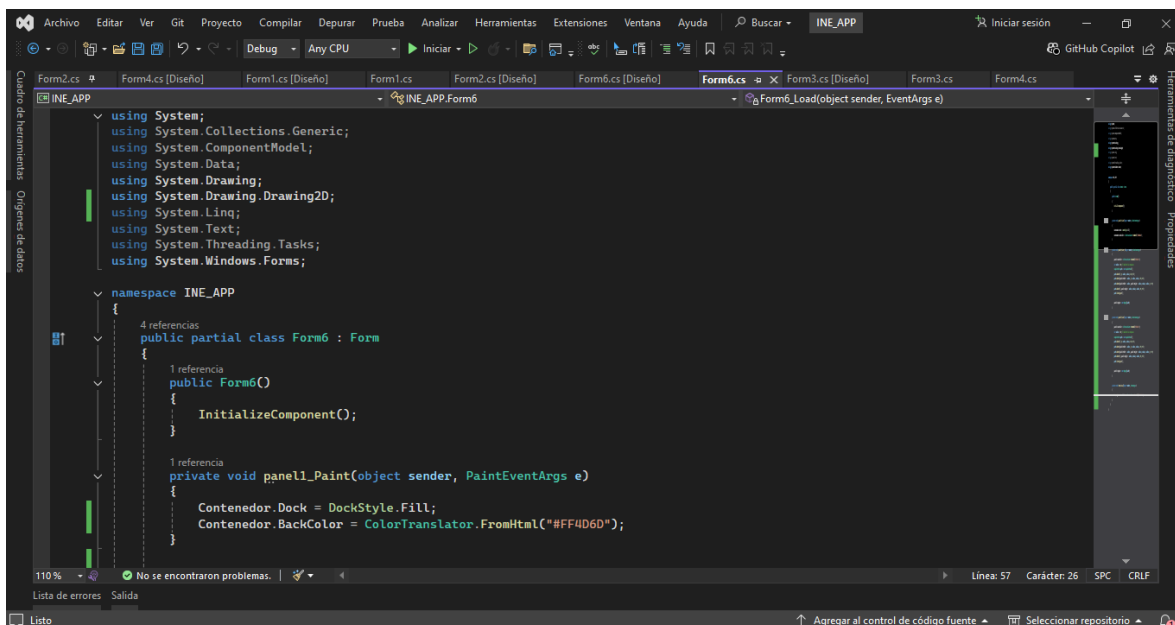
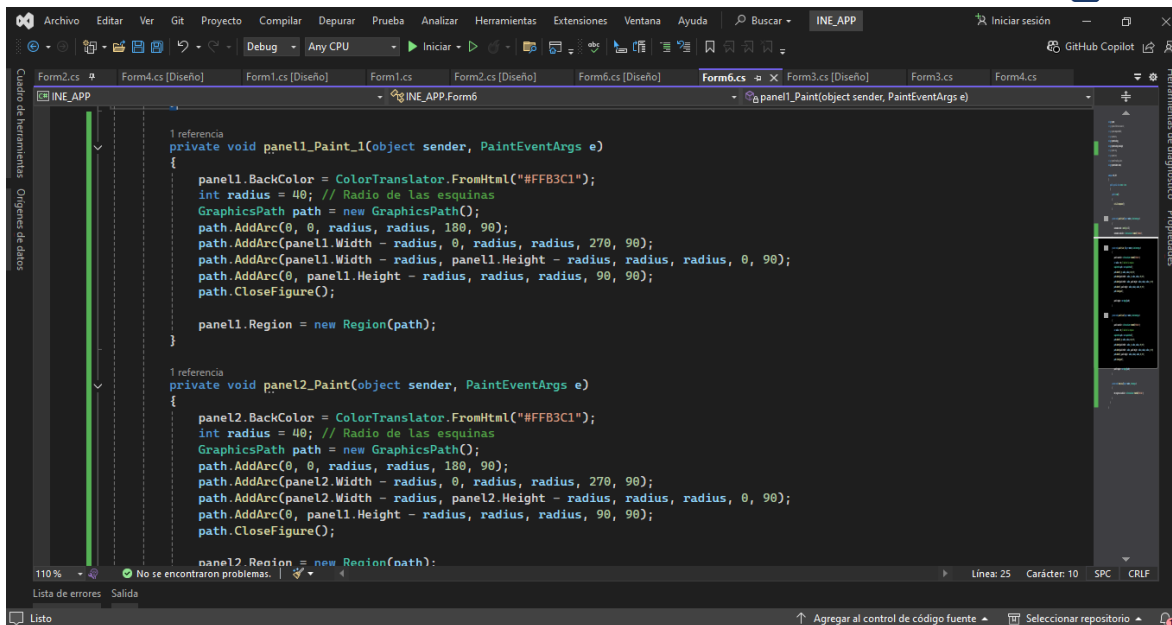


Figure 7 Código del diseño de registro





```

1 referencia
private void panel1_Paint_1(object sender, PaintEventArgs e)
{
    panel1.BackColor = ColorTranslator.FromHtml("#FFB3C1");
    int radius = 40; // Radio de las esquinas
    GraphicsPath path = new GraphicsPath();
    path.AddArc(0, 0, radius, radius, 180, 90);
    path.AddArc(panel1.Width - radius, 0, radius, radius, 270, 90);
    path.AddArc(panel1.Width - radius, panel1.Height - radius, radius, radius, 0, 90);
    path.AddArc(0, panel1.Height - radius, radius, radius, 90, 90);
    path.CloseFigure();

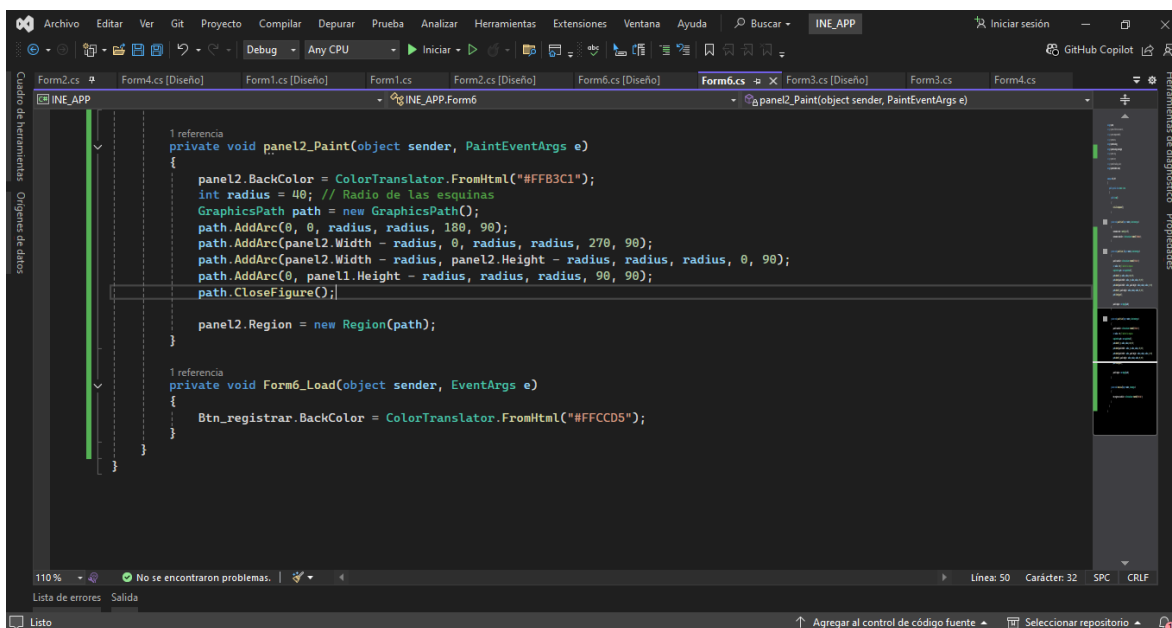
    panel1.Region = new Region(path);
}

1 referencia
private void panel2_Paint(object sender, PaintEventArgs e)
{
    panel2.BackColor = ColorTranslator.FromHtml("#FFB3C1");
    int radius = 40; // Radio de las esquinas
    GraphicsPath path = new GraphicsPath();
    path.AddArc(0, 0, radius, radius, 180, 90);
    path.AddArc(panel2.Width - radius, 0, radius, radius, 270, 90);
    path.AddArc(panel2.Width - radius, panel2.Height - radius, radius, radius, 0, 90);
    path.AddArc(0, panel1.Height - radius, radius, radius, 90, 90);
    path.CloseFigure();

    panel2.Region = new Region(path);
}

```

Figure 8 Código de la Interfaz del diseño de Registro



```

1 referencia
private void panel2_Paint(object sender, PaintEventArgs e)
{
    panel2.BackColor = ColorTranslator.FromHtml("#FFB3C1");
    int radius = 40; // Radio de las esquinas
    GraphicsPath path = new GraphicsPath();
    path.AddArc(0, 0, radius, radius, 180, 90);
    path.AddArc(panel2.Width - radius, 0, radius, radius, 270, 90);
    path.AddArc(panel2.Width - radius, panel2.Height - radius, radius, radius, 0, 90);
    path.AddArc(0, panel1.Height - radius, radius, radius, 90, 90);
    path.CloseFigure();

    panel2.Region = new Region(path);
}

1 referencia
private void Form6_Load(object sender, EventArgs e)
{
    Btn_registrar.BackColor = ColorTranslator.FromHtml("#FFCCD5");
}

```

Figure 9 Código de la Interfaz del diseño de Registro

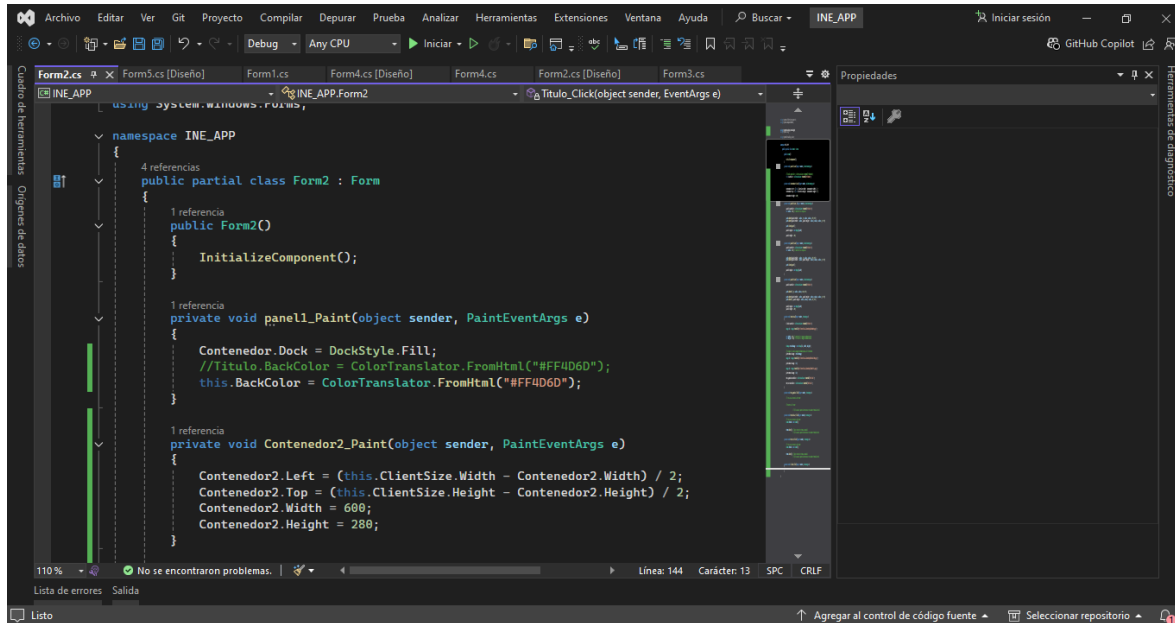
Figure 10 Resultado de la Interfaz del Diseño de Registro

### ♣ Diseño De La Pantalla Principal.

En esta interfaz cuenta con 3 apartados , los cuales son Agendar cita, Ubicación de tu módulo y Trámites.

Figure 11 Interfaz de la Pantalla Principal

## ♣ Código (Comentado)



```

using System.Windows.Forms;

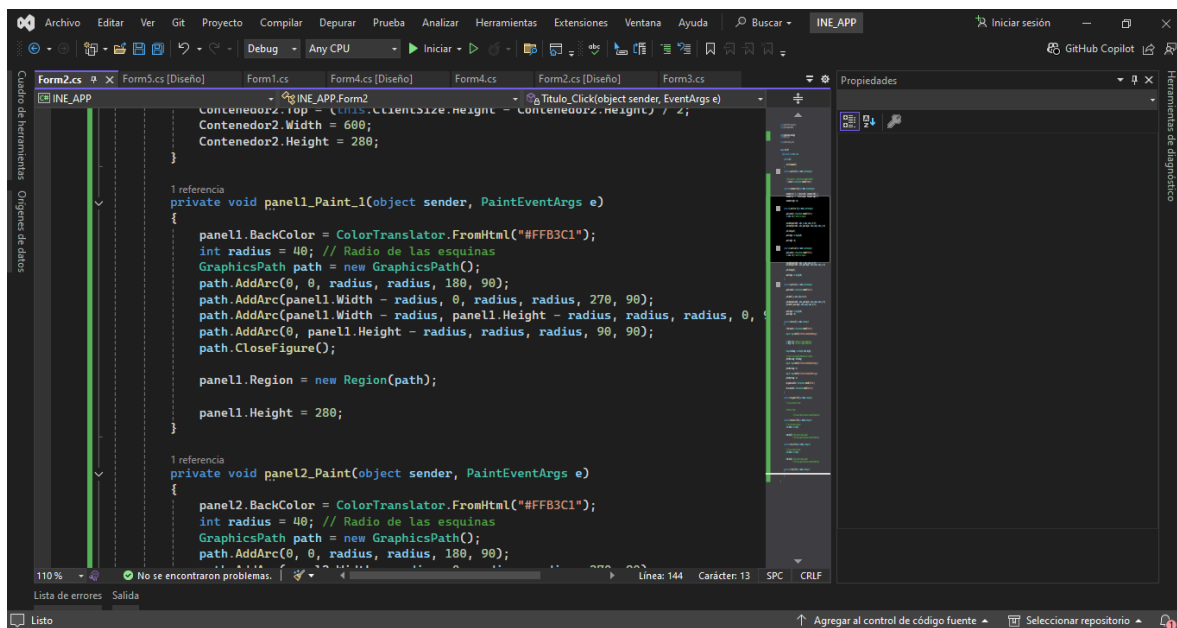
namespace INE_APP
{
    4 referencias
    public partial class Form2 : Form
    {
        1 referencia
        public Form2()
        {
            InitializeComponent();
        }

        1 referencia
        private void panell_Paint(object sender, PaintEventArgs e)
        {
            Contenedor.Dock = DockStyle.Fill;
            //Titulo.BackColor = ColorTranslator.FromHtml("#FF4D6D");
            this.BackColor = ColorTranslator.FromHtml("#FF4D6D");
        }

        1 referencia
        private void Contenedor2_Paint(object sender, PaintEventArgs e)
        {
            Contenedor2.Left = (this.ClientSize.Width - Contenedor2.Width) / 2;
            Contenedor2.Top = (this.ClientSize.Height - Contenedor2.Height) / 2;
            Contenedor2.Width = 600;
            Contenedor2.Height = 280;
        }
    }
}

```

Figure 12 Código de la Interfaz de la pantalla de Principal



```

Contenedor2.Top = (this.ClientSize.Height - Contenedor2.Height) / 2;
Contenedor2.Width = 600;
Contenedor2.Height = 280;
}

1 referencia
private void panell_Paint_1(object sender, PaintEventArgs e)
{
    panell.BackColor = ColorTranslator.FromHtml("#FFB3C1");
    int radius = 40; // Radio de las esquinas
    GraphicsPath path = new GraphicsPath();
    path.AddArc(0, 0, radius, radius, 180, 90);
    path.AddArc(panell.Width - radius, 0, radius, radius, 270, 90);
    path.AddArc(panell.Width - radius, panell.Height - radius, radius, radius, 0, 90);
    path.AddArc(0, panell.Height - radius, radius, radius, 90, 90);
    path.CloseFigure();

    panell.Region = new Region(path);

    panell.Height = 280;
}

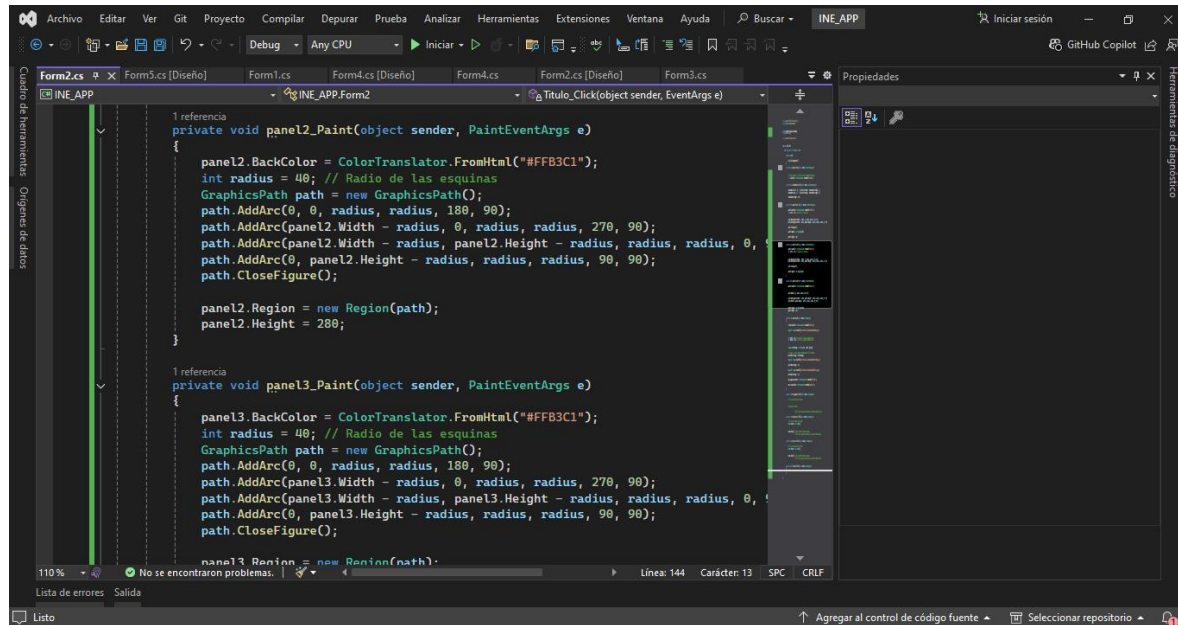
1 referencia
private void panel2_Paint(object sender, PaintEventArgs e)
{
    panel2.BackColor = ColorTranslator.FromHtml("#FFB3C1");
    int radius = 40; // Radio de las esquinas
    GraphicsPath path = new GraphicsPath();
    path.AddArc(0, 0, radius, radius, 180, 90);
    path.AddArc(0, 0, radius, radius, 180, 90);
    path.AddArc(0, 0, radius, radius, 180, 90);
    path.AddArc(0, 0, radius, radius, 180, 90);
    path.CloseFigure();

    panel2.Region = new Region(path);

    panel2.Height = 280;
}
}

```

Figure 13 Código de la Interfaz de la Pantalla Principal



```

1 referencia
private void panel2_Paint(object sender, PaintEventArgs e)
{
    panel2.BackColor = ColorTranslator.FromHtml("#FFB3C1");
    int radius = 40; // Radio de las esquinas
    GraphicsPath path = new GraphicsPath();
    path.AddArc(0, 0, radius, radius, 180, 90);
    path.AddArc(panel2.Width - radius, 0, radius, radius, 270, 90);
    path.AddArc(panel2.Width - radius, panel2.Height - radius, radius, radius, 0, 90);
    path.AddArc(0, panel2.Height - radius, radius, radius, 90, 90);
    path.CloseFigure();

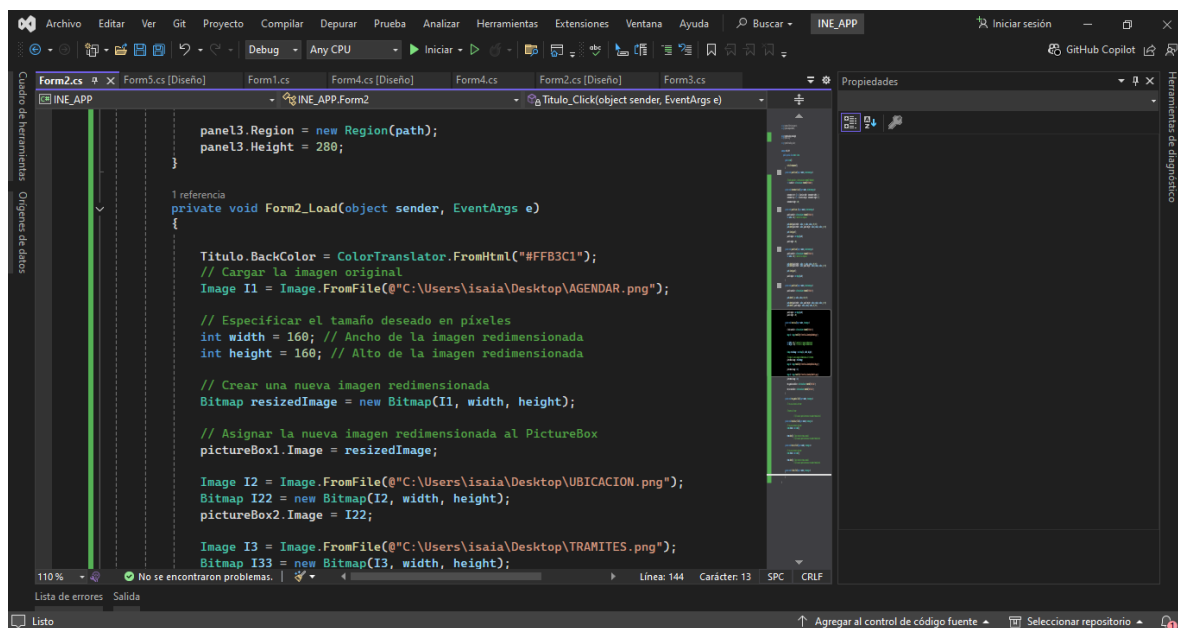
    panel2.Region = new Region(path);
    panel2.Height = 280;
}

1 referencia
private void panel3_Paint(object sender, PaintEventArgs e)
{
    panel3.BackColor = ColorTranslator.FromHtml("#FFB3C1");
    int radius = 40; // Radio de las esquinas
    GraphicsPath path = new GraphicsPath();
    path.AddArc(0, 0, radius, radius, 180, 90);
    path.AddArc(panel3.Width - radius, 0, radius, radius, 270, 90);
    path.AddArc(panel3.Width - radius, panel3.Height - radius, radius, radius, 0, 90);
    path.AddArc(0, panel3.Height - radius, radius, radius, 90, 90);
    path.CloseFigure();

    panel3.Region = new Region(path);
}

```

Figure 14 Código de la Interfaz de la Pantalla de Inicio



```

panel3.Region = new Region(path);
panel3.Height = 280;

1 referencia
private void Form2_Load(object sender, EventArgs e)
{
    Titulo.BackColor = ColorTranslator.FromHtml("#FFB3C1");
    // Cargar la imagen original
    Image I1 = Image.FromFile(@"C:\Users\isaia\Desktop\AGENDAR.png");

    // Especificar el tamaño deseado en pixeles
    int width = 160; // Ancho de la imagen redimensionada
    int height = 160; // Alto de la imagen redimensionada

    // Crear una nueva imagen redimensionada
    Bitmap resizedImage = new Bitmap(I1, width, height);

    // Asignar la nueva imagen redimensionada al PictureBox
    pictureBox1.Image = resizedImage;

    Image I2 = Image.FromFile(@"C:\Users\isaia\Desktop\UBICACION.png");
    Bitmap I22 = new Bitmap(I2, width, height);
    pictureBox2.Image = I22;

    Image I3 = Image.FromFile(@"C:\Users\isaia\Desktop\TRAMITES.png");
    Bitmap I33 = new Bitmap(I3, width, height);
}

```

Figure 15 Código de la Interfaz de la pantalla de Inicio

```

// Muestra el Form2
form3.Show(); // Abre Form2 de forma no modal
// (el usuario puede interactuar con ambos formularios).}

1 referencia
private void Btn_ubicar_Click(object sender, EventArgs e)
{
    // Crea una instancia de Form2
    Form4 form4 = new Form4();

    // Muestra el Form2
    form4.Show(); // Abre Form2 de forma no modal
    // (el usuario puede interactuar con ambos formularios).
}

1 referencia
private void Btn_ver_Click(object sender, EventArgs e)
{
    // Crea una instancia de Form2
    Form5 form5 = new Form5();

    // Muestra el Form2
    form5.Show(); // Abre Form2 de forma no modal
    // (el usuario puede interactuar con ambos formularios).
}

1 referencia
private void Titulo_Click(object sender, EventArgs e)

```

Figure 16 Código de la Interfaz de la pantalla de Inicio

```

Bitmap I22 = new Bitmap(122, width, height);
pictureBox2.Image = I22;

Image I3 = Image.FromFile(@"C:\Users\isaia\Desktop\TRAMITES.png");
Bitmap I33 = new Bitmap(I3, width, height);
pictureBox3.Image = I33;

Btn_agendar.BackColor = ColorTranslator.FromHtml("#FFCCD5");
Btn_ubicar.BackColor = ColorTranslator.FromHtml("#FFCCD5");
Btn_ver.BackColor = ColorTranslator.FromHtml("#FFCCD5");

1 referencia
private void Btn_agendar_Click(object sender, EventArgs e)
{
    // Crea una instancia de Form2
    Form3 form3 = new Form3();

    // Muestra el Form2
    form3.Show(); // Abre Form2 de forma no modal
    // (el usuario puede interactuar con ambos formularios).}

1 referencia
private void Btn_ubicar_Click(object sender, EventArgs e)
{
    // Crea una instancia de Form2
    Form4 form4 = new Form4();

```

Figure 17 Código de la Interfaz de la pantalla Principal

## ♣ Resultados



Figure 18 Diseño final de la pantalla Principal

## ♣ Diseño De Interfaz De Agendar Cita

En este apartado realizamos la interfaz de agendar cita, creando botones respecto a los requisitos.

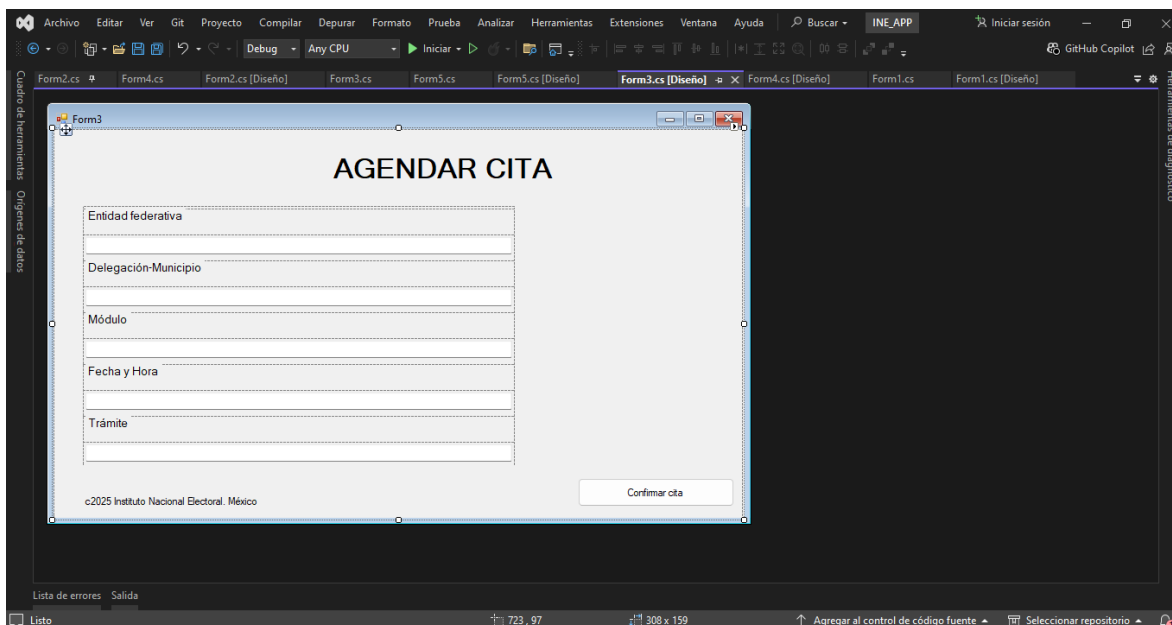
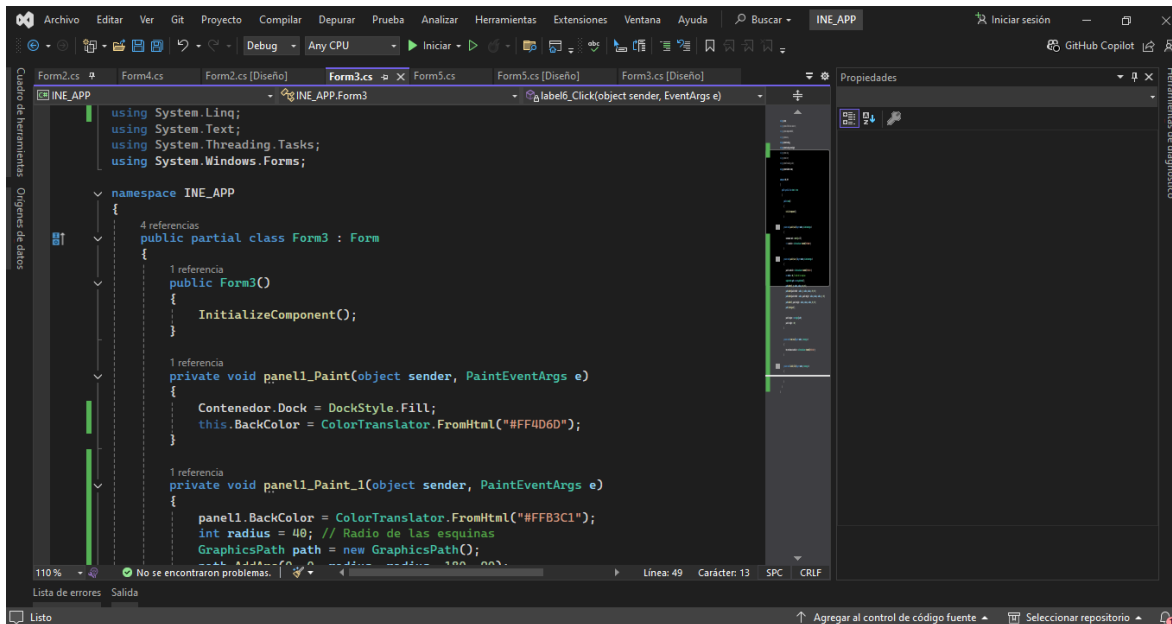


Figure 19 Diseño de la Interfaz de Agendar Cita

## ♣ Código(Comentado)



```

using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace INE_APP
{
    4 referencias
    public partial class Form3 : Form
    {
        1 referencia
        public Form3()
        {
            InitializeComponent();
        }

        1 referencia
        private void panell_Paint(object sender, PaintEventArgs e)
        {
            Contenedor.Dock = DockStyle.Fill;
            this.BackColor = ColorTranslator.FromHtml("#FF4D6D");
        }

        1 referencia
        private void panell_Paint_1(object sender, PaintEventArgs e)
        {
            panell.BackColor = ColorTranslator.FromHtml("#FFB3C1");
            int radius = 40; // Radio de las esquinas
            GraphicsPath path = new GraphicsPath();
            path.AddArc(0, 0, radius, radius, 180, 90);
            path.AddArc(panell.Width - radius, 0, radius, radius, 270, 90);
            path.AddArc(panell.Width - radius, panell.Height - radius, radius, radius, 0, 90);
            path.AddArc(0, panell.Height - radius, radius, radius, 90, 90);
            path.CloseFigure();

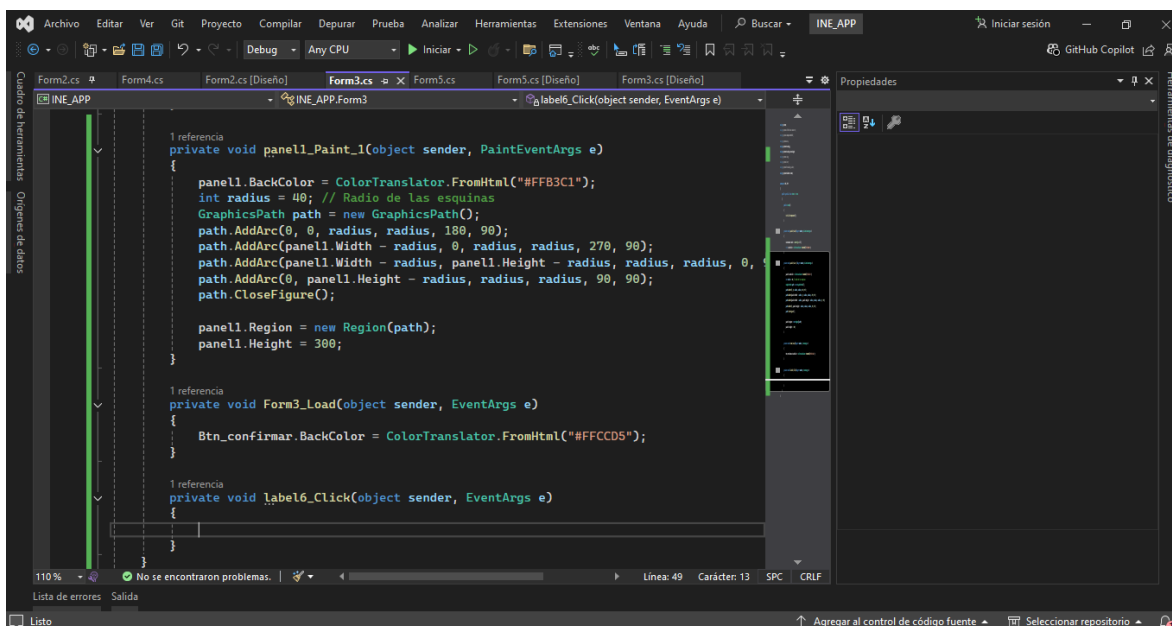
            panell.Region = new Region(path);
            panell.Height = 300;
        }

        1 referencia
        private void Form3_Load(object sender, EventArgs e)
        {
            Btn_confirmar.BackColor = ColorTranslator.FromHtml("#FFCCD5");
        }

        1 referencia
        private void label6_Click(object sender, EventArgs e)
        {
        }
    }
}

```

Figure 20 Código de la Interfaz de Agendar Cita



```

1 referencia
private void panell_Paint_1(object sender, PaintEventArgs e)
{
    panell.BackColor = ColorTranslator.FromHtml("#FFB3C1");
    int radius = 40; // Radio de las esquinas
    GraphicsPath path = new GraphicsPath();
    path.AddArc(0, 0, radius, radius, 180, 90);
    path.AddArc(panell.Width - radius, 0, radius, radius, 270, 90);
    path.AddArc(panell.Width - radius, panell.Height - radius, radius, radius, 0, 90);
    path.AddArc(0, panell.Height - radius, radius, radius, 90, 90);
    path.CloseFigure();

    panell.Region = new Region(path);
    panell.Height = 300;
}

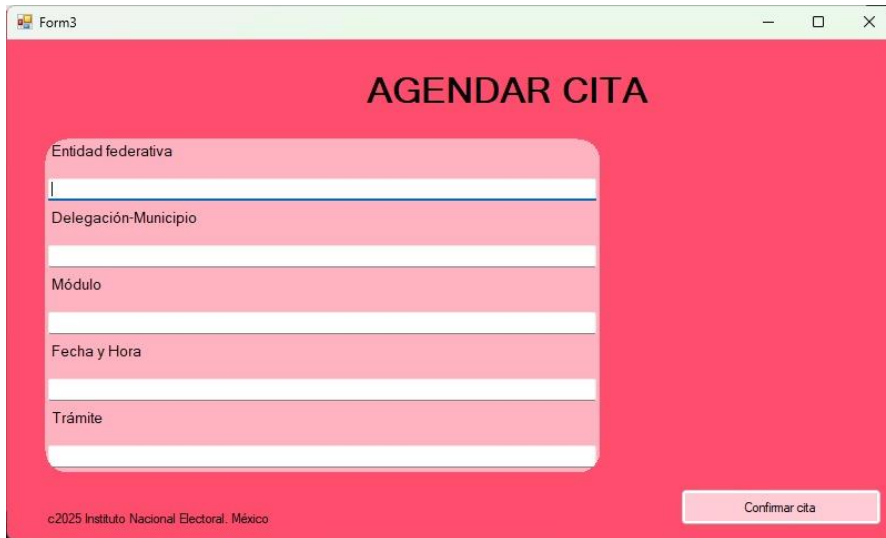
1 referencia
private void Form3_Load(object sender, EventArgs e)
{
    Btn_confirmar.BackColor = ColorTranslator.FromHtml("#FFCCD5");
}

1 referencia
private void label6_Click(object sender, EventArgs e)
{
}

```

Figure 21 Código de la Interfaz de Agendar Cita

## ♣ Resultado

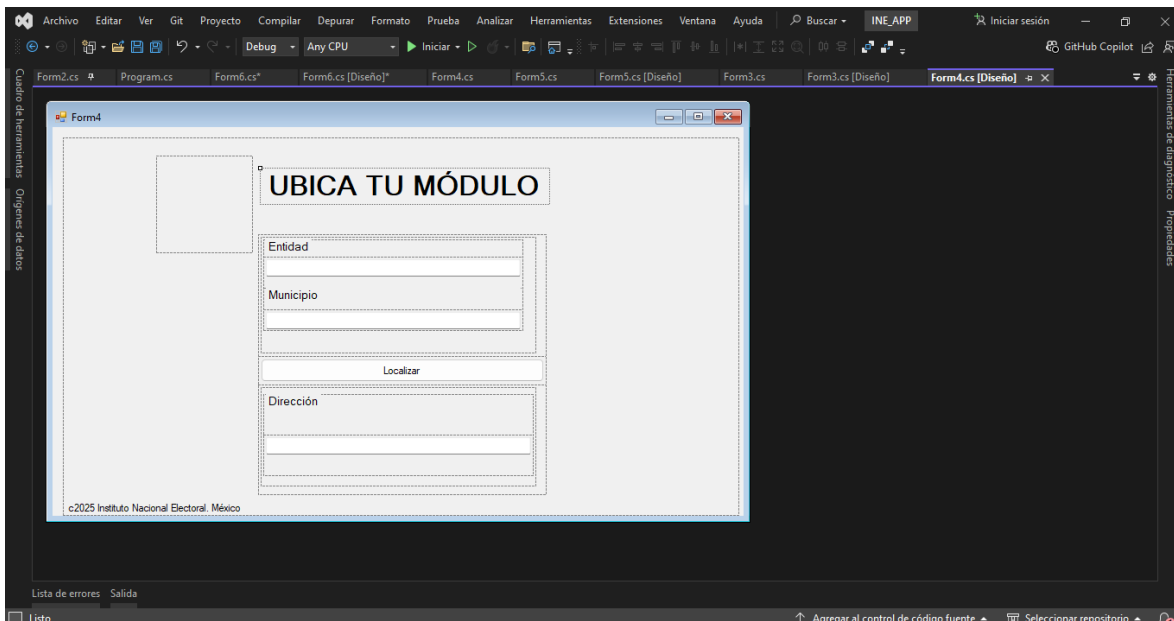


The screenshot shows a web browser window titled "Form3" with a red background. The main heading is "AGENDAR CITA". Below it is a form with the following fields: "Entidad federativa", "Delegación-Municipio", "Módulo", "Fecha y Hora", and "Trámite". Each field is represented by a white input box with a light blue border. At the bottom right of the form is a button labeled "Confirmar cita". In the bottom left corner, there is a small text string: "c2025 Instituto Nacional Electoral. México".

Figure 22 Diseño final de la Interfaz Agendar cita

## ♣ Diseño De Interfaz De Ubicación De Módulos

Se realizó la creación de la ubicación del módulo, en donde el usuario puede ingresar los datos de su municipio y al dar clic al botón de localizar le arroja datos referentes a su búsqueda.

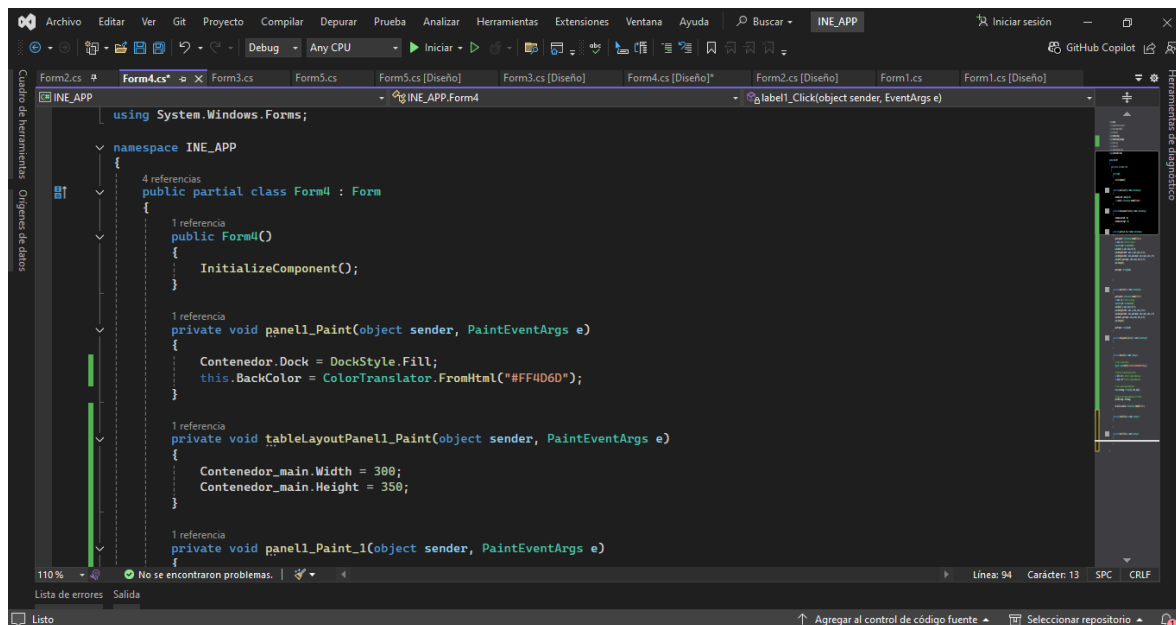


The screenshot shows a Visual Studio Code IDE window with a dark theme. The active file is "Form4.cs [Diseño]". The browser window displays a form titled "UBICA TU MÓDULO". The form has the following fields: "Entidad", "Municipio", "Localizar" (a button), and "Dirección". Each field is represented by a white input box with a light blue border. In the bottom left corner of the form, there is a small text string: "c2025 Instituto Nacional Electoral. México". The IDE interface includes a menu bar at the top with options like "Archivo", "Editar", "Ver", "Git", "Proyecto", "Compilar", "Depurar", "Formato", "Prueba", "Analizar", "Herramientas", "Extensiones", "Ventana", "Ayuda", and "INE\_APP". The status bar at the bottom shows "Listo" and "Agregar al control de código fuente".

Figure 23 Diseño de la Interfaz de Ubicación de Módulos



## ♣ Código(Comentado)



```
using System.Windows.Forms;

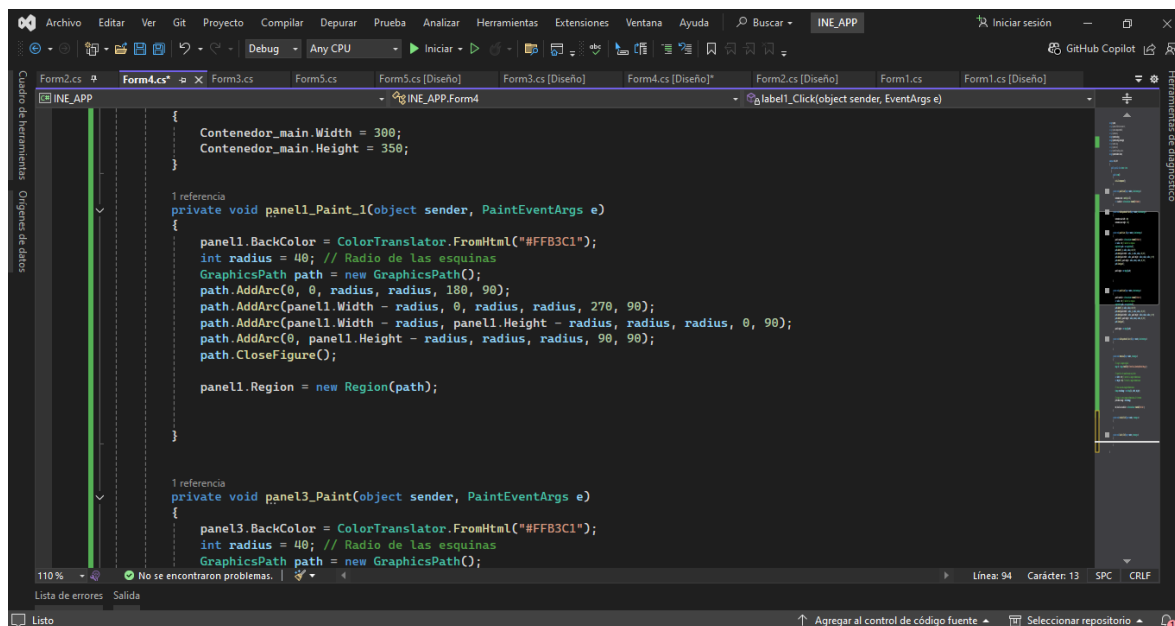
namespace INE_APP
{
    4 referencias
    public partial class Form4 : Form
    {
        1 referencia
        public Form4()
        {
            InitializeComponent();
        }

        1 referencia
        private void panell_Paint(object sender, PaintEventArgs e)
        {
            Contenedor.Dock = DockStyle.Fill;
            this.BackColor = ColorTranslator.FromHtml("#FF4D6D");
        }

        1 referencia
        private void tableLayoutPanell_Paint(object sender, PaintEventArgs e)
        {
            Contenedor_main.Width = 300;
            Contenedor_main.Height = 350;
        }

        1 referencia
        private void panell_Paint_1(object sender, PaintEventArgs e)
        {
        }
    }
}
```

Figure 24 Código de la Interfaz de Ubicar Módulos



```
{
    Contenedor_main.Width = 300;
    Contenedor_main.Height = 350;
}

1 referencia
private void panell_Paint_1(object sender, PaintEventArgs e)
{
    panell.BackColor = ColorTranslator.FromHtml("#FFB3C1");
    int radius = 40; // Radio de las esquinas
    GraphicsPath path = new GraphicsPath();
    path.AddArc(0, 0, radius, radius, 180, 90);
    path.AddArc(panell.Width - radius, 0, radius, radius, 270, 90);
    path.AddArc(panell.Width - radius, panell.Height - radius, radius, radius, 0, 90);
    path.AddArc(0, panell.Height - radius, radius, radius, 90, 90);
    path.CloseFigure();

    panell.Region = new Region(path);
}

1 referencia
private void panel3_Paint(object sender, PaintEventArgs e)
{
    panel3.BackColor = ColorTranslator.FromHtml("#FFB3C1");
    int radius = 40; // Radio de las esquinas
    GraphicsPath path = new GraphicsPath();
}
```

Figure 25 Código de la Interfaz Ubicar Módulos

```

1 referencia
private void panel3_Paint(object sender, PaintEventArgs e)
{
    panel3.BackColor = ColorTranslator.FromHtml("#FFB3C1");
    int radius = 40; // Radio de las esquinas
    GraphicsPath path = new GraphicsPath();
    path.AddArc(0, 0, radius, radius, 180, 90);
    path.AddArc(panel3.Width - radius, 0, radius, radius, 270, 90);
    path.AddArc(panel3.Width - radius, panel3.Height - radius, radius, radius, 0, 90);
    path.AddArc(0, panel3.Height - radius, radius, radius, 90, 90);
    path.CloseFigure();

    panel3.Region = new Region(path);
}

1 referencia
private void tableLayoutPanel1_Paint_1(object sender, PaintEventArgs e)
{
}

1 referencia
private void Form4_Load(object sender, EventArgs e)
{
    // Cargar la imagen original
    Image I1 = Image.FromFile(@"C:\Users\isaia\Desktop\UBICACION.png");

    // Form4_Load: el tamaño deseado en pixels
}

```

Figure 26 Código de la Interfaz de Ubicar Módulos

```

// Especificar el tamaño deseado en pixels
int width = 100; // Ancho de la imagen redimensionada
int height = 100; // Alto de la imagen redimensionada

// Crear una nueva imagen redimensionada
Bitmap resizedImage = new Bitmap(I1, width, height);

// Asignar la nueva imagen redimensionada al PictureBox
pictureBox1.Image = resizedImage;

Btn_localizar.BackColor = ColorTranslator.FromHtml("#FFCCD5");
}

1 referencia
private void Entidad_Click(object sender, EventArgs e)
{
}

1 referencia
private void label1_Click(object sender, EventArgs e)
{
}

```

Figure 27Código de la Interfaz de Ubicar Módulos

## ♣ Resultado

Figure 28 Diseño Final de la Interfaz de Ubica Tu Módulo

## ♣ Diseño De Interfaz De Consulta De Trámites

En esta interfaz creamos el diseño para que el usuario pueda rellenar los campos con sus datos personales para generar sus trámites deseados, creamos botones que al darles clic te mandan al trámite requerido.

Figure 29 Diseño de la Interfaz de Trámites

## ♣ Código(Comentado)

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace INE_APP
{
    4 referencias
    public partial class Form5 : Form
    {
        1 referencia
        public Form5()
        {
            InitializeComponent();
        }

        1 referencia
        private void panel1_Paint(object sender, PaintEventArgs e)
        {
            Contenedor.Dock = DockStyle.Fill;
            Contenedor.BackColor = ColorTranslator.FromHtml(\"#FF4D6D\");
        }
    }
}

```

Figure 29 Código de la Interfaz de Tramites

```

1 referencia
private void panel1_Paint_1(object sender, PaintEventArgs e)
{
    panel1.BackColor = ColorTranslator.FromHtml(\"#FFB3C1\");
    int radius = 40; // Radio de las esquinas
    GraphicsPath path = new GraphicsPath();
    path.AddArc(0, 0, radius, radius, 180, 90);
    path.AddArc(panel1.Width - radius, 0, radius, radius, 270, 90);
    path.AddArc(panel1.Width - radius, panel1.Height - radius, radius, radius, 0, 90);
    path.AddArc(0, panel1.Height - radius, radius, radius, 90, 90);
    path.CloseFigure();

    panel1.Region = new Region(path);
}

1 referencia
private void label5_Click(object sender, EventArgs e)
{
}

1 referencia
private void Form5_Load(object sender, EventArgs e)
{
    // Cargar la imagen original
    Image I1 = Image.FromFile(0\"C:\Users\isaia\Desktop\TRAMITES.png\");

    // Especificar el tamaño deseado en pixeles
}

```

Figure 30 Código de la Interfaz de Trámites

```

Form2.cs Program.cs Form6.cs Form6.cs [Diseño] Form4.cs Form5.cs Form5.cs [Diseño] Form3.cs Form3.cs [Diseño] Form4.cs [Diseño]
INE_APP
Form5_Load(object sender, EventArgs e)
    Image I1 = Image.FromFile(@"C:\Users\isaia\Desktop\TRAMITES.png");

    // Especificar el tamaño deseado en pixeles
    int width = 100; // Ancho de la imagen redimensionada
    int height = 100; // Alto de la imagen redimensionada

    // Crear una nueva imagen redimensionada
    Bitmap resizedImage = new Bitmap(I1, width, height);

    // Asignar la nueva imagen redimensionada al PictureBox
    pictureBox1.Image = resizedImage;
  
```

Figure 31 Código de la Interfaz de Trámites

## ♣ **Resultados**

Credencial Electoral	Renovación de credencial
Obtén tu credencial electoral Documentos: -Acta de nacimiento -Identificación -Comprobante de domicilio	Renueva tu credencial electoral Documentos: -Acta de nacimiento -Identificación -Comprobante de domicilio
Actualizar Domicilio Actualiza tu Domicilio Documentos: -Identificación -Comprobante de domicilio	Corregir Errores Corrige algún dato de tu credencial electoral Documentos: -Acta de nacimiento -Cup -Identificación -Comprobante de domicilio

Figure 32 Resultado del Diseño de Trámites

## RESULTADOS

Form1

### INE

Correo Electrónico

Contraseña

ENTRAR

[Olvidé contraseña](#) [Regístrate](#)

Form6

### REGISTRO

Nombre

Apellido Paterno

Apellido Materno

Teléfono

Correo Electrónico

Confirma tu correo electrónico

Contraseña

Fecha de nacimiento

Regístrate

©2025 Instituto Nacional Electoral, México

Form2

### BIENVENIDO

CITAS

AGENDAR

UBICA TU MÓDULO

UBICAR

TRÁMITES

VER

Form3

### AGENDAR CITA

Entidad federativa

Delegación-Municipio

Módulo

Fecha y Hora

Trámite

Confirmar cita

©2025 Instituto Nacional Electoral, México

Form4

### UBICA TU MÓDULO

Entidad

Municipio

Localizar

Dirección

©2025 Instituto Nacional Electoral, México

Form5

### TRÁMITES

Credencial Electoral

Obtén tu credencial electoral

Documentos:

- Acta de nacimiento
- Identificación
- Comprobante de domicilio

Renovación de credencial

Renueva tu credencial electoral

Documentos:

- Acta de nacimiento
- Identificación
- Comprobante de domicilio

Actualizar Domicilio

Actualiza tu Domicilio

Documentos:

- Identificación
- Comprobante de domicilio

Corregir Errores

Corrige algún dato de tu credencial electoral

Documentos:

- Acta de nacimiento
- Cup
- Identificación
- Comprobante de domicilio

## CONCLUSIÓN

El desarrollo de esta práctica en Visual Studio ha sido una experiencia que permitió comprender el proceso de diseño y creación de una aplicación con interfaz gráfica. A lo largo de la práctica, se aplican conocimientos teóricos y prácticos en la configuración del entorno de desarrollo, el uso de herramientas, la estructuración de proyectos y la implementación de código eficiente en C# (C Sharp).

Uno de los aspectos de la práctica fue la creación de la interfaz gráfica de usuario utilizando Windows Forms, lo que permitió explorar conceptos clave de UI/UX para mejorar la interacción y experiencia del usuario. Se trabajó en la disposición de elementos, la personalización de controles y la aplicación de estilos para lograr una interfaz intuitiva y funcional.

Además, se implementaron diversas funcionalidades mediante la programación orientada a eventos, permitiendo que los usuarios interactúen con la aplicación de manera dinámica. Se programaron botones, cuadros de texto, listas desplegables y otros elementos que respondían a acciones del usuario, facilitando la manipulación de datos y la ejecución de operaciones específicas dentro de la aplicación.

Asimismo, se hizo énfasis la prueba del código, utilizando las herramientas de Visual Studio para detectar y corregir errores. Este proceso ayudó a mejorar la calidad del software, asegurando su correcto funcionamiento y optimizando su rendimiento.