

DEPARTAMENTO: Ciencias de la Ingeniería
CARRERA: Sistemas de Información
CURSO: Séptimo **PARALELO:** "A"
ASIGNATURA: Plataformas de Desarrollo 1

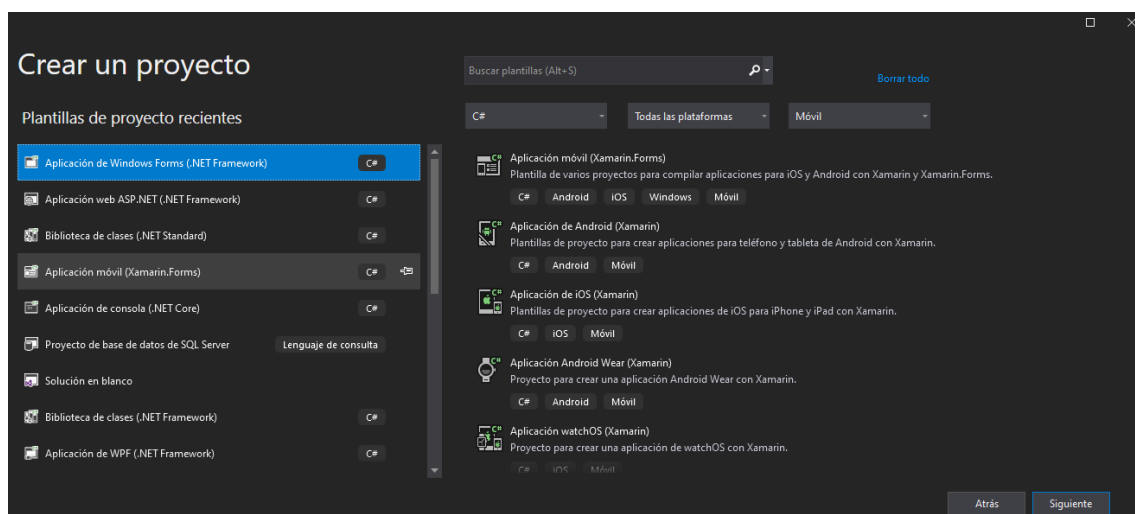
PROFESOR: Mg. Luis Fernando Aguas B.
ESTUDIANTE: Marco Antonio Ayala Lituma
DESCRIPCIÓN: Tarea 1-S7

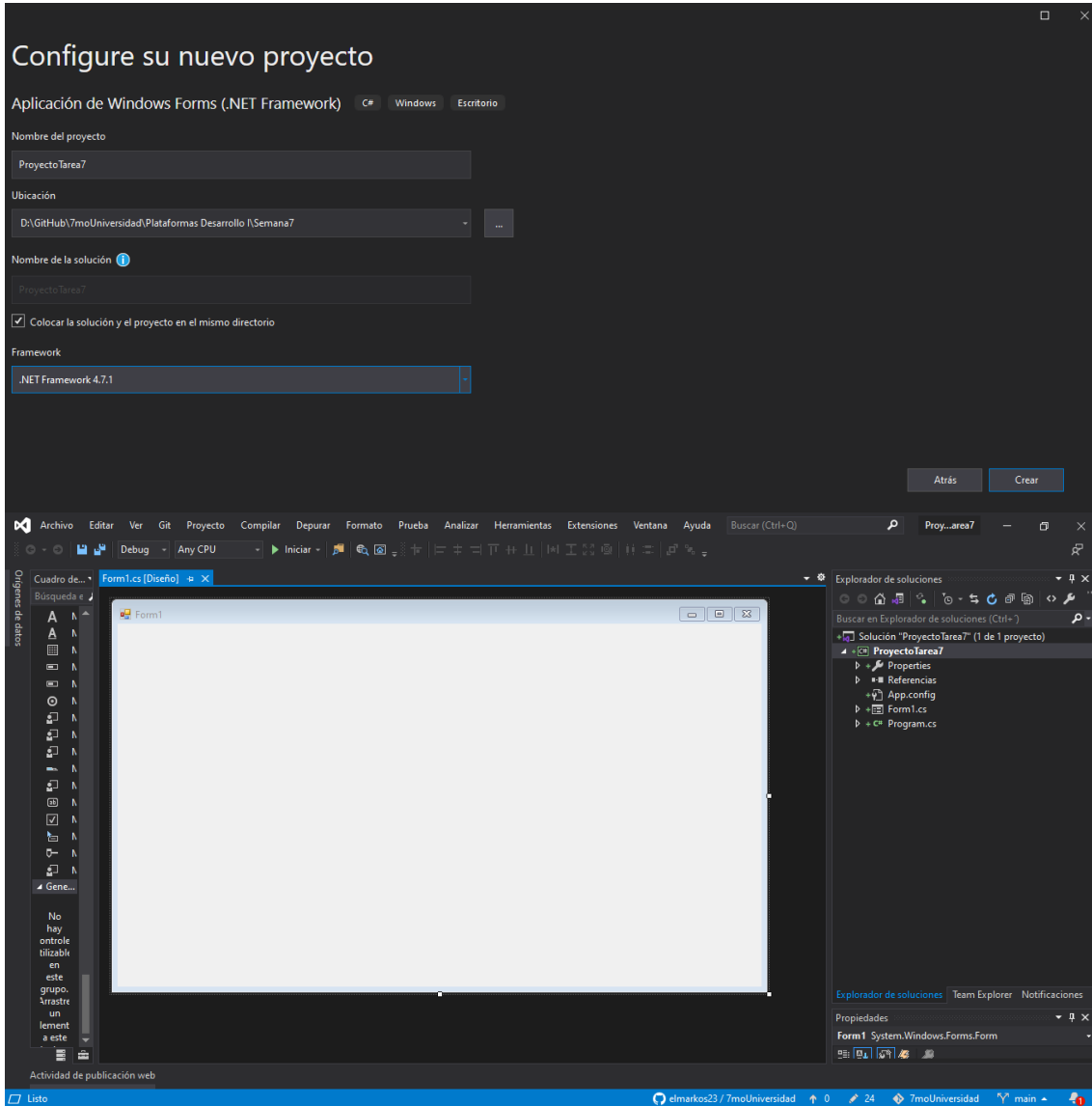
TEMA: Sistemas de Tutorías

Realice un programa en el cual implemente un sistema de tutorías, el cual permita ingresar un tutor, asignarle un estudiante, registrar la reunión de tutorías, registrar las actividades extracurriculares, registrar los acuerdos que se lleguen. El programa debe estar hecho con:

- Estándares de codificación.
- Mínimo 2 Gráficos estadísticos
- Clases
- Estándares de interfases

DESARROLLO:





Agregar nuevo elemento - ProyectoTarea7

Ordenar por: Predeterminado

Buscar (Ctrl+E)

Elementos de Visual C#

- General
- Web
- Windows Forms
- Prism
- SQL Server
- Storm Items
- Xamarin.Forms
- WPF
- Datos
- Código
- Gráficos

En línea

Icono	Nombre	Tipo
Clase	Clase	Elementos de Visual C#
Página con pestañas	Página con pestañas	Elementos de Visual C#
Página de contenido	Página de contenido	Elementos de Visual C#
Página de contenido (C#)	Página de contenido (C#)	Elementos de Visual C#
Página de maestro y detalles	Página de maestro y detalles	Elementos de Visual C#
Página de vista de lista	Página de vista de lista	Elementos de Visual C#
Vista de contenido	Vista de contenido	Elementos de Visual C#
Vista de contenido (C#)	Vista de contenido (C#)	Elementos de Visual C#
Class for U-SQL	Class for U-SQL	Elementos de Visual C#
Interfaz	Interfaz	Elementos de Visual C#
Formulario (Windows Forms)	Formulario (Windows Forms)	Elementos de Visual C#
Celda de vista	Celda de vista	Elementos de Visual C#

Tipo: Elementos de Visual C#
Definición de clase vacía

Nombre:

Agregar Cancelar

Estudiante.cs* Persona.cs Form1.cs [Diseño]

C# ProyectoTarea7 ProyectoTarea7.Modelos.Persona

```
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace ProyectoTarea7.Modelos
8 {
9     1 referencia | 0 cambios | 0 autores, 0 cambios
10     public class Persona
11     {
12         0 referencias | 0 cambios | 0 autores, 0 cambios
13         public string identificacion { get; set; }
14         0 referencias | 0 cambios | 0 autores, 0 cambios
15         public string nombres { get; set; }
16         0 referencias | 0 cambios | 0 autores, 0 cambios
17         public string apellidos { get; set; }
18     }
19 }
```



Agregar nuevo elemento - ProyectoTarea7

Ordenar por: Predeterminado

Buscar (Ctrl+E)

Elementos de Visual C#

- General
- Web
- Windows Forms
- Prism
- SQL Server
- Storm Items
- Xamarin.Forms
- WPF
- Datos
- Código
- Gráficos
- En línea

Elemento	Tipo
Clase	Elementos de Visual C#
Página con pestañas	Elementos de Visual C#
Página de contenido	Elementos de Visual C#
Página de contenido (C#)	Elementos de Visual C#
Página de maestro y detalles	Elementos de Visual C#
Página de vista de lista	Elementos de Visual C#
Vista de contenido	Elementos de Visual C#
Vista de contenido (C#)	Elementos de Visual C#
Class for U-SQL	Elementos de Visual C#
Interfaz	Elementos de Visual C#
Formulario (Windows Forms)	Elementos de Visual C#
Celda de vista	Elementos de Visual C#

Tipo: Elementos de Visual C#
Definición de clase vacía

Nombre: Estudiante

Agregar Cancelar

Estudiante.cs Persona.cs Form1.cs [Diseño]

ProyectoTarea7 ProyectoTarea7.Modelos.Estudiante

```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace ProyectoTarea7.Modelos
8 {
9     0 referencias | 0 cambios | 0 autores, 0 cambios
10     public class Estudiante : Persona
11     {
12         0 referencias | 0 cambios | 0 autores, 0 cambios
13         public string carrera { get; set; }
14         0 referencias | 0 cambios | 0 autores, 0 cambios
15         public string nivel { get; set; }
16     }
17 }

```

Agregar nuevo elemento - ProyectoTarea7

Ordenar por: Predeterminado

Buscar (Ctrl+E)

Elementos de Visual C#

- General
- Web
- Windows Forms
- Prism
- SQL Server
- Storm Items
- Xamarin.Forms
- WPF
- Datos
- Código
- Gráficos

En línea

Icono	Nombre	Elementos de Visual C#
Clase	Clase	Elementos de Visual C#
Página con pestañas	Página con pestañas	Elementos de Visual C#
Página de contenido	Página de contenido	Elementos de Visual C#
Página de contenido (C#)	Página de contenido (C#)	Elementos de Visual C#
Página de maestro y detalles	Página de maestro y detalles	Elementos de Visual C#
Página de vista de lista	Página de vista de lista	Elementos de Visual C#
Vista de contenido	Vista de contenido	Elementos de Visual C#
Vista de contenido (C#)	Vista de contenido (C#)	Elementos de Visual C#
Class for U-SQL	Class for U-SQL	Elementos de Visual C#
Interfaz	Interfaz	Elementos de Visual C#
Formulario (Windows Forms)	Formulario (Windows Forms)	Elementos de Visual C#
Celda de vista	Celda de vista	Elementos de Visual C#

Tipo: Elementos de Visual C#
Definición de clase vacía

Nombre: Tutor

Agregar Cancelar

Tutor.cs

C# ProyectoTarea7

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace ProyectoTarea7.Modelos
8 {
9     2 referencias | 0 cambios | 0 autores, 0 cambios
10     public class Tutor : Persona
11     {
12         4 referencias | 0 cambios | 0 autores, 0 cambios
13         public string facultad { get; set; }
14         4 referencias | 0 cambios | 0 autores, 0 cambios
15         public string disponibilidad { get; set; }
16     }
17 }
```



Agregar nuevo elemento - ProyectoTarea7

Ordenar por: Predeterminado

Buscar (Ctrl+E)

Elementos de Visual C#

- General
- Web
- Windows Forms
- Prism
- SQL Server
- Storm Items
- Xamarin.Forms
- WPF
- Datos
- Código
- Gráficos
- En línea

Clase

Página con pestañas

Página de contenido

Página de contenido (C#)

Página de maestro y detalles

Página de vista de lista

Vista de contenido

Vista de contenido (C#)

Class for U-SQL

Interfaz

Formulario (Windows Forms)

Celda de vista

Tipo: Elementos de Visual C#

Definición de clase vacía

Nombre: AsignacionEstudiante

Agregar Cancelar

AsignacionEstudiante.cs Tutor.cs

ProyectoTarea7 ProyectoTarea7.Modelos.AsignacionEstudiante

```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace ProyectoTarea7.Modelos
8 {
9     2 referencias | 0 cambios | 0 autores, 0 cambios
10     public class AsignacionEstudiante
11     {
12         1 referencia | 0 cambios | 0 autores, 0 cambios
13         public int idAsignacion { get; set; }
14         4 referencias | 0 cambios | 0 autores, 0 cambios
15         public string identificacionTutor { get; set; }
16         1 referencia | 0 cambios | 0 autores, 0 cambios
17         public string nombresTutor { get; set; }
18         2 referencias | 0 cambios | 0 autores, 0 cambios
19         public string identificacionEstudiante { get; set; }
20         1 referencia | 0 cambios | 0 autores, 0 cambios
21         public string nombresEstudiante { get; set; }
22         1 referencia | 0 cambios | 0 autores, 0 cambios
23         public DateTime fechaAsignacion { get; set; }
24     }
25 }

```



Agregar nuevo elemento - ProyectoTarea7

Ordenar por: Predeterminado

Buscar (Ctrl+E)

Elementos de Visual C#

- General
- Web
- Windows Forms
- Prism
- SQL Server
- Storm Items
- Xamarin.Forms
- WPF
- Datos
- Código
- Gráficos
- En línea

Icono	Nombre	Tipo
	Clase	Elementos de Visual C#
	Página con pestañas	Elementos de Visual C#
	Página de contenido	Elementos de Visual C#
	Página de contenido (C#)	Elementos de Visual C#
	Página de maestro y detalles	Elementos de Visual C#
	Página de vista de lista	Elementos de Visual C#
	Vista de contenido	Elementos de Visual C#
	Vista de contenido (C#)	Elementos de Visual C#
	Class for U-SQL	Elementos de Visual C#
	Interfaz	Elementos de Visual C#
	Formulario (Windows Forms)	Elementos de Visual C#
	Celda de vista	Elementos de Visual C#

Tipo: Elementos de Visual C#
Definición de clase vacía

Nombre:

Agregar Cancelar

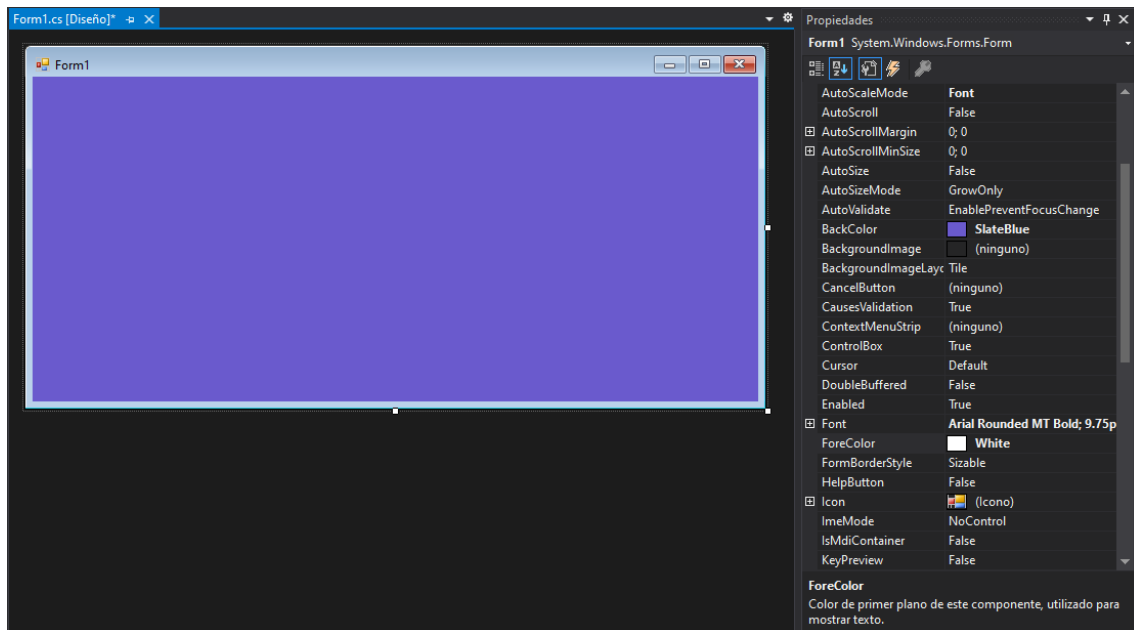
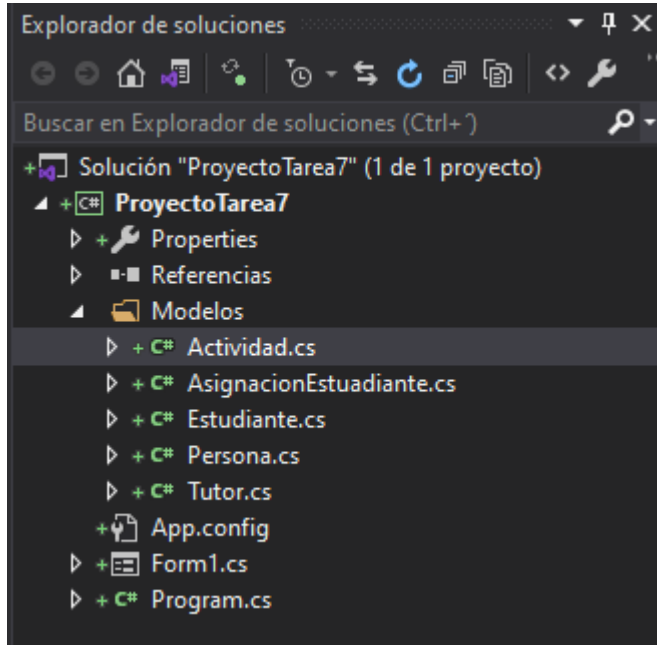
Actividad.cs | AsignacionEstudiante.cs | Tutor.cs

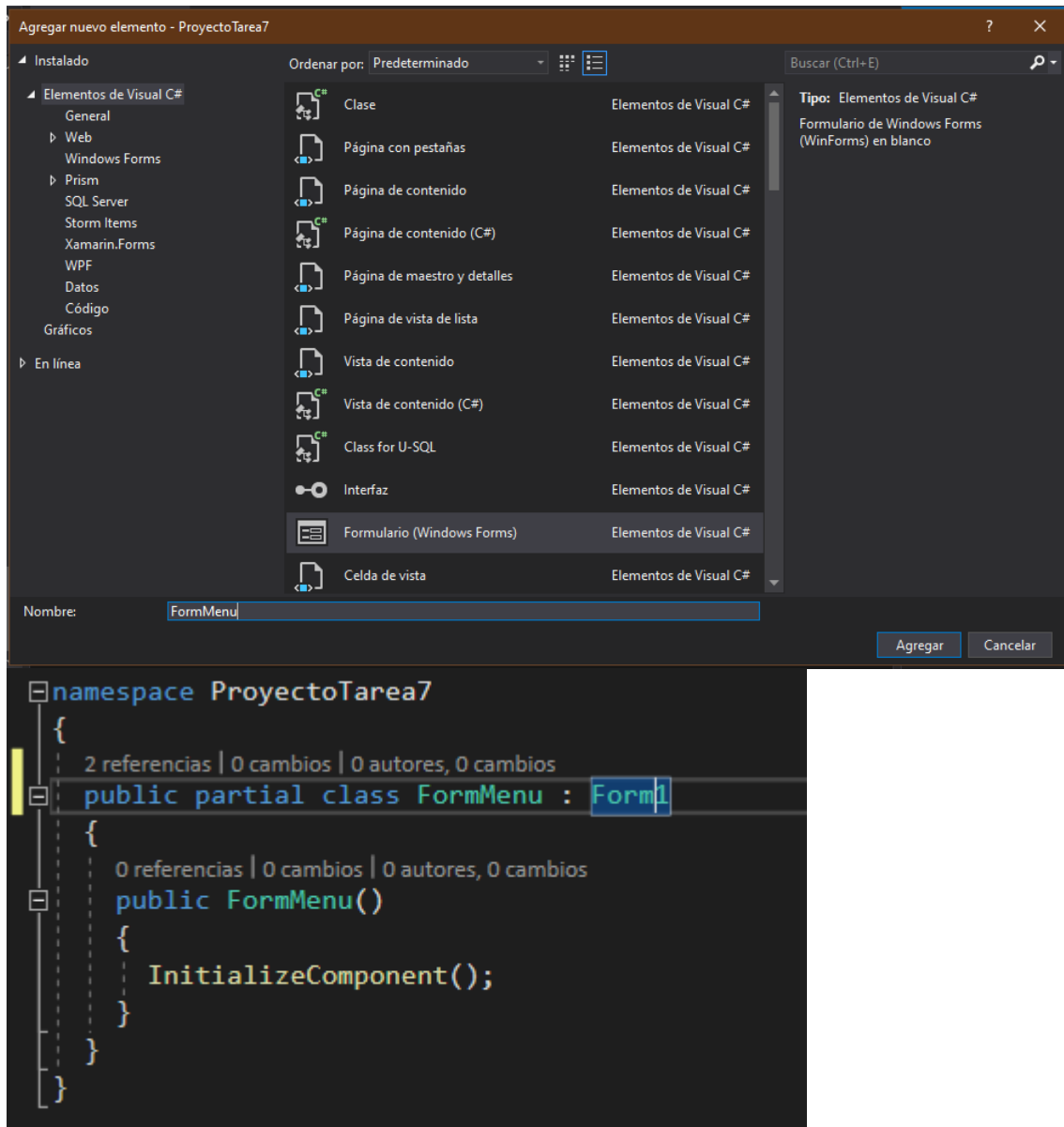
C# ProyectoTarea7

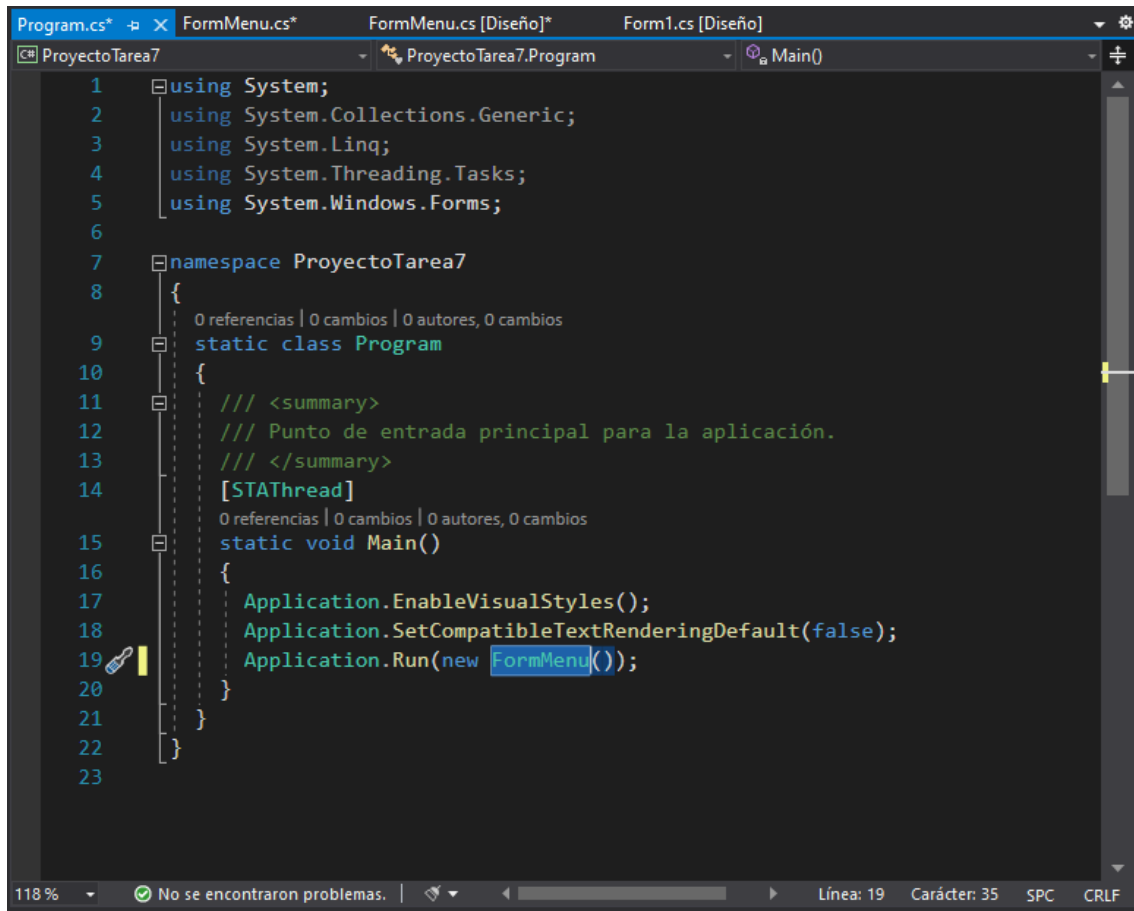
```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace ProyectoTarea7.Modelos
8 {
9     public class Actividad
10    {
11        public string identificacionTutor { get; set; }
12        public string identificacionEstudiante { get; set; }
13        public string nombreActividad { get; set; }
14        public string detalleActividad { get; set; }
15        public DateTime fechaActividad { get; set; }
16        public string estadoActividad { get; set; }
17    }
18 }
19

```

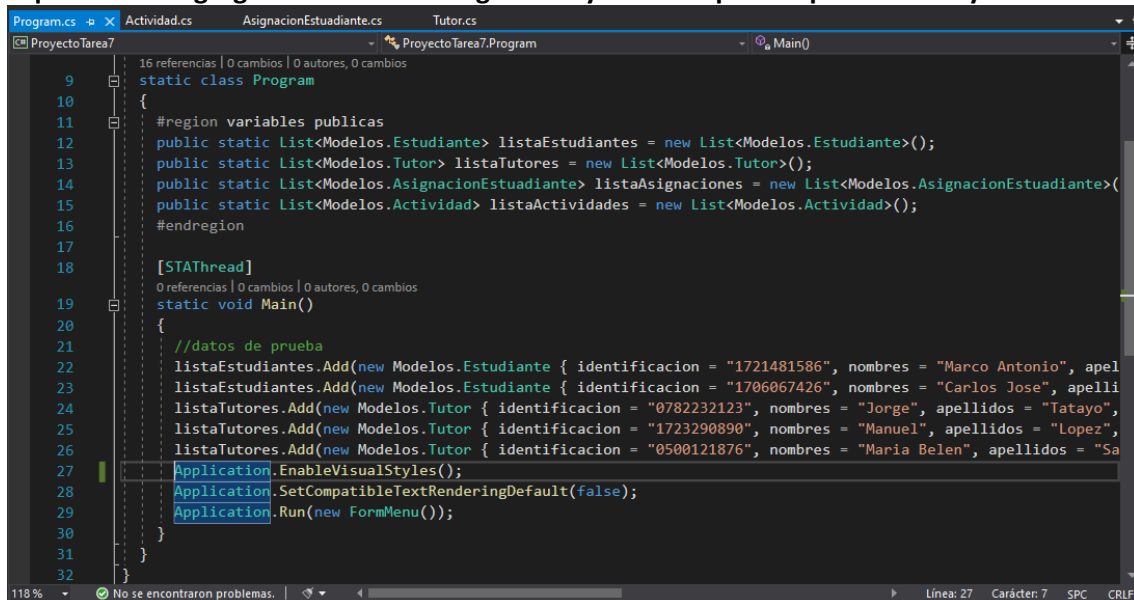




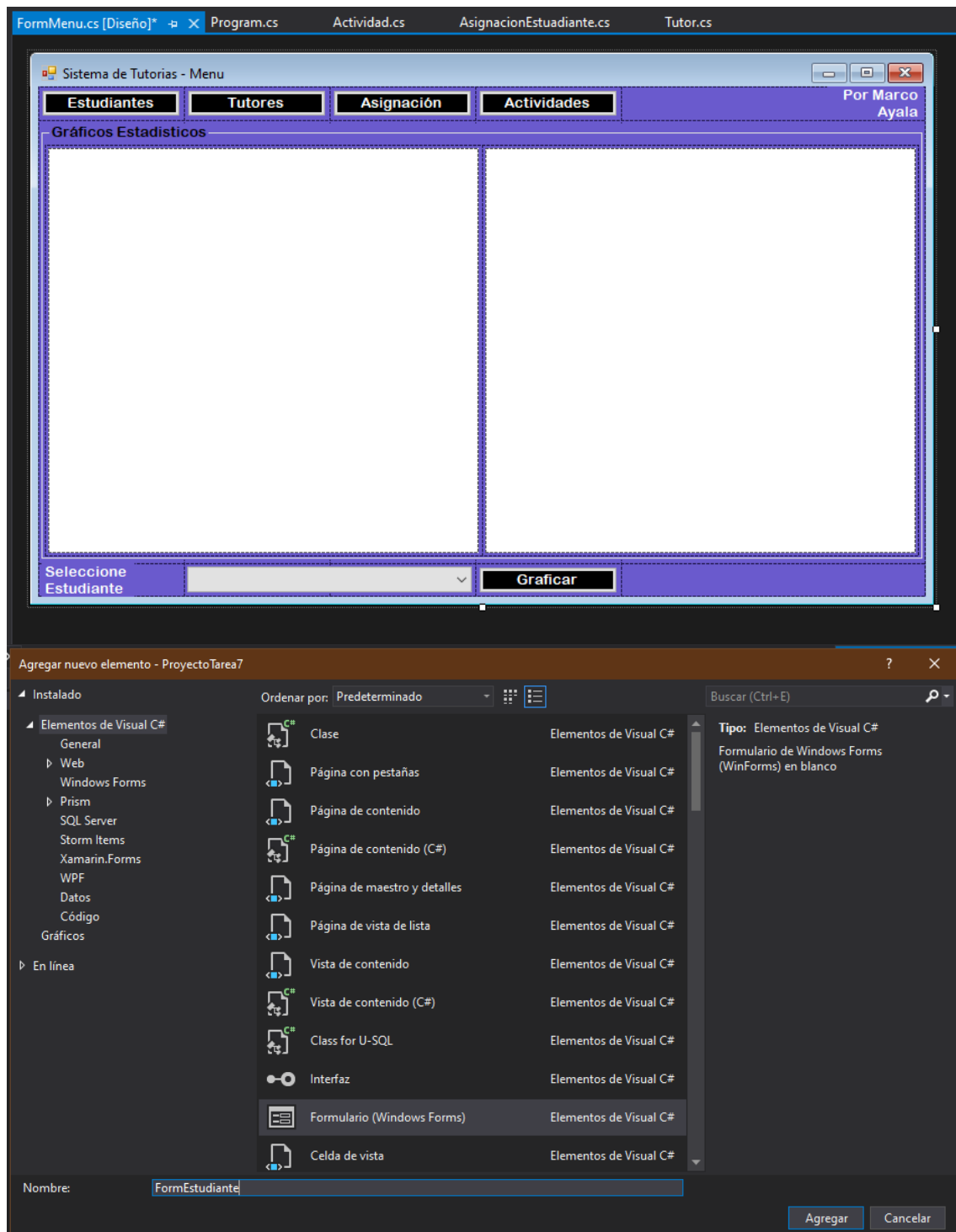


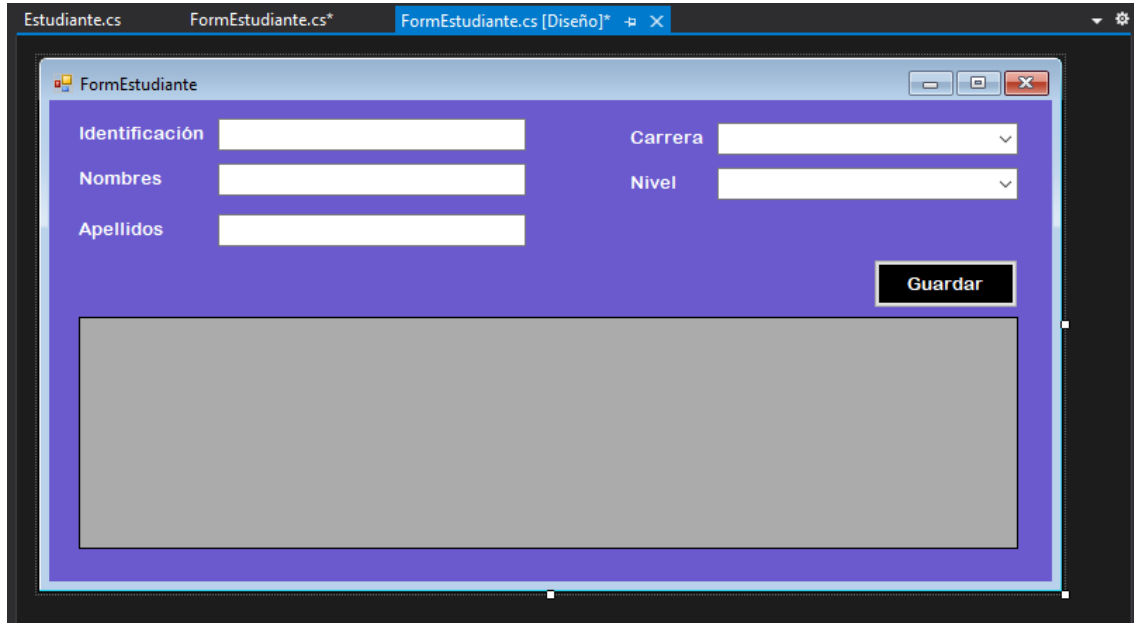
```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5 using System.Windows.Forms;
6
7 namespace ProyectoTarea7
8 {
9     0 referencias | 0 cambios | 0 autores, 0 cambios
10     static class Program
11     {
12         /// <summary>
13         /// Punto de entrada principal para la aplicación.
14         /// </summary>
15         [STAThread]
16         0 referencias | 0 cambios | 0 autores, 0 cambios
17         static void Main()
18         {
19             Application.EnableVisualStyles();
20             Application.SetCompatibleTextRenderingDefault(false);
21             Application.Run(new FormMenu());
22         }
23     }
```

Aquí también agregamos las variables globales y datos de prueba para tutores y estudiantes



```
9 static class Program
10 {
11     #region variables publicas
12     public static List<Modelos.Estudiante> listaEstudiantes = new List<Modelos.Estudiante>();
13     public static List<Modelos.Tutor> listaTutores = new List<Modelos.Tutor>();
14     public static List<Modelos.AsignacionEstudiante> listaAsignaciones = new List<Modelos.AsignacionEstudiante>();
15     public static List<Modelos.Actividad> listaActividades = new List<Modelos.Actividad>();
16     #endregion
17
18     [STAThread]
19     0 referencias | 0 cambios | 0 autores, 0 cambios
20     static void Main()
21     {
22         //datos de prueba
23         listaEstudiantes.Add(new Modelos.Estudiante { identificacion = "1721481586", nombres = "Marco Antonio", apeli
24         listaEstudiantes.Add(new Modelos.Estudiante { identificacion = "1706067426", nombres = "Carlos Jose", apelli
25         listaTutores.Add(new Modelos.Tutor { identificacion = "0782232123", nombres = "Jorge", apellidos = "Tatayo",
26         listaTutores.Add(new Modelos.Tutor { identificacion = "1723290890", nombres = "Manuel", apellidos = "Lopez",
27         listaTutores.Add(new Modelos.Tutor { identificacion = "0500121876", nombres = "Maria Belen", apellidos = "Sa
28         Application.EnableVisualStyles();
29         Application.SetCompatibleTextRenderingDefault(false);
30         Application.Run(new FormMenu());
31     }
32 }
```





Editor de colección de cadenas

Escriba las cadenas en la colección (una por línea):

Primero
Segundo
Tercero
Cuarto
Quinto
Sexto
Septimo
Octavo
Noveno

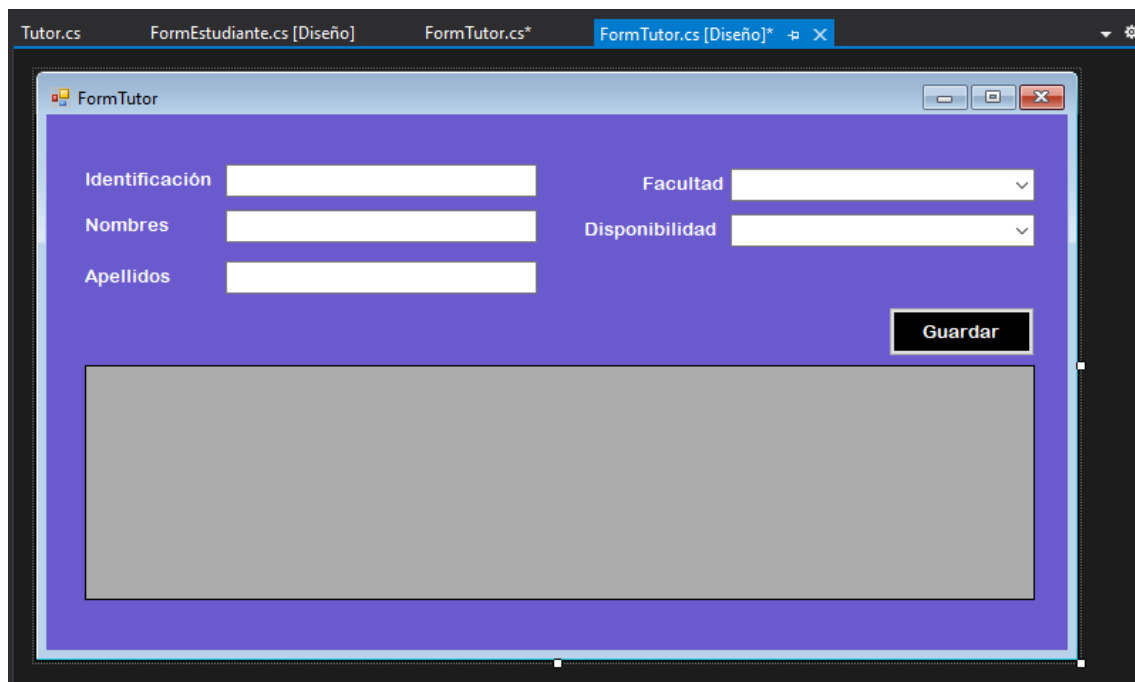
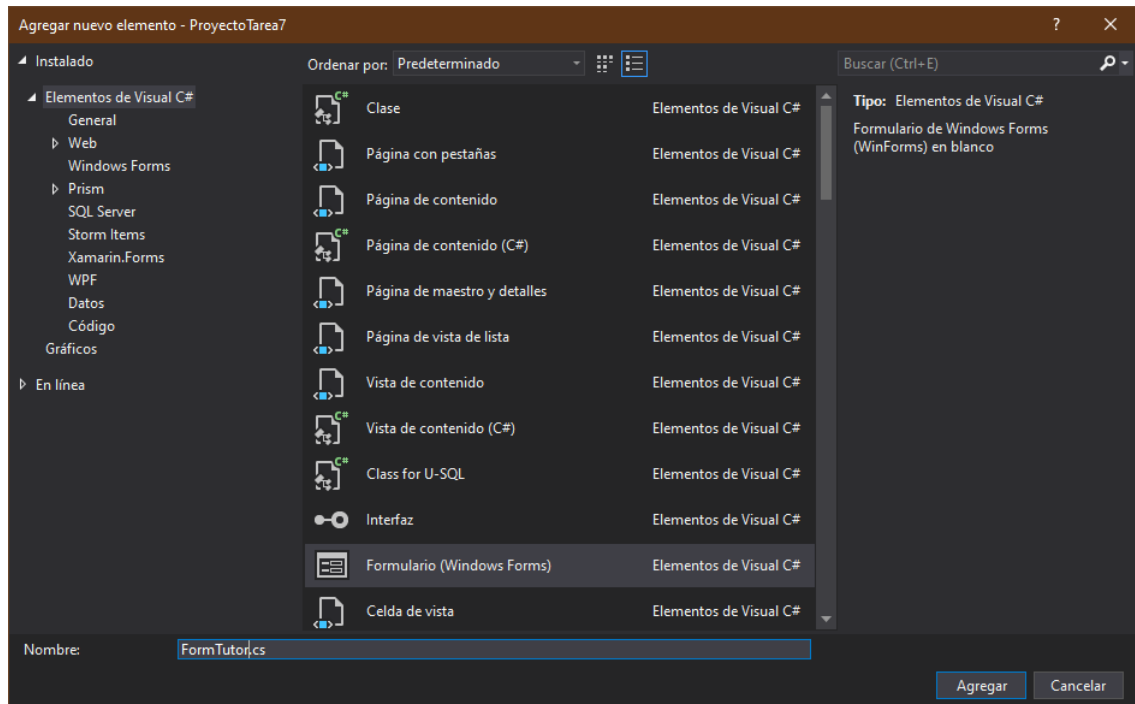
Aceptar Cancelar

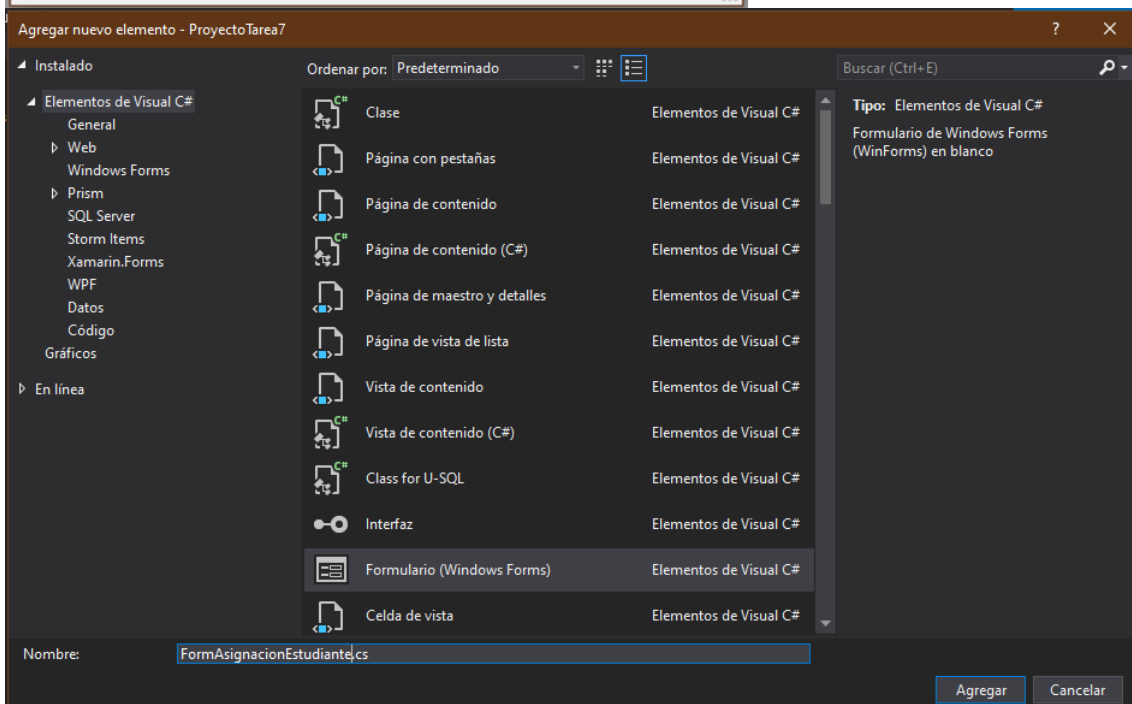
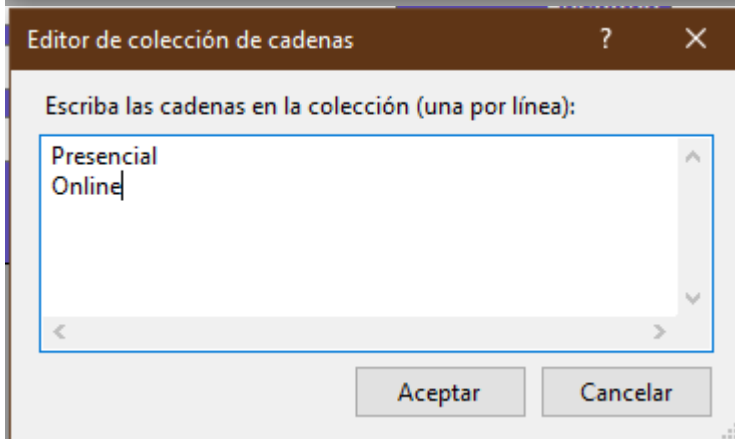
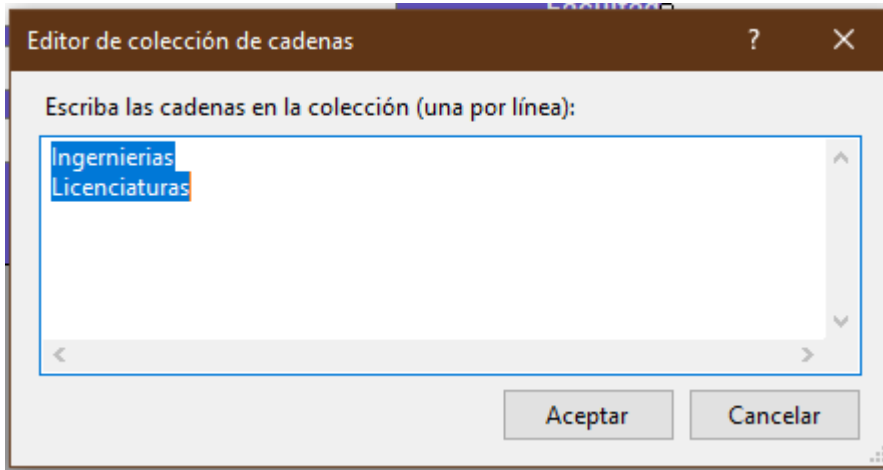
Editor de colección de cadenas

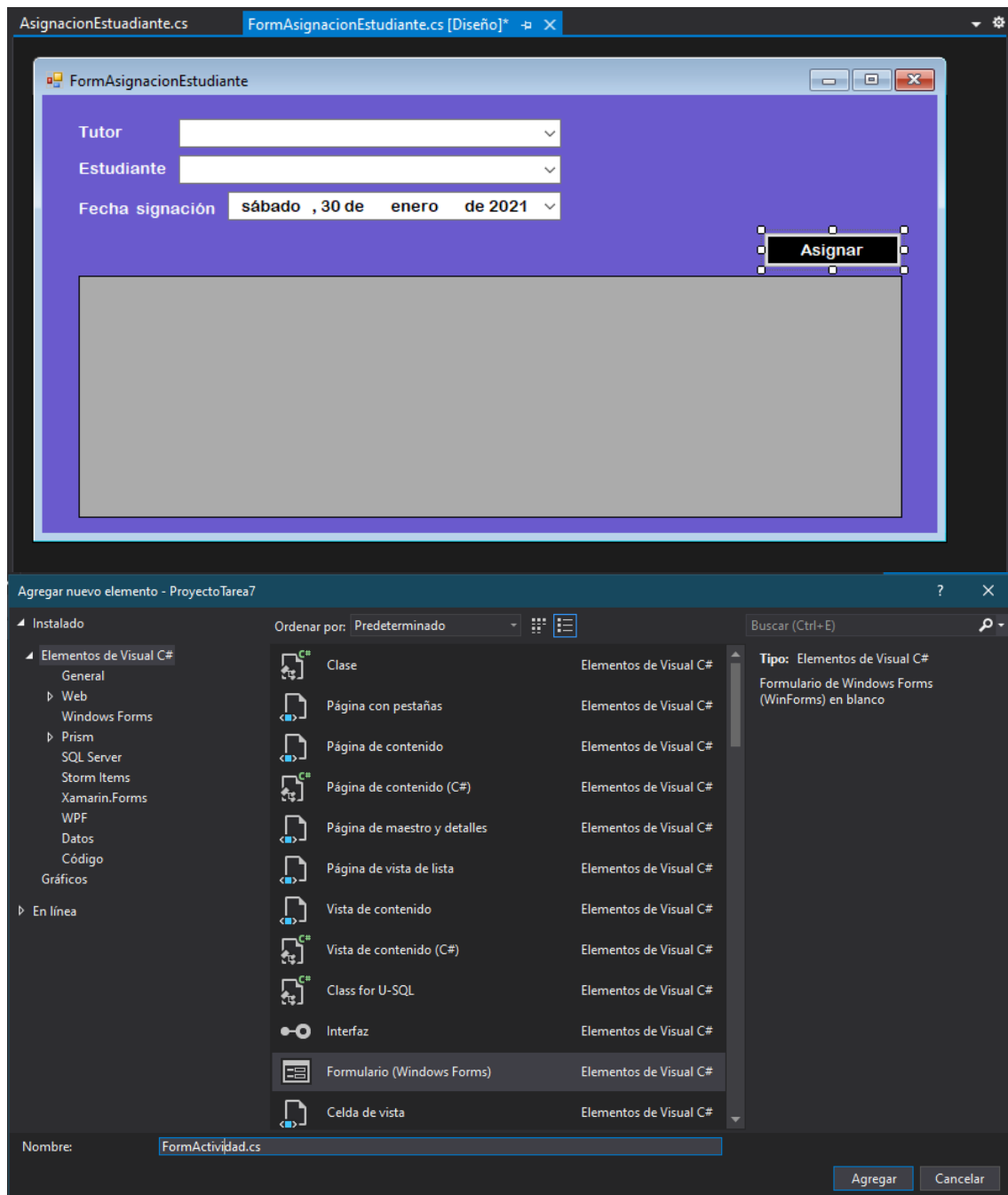
Escriba las cadenas en la colección (una por línea):

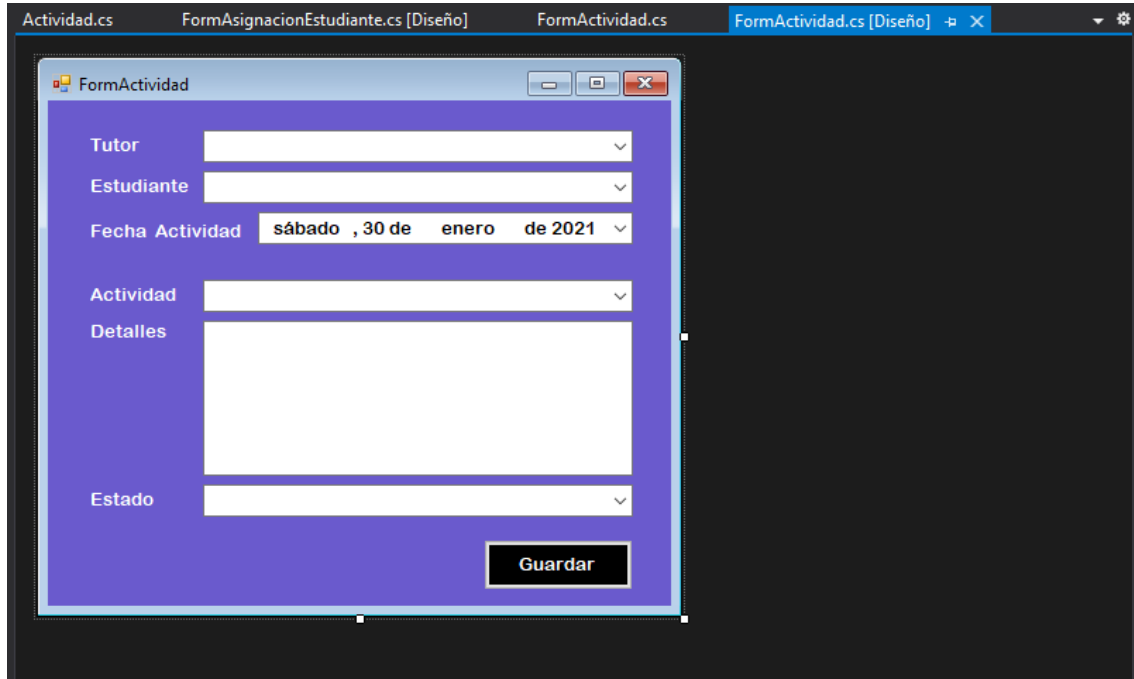
Ingeniería en Sistema
Ingeniería en Telecomunicaciones
Ingeniería en Administración de Empresas

Aceptar Cancelar









La extracción del código de pantallas

Pantalla Menu que contiene los eventos de las opciones de ingreso y la visualización de graficos

```
public partial class FormMenu : Form1
{
    int i = 0;
    Pen f = new Pen(Color.Red);
    Pen f1 = new Pen(Color.Blue);
    Pen f2 = new Pen(Color.Beige);
    Font tipo1 = new Font("Arial", 10);
    Font tipo2 = new Font("Comic Sans MS", 9);
    bool band = false;
    List<Modelos.Actividad> listaActividades = new List<Modelos.Actividad>();
    public FormMenu()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        FormEstudiante obj = new FormEstudiante();
        obj.ShowDialog();
    }

    private void button2_Click(object sender, EventArgs e)
    {
        FormTutor obj = new FormTutor();
        obj.ShowDialog();
    }
}
```

```

private void button3_Click(object sender, EventArgs e)
{
    FormAsignacionEstudiante obj = new FormAsignacionEstudiante();
    obj.ShowDialog();
    CargarEstudiantes();
}

private void button4_Click(object sender, EventArgs e)
{
    FormActividad obj = new FormActividad();
    obj.ShowDialog();
}

private void panel1_Paint(object sender, PaintEventArgs e)
{
    if (band)
    {
        graficoBarras(panel1.CreateGraphics(), listaActividades, 100, 300);
    }
}

private void panel2_Paint(object sender, PaintEventArgs e)
{
    if (band)
    {
        graficoPastel(panel2.CreateGraphics(), 50, 125);
    }
}

public void graficoBarras(Graphics t, List<Modelos.Actividad> A, int x, int y)
{
    int contS = 0, contC = 0, a = 20, ancho = 40;
    for (int i = 0; i < A.Count; i++)
    {
        if (A[i].estadoActividad.Equals("ASIGNADO")) contS++;
        if (A[i].estadoActividad.Equals("TERMINADO")) contC++;
    }
    t.DrawLine(f1, x, y, x + 300, y); t.DrawLine(f1, x, y, x, y - 250);
    t.DrawString("ESTADOS", tipo2, Brushes.Black, x + 300, y);
    t.DrawString("# ACTIVIDADES", tipo2, Brushes.Black, x - 100, y - 250);
    t.DrawRectangle(f1, x + 250, y - 190, 160, 100); t.DrawString("LEYENDA", tipo2,
Brushes.Black, x + 270, y - 170);
    t.FillRectangle(Brushes.SkyBlue, x + 270, y - 150, ancho, ancho / 2);
    t.FillRectangle(Brushes.Purple, x + 270, y - 120, ancho, ancho / 2);
    t.DrawRectangle(f1, x + 270, y - 150, ancho, ancho / 2);
    t.DrawRectangle(f1, x + 270, y - 120, ancho, ancho / 2);
    t.DrawString("ASIGNADO", tipo2, Brushes.Black, x + 320, y - 150);
    t.DrawString("TERMINADO", tipo2, Brushes.Black, x + 320, y - 120);
    t.DrawRectangle(f1, x + 270, y - 150, ancho, ancho / 2);
    t.DrawRectangle(f1, x + 270, y - 120, ancho, ancho / 2);
    t.DrawString("ASIGNADO", tipo2, Brushes.Black, x + 320, y - 150);
    t.DrawString("TERMINADO", tipo2, Brushes.Black, x + 320, y - 120);
}

```

```

t.FillRectangle(Brushes.SkyBlue, x + 50, y - (contS * a), ancho, contS * a);
t.DrawRectangle(f1, x + 50, y - (contS * a), ancho, contS * a);
t.DrawLine(f1, x, y - (contS * a), x + 50 + ancho, y - (contS * a));

t.DrawString("ASIGNADO", tipo2, Brushes.Black, x + 50, y + 30);
t.DrawString("'" + contS, tipo2, Brushes.Black, x - 20, y - (contS * a));
t.FillRectangle(Brushes.Purple, x + 140, y - (contC * a), ancho, contC * a);
t.DrawRectangle(f1, x + 140, y - (contC * a), ancho, contC * a);
t.DrawLine(f1, x + 50 + ancho, y - (contC * a), x + 140 + ancho, y - (contC * a));
t.DrawString("TERMINADO", tipo2, Brushes.Black, x + 140, y + 30);
t.DrawString("'" + contC, tipo2, Brushes.Black, x - 20, y - (contC * a));
}
public void graficoPastel(Graphics g, int x, int y)
{
    int ptot = 0; int s = 0, ns = 0;
    for (int i = 0; i < this.listaActividades.Count; i++)
    {
        if (this.listaActividades[i].estadoActividad.Equals("ASIGNADO")) s++;
        if (this.listaActividades[i].estadoActividad.Equals("TERMINADO")) ns++;
    }
    ptot = this.listaActividades.Count; if (ptot != 0)
    {
        g.DrawRectangle(f1, x + 200, y + 20, 200, 100);
        g.DrawString("LEYENDA:", tipo2, Brushes.Black, x + 210, y + 40);
        g.DrawString("ASIGNADO " + (double)(s * 100 / ptot) + " %", tipo2, Brushes.Black, x + 230,
y + 60);
        g.DrawString("TERMINADO " + (double)(ns * 100 / ptot) + " %", tipo2, Brushes.Black, x +
230, y + 80);
        g.DrawRectangle(f1, x + 210, y + 60, 10, 10);
        g.DrawRectangle(f1, x + 210, y + 80, 10, 10);
        g.FillEllipse(Brushes.DarkBlue, x - 3, y - 3, 156, 156);

        g.FillPie(Brushes.SkyBlue, x, y, 150, 150, 0, (int)(s * 360 / ptot));
        g.FillRectangle(Brushes.SkyBlue, x + 210, y + 60, 10, 10);
        g.FillPie(Brushes.Purple, x, y, 150, 150, (int)(s * 360 / ptot), (int)(ns * 360 / ptot));
        g.FillRectangle(Brushes.Purple, x + 210, y + 80, 10, 10);
    }
}

private void btnGraficar_Click(object sender, EventArgs e)
{
    listaActividades.Clear();
    listaActividades =
Program.listaActividades.Where(x=>x.identificacionEstudiante.Equals(cmbEstudiante.Selected
Value.ToString())).ToList();
    band = true;
    this.Refresh();
}
public void CargarEstudiantes()
{

```

```

        cmbEstudiante.DataSource = Program.listaAsignaciones.ToList();
        cmbEstudiante.DisplayMember = "nombresEstudiante";
        cmbEstudiante.ValueMember = "IdentificacionEstudiante";
    }
}

```

Pantalla Asignación de Tutores

```

public partial class FormAsignacionEstudiante : Form1
{
    public FormAsignacionEstudiante()
    {
        InitializeComponent();
        cmbTutor.DataSource = Program.listaTutores.ToList();
        cmbTutor.DisplayMember = "Nombres";
        cmbTutor.ValueMember = "Identificacion";

        cmbEstudiante.DataSource = Program.listaEstudiantes.ToList();
        cmbEstudiante.DisplayMember = "Nombres";
        cmbEstudiante.ValueMember = "Identificacion";
        dtpFecha.Enabled = false;
    }

    private void btnAsignar_Click(object sender, EventArgs e)
    {
        if (Program.listaAsignaciones.FirstOrDefault(x =>
x.identificacionTutor.Equals(cmbTutor.SelectedValue.ToString()) &&
x.identificacionEstudiante.Equals(cmbEstudiante.SelectedValue.ToString()))==null)
        {
            Program.listaAsignaciones.Add(
                new Modelos.AsignacionEstudiante
                {
                    identificacionTutor = cmbTutor.SelectedValue.ToString(),
                    nombresTutor = cmbTutor.Text.ToString(),
                    identificacionEstudiante = cmbEstudiante.SelectedValue.ToString(),
                    nombresEstudiante = cmbEstudiante.Text.ToString(),
                    fechaAsignacion = dtpFecha.Value,
                    idAsignacion = (Program.listaAsignaciones.Count + 1)
                }
            );
            CargarInformacion();
        }
    }

    public void CargarInformacion()
    {
        dataGridView1.DataSource = Program.listaAsignaciones.ToList();
        dataGridView1.Refresh();
    }

    private void FormAsignacionEstudiante_Load(object sender, EventArgs e)
    {

```

```

        CargarInformacion();
    }
}

```

Pantalla Asignación de Actividades

```

public FormActividad()
{
    InitializeComponent();

    cmbTutor.DataSource = Program.listaTutores.ToList();
    cmbTutor.DisplayMember = "Nombres";
    cmbTutor.ValueMember = "Identificacion";
}

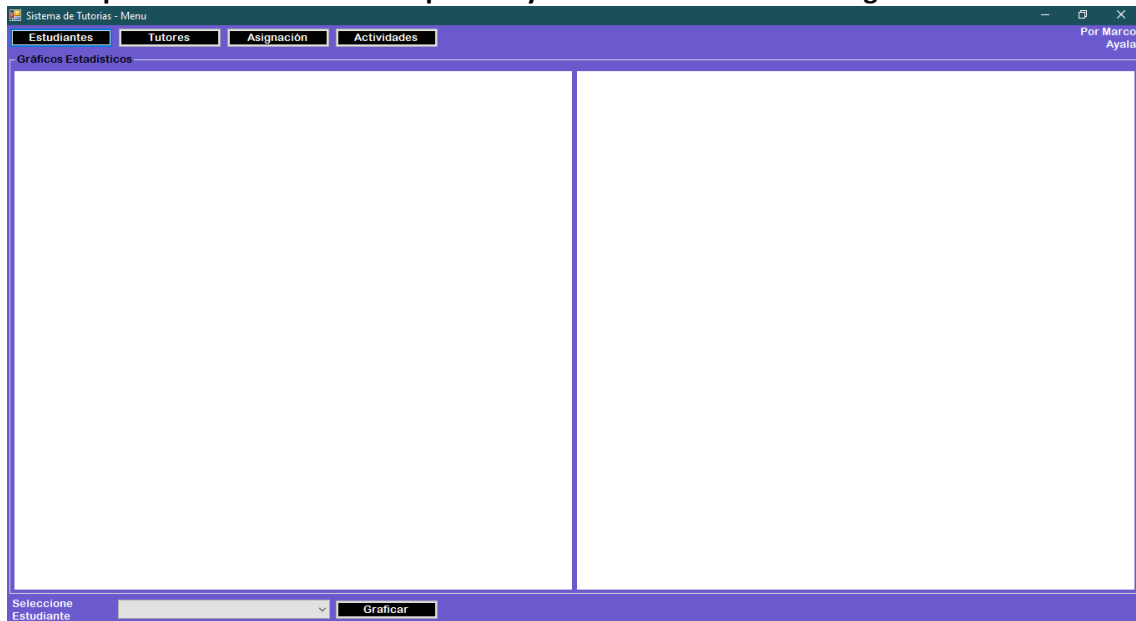
private void cmbTutor_SelectedIndexChanged(object sender, EventArgs e)
{
    if (Program.listaAsignaciones.Where(x =>
x.identificacionTutor.Equals(cmbTutor.SelectedValue.ToString())).ToList().Count > 0)
    {
        cmbEstudiante.DataSource = Program.listaAsignaciones.Where(x =>
x.identificacionTutor.Equals(cmbTutor.SelectedValue.ToString())).ToList();
        cmbEstudiante.DisplayMember = "nombresEstudiante";
        cmbEstudiante.ValueMember = "identificacionEstudiante";
    }
}

private void btnGuardar_Click(object sender, EventArgs e)
{
    Program.listaActividades.Add(new Modelos.Actividad
    {
        identificacionTutor = cmbTutor.SelectedValue.ToString(),
        identificacionEstudiante = cmbEstudiante.SelectedValue.ToString(),
        nombreActividad = cmbActividad.Text,
        detalleActividad = txtDetalles.Text,
        estadoActividad = cmbEstado.Text.ToString(),
        fechaActividad = dtpFechaActividad.Value
    });
    MessageBox.Show("Tarea Asignada");
    cmbActividad.SelectedIndex = -1;
    cmbTutor.SelectedIndex = -1;
    cmbEstado.SelectedIndex = -1;
    txtDetalles.Text = string.Empty;
}
}

```

Ejecución del Proyecto

En esta pantalla será el menú de opciones y también se visualizará los gráficos estadísticos



En esta pantalla se crea los estudiantes con carrera y nivel

	carrera	nivel	identificacion	nombres	apellidos
▶	Ingeniería e...	Septimo	1721481586	Marco Anto...	Ayala Lituma
	Ingeniería e...	Octavo	1706067426	Carlos Jose	Moreno Car...

En esta pantalla se crear los tutores y la modalidad que imparte la tutoría

FormTutor

Identificación: Facultad:

Nombres: Disponibilidad:

Apellidos:

Guardar

	facultad	disponibilidad	identificacion	nombres	apellidos
▶	Ingenierias	Virtual	0782232123	Jorge	Tatayo
	Licenciaturas	Presencial	1723290890	Manuel	Lopez
	Licenciaturas	Virtual	0500121876	Maria Belen	Santos

En esta pantalla se asigna los estudiantes al tutor

FormAsignacionEstudiante

Tutor:

Estudiante:

Fecha signación:

Asignar

	idAsignacion	identificacion	nombresTutor	identificacion	nombresEstuc	fechaAsignaci
▶	1	0500121876	Maria Belen	1706067426	Carlos Jose	31/1/2021 8:...
	2	0500121876	Maria Belen	0987654321	Leonardo	31/1/2021 8:...
	3	0500121876	Maria Belen	1721481586	Marco Anto...	31/1/2021 8:...

En esta pantalla se asigna las actividades a los estudiantes según la asignación del tutor

FormActividad

Tutor: **Maria Belen**

Estudiante: **Leonardo**

Fecha Actividad: **domingo , 31 de enero de 2021**

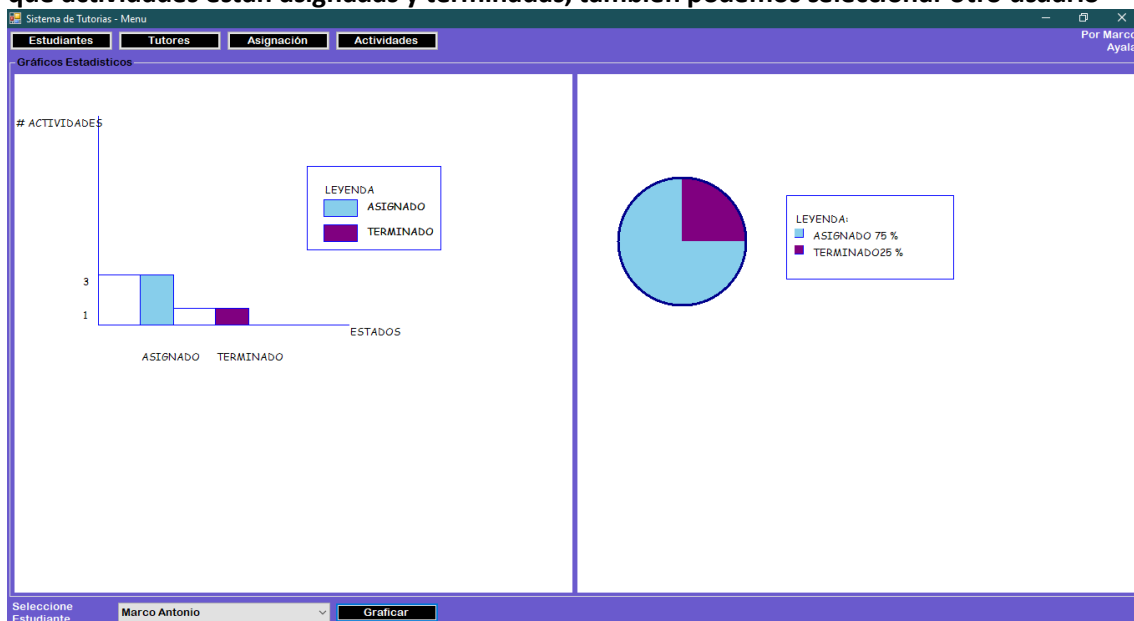
Actividad: **Proyecto**

Detalles: **Creacion del ambiente de desarrollo**

Estado: **ASIGNADO**

Guardar

En la parte inferior tenemos el botón graficas y debemos seleccionar el estudiante para ver que actividades están asignadas y terminadas, también podemos seleccionar otro usuario



BIBLIOGRAFÍA:

Titulo	Autor	Año	Editorial	URL/Observacion
Java 2 Lenguaje y Aplicaciones	Ceballos Sierra F.J.	2015	RA-MA-Editorial.	https://elibro.net/es/lc/uisrael/titulos/624
Empezar a programar usando Java (3ra. Ed)	Prieto Saez, N y Casanova Faus A.	2016	Edittorial de la Universidad Politecnica de Valencia	https://elibro.net/es/lc/uisrael/titulos/574

