

**Universidad Israel**

**DEPARTAMENTO:** Ciencias de la Ingeniería **PROFESOR:** Mg. Luis Fernando Aguas B.

**CARRERA:** Sistemas de Información **ESTUDIANTE:** Marco Antonio Ayala Lituma

**CURSO:** Séptimo **PARALELO:** “A“ **DESCRIPCIÓN:** Tarea 1-S7

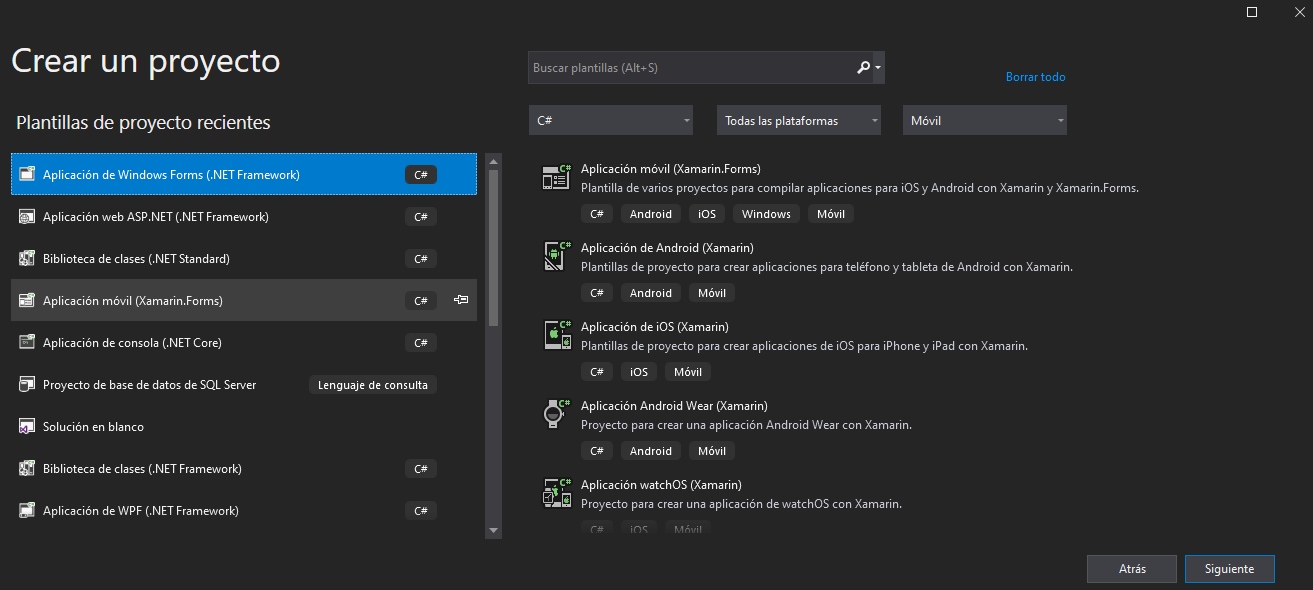
**ASIGNATURA:** Plataformas de Desarrollo 1

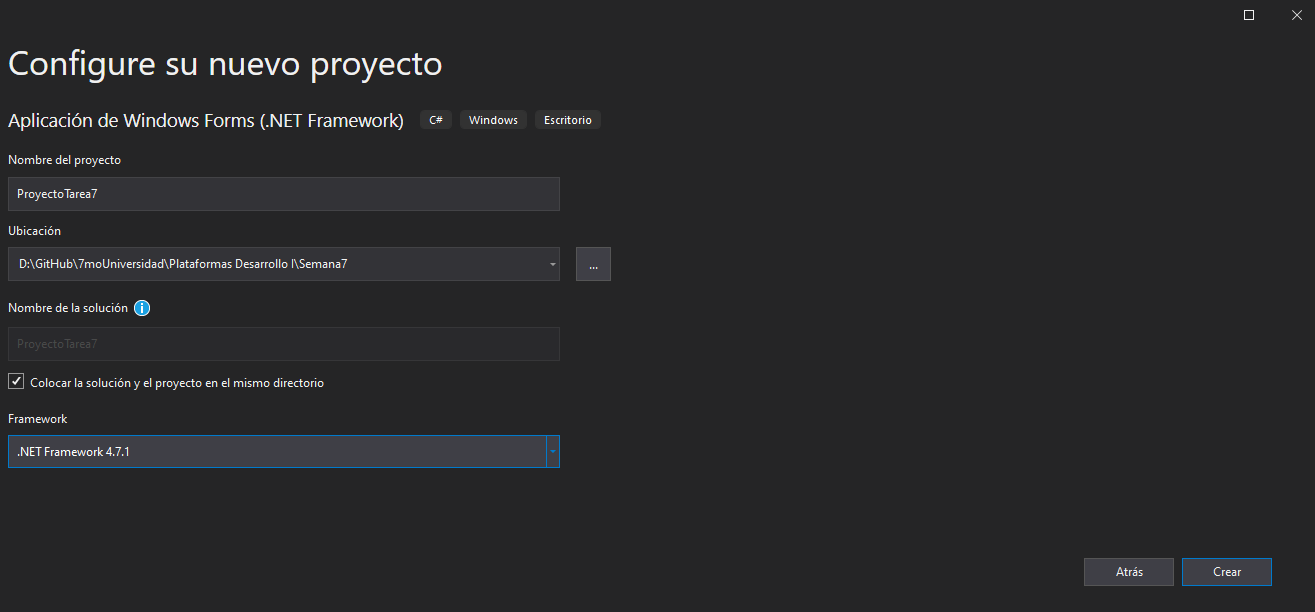
**TEMA: Sistemas de Tutorias**

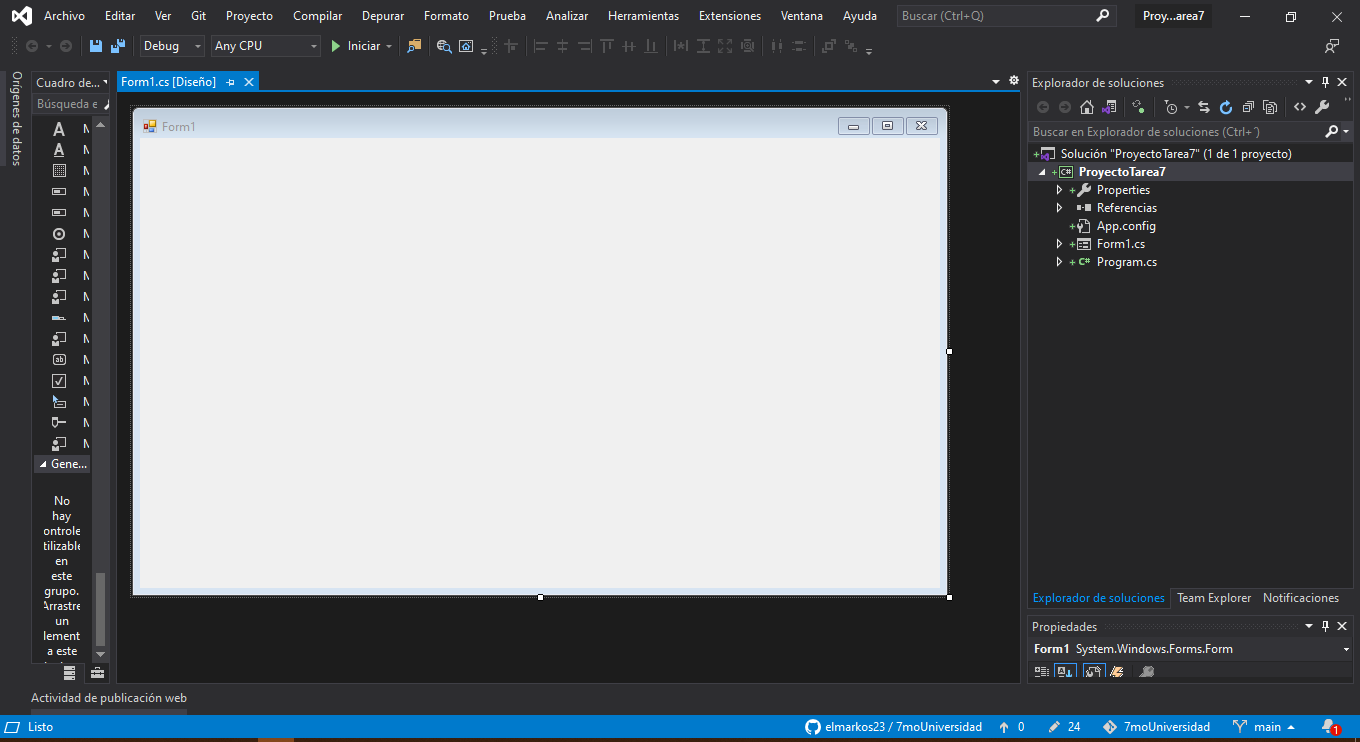
**Realice un programa en el cual implemente un sistema de tutorías, el cual permita ingresar un tutor, asignarle un estudiante, registrar la reunión de tutorías, registrar las actividades extracurriculares, registrar los acuerdos que se llegan. El programa debe estar hecho con:**

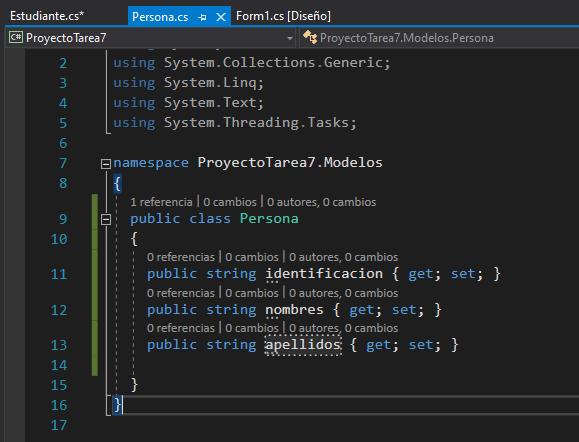
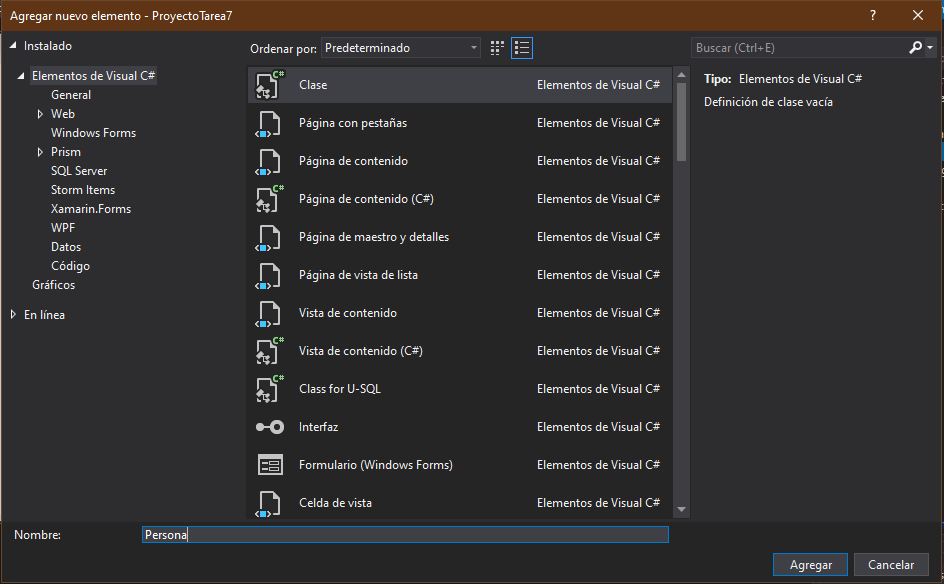
* **Estándares de codificación.**
* **Mínimo 2 Gráficos estadísticos**
* **Clases**
* **Estándares de interfases**

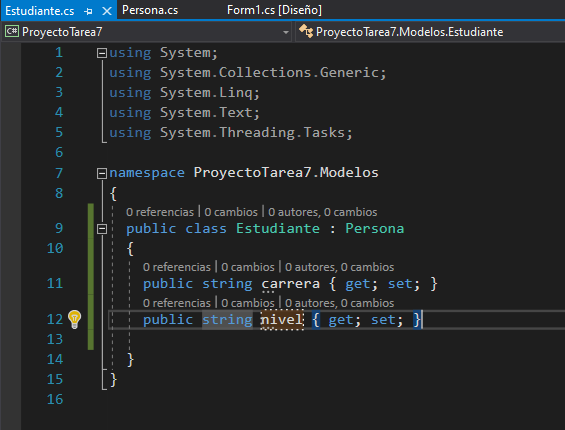
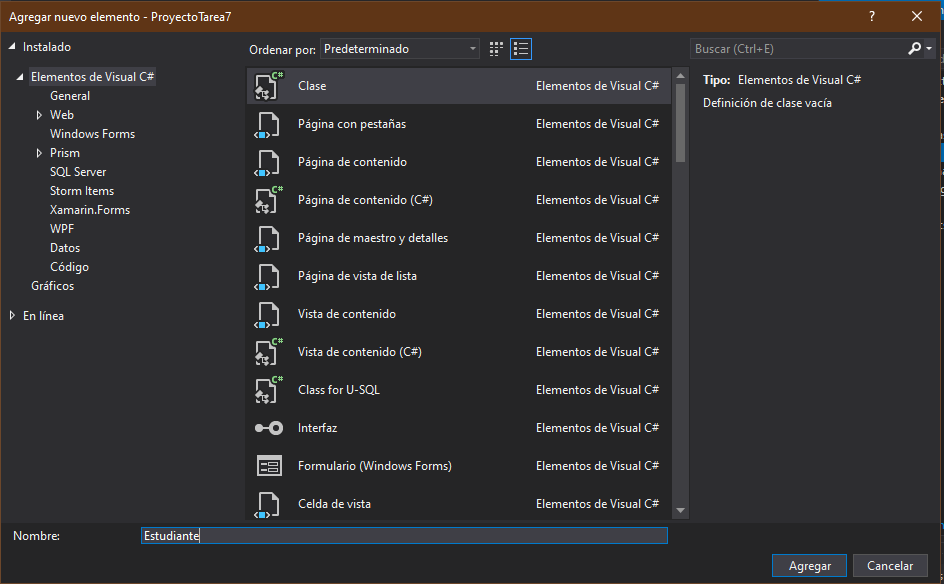
**DESARROLLO:**

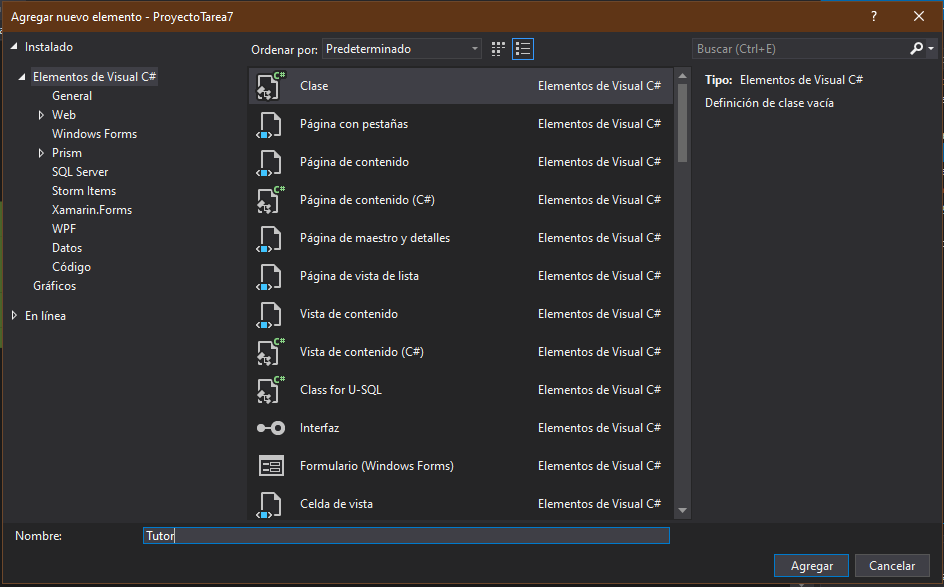


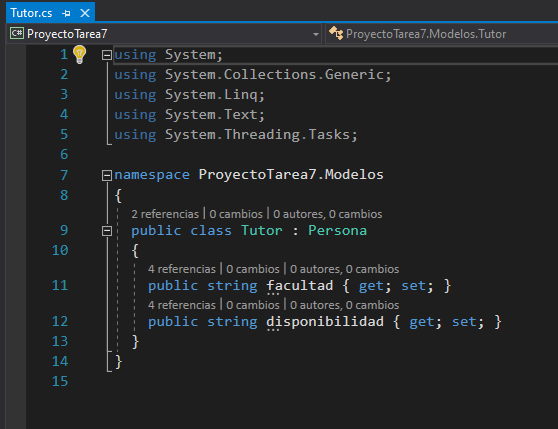


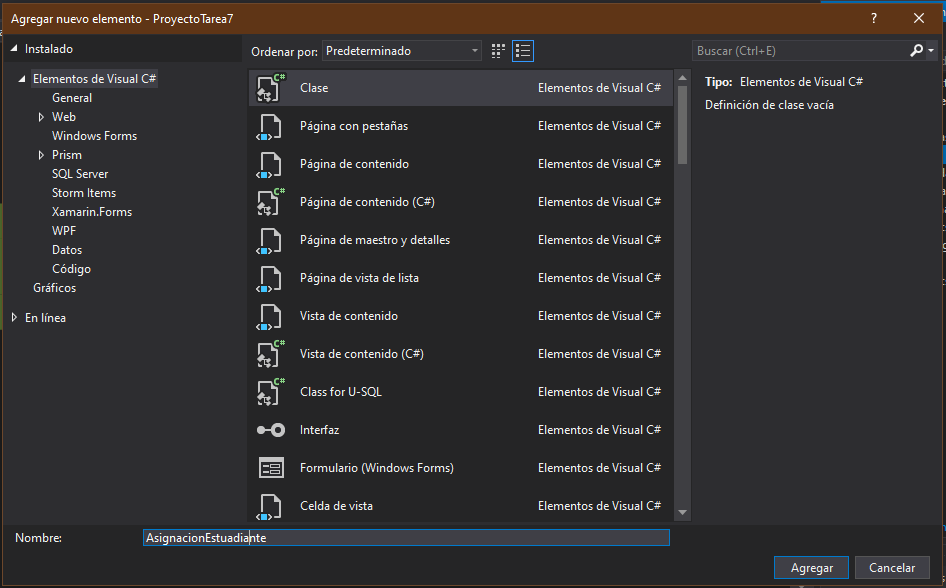


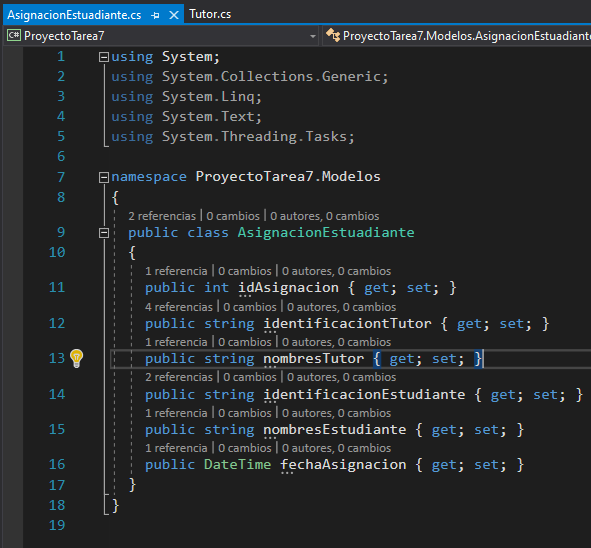


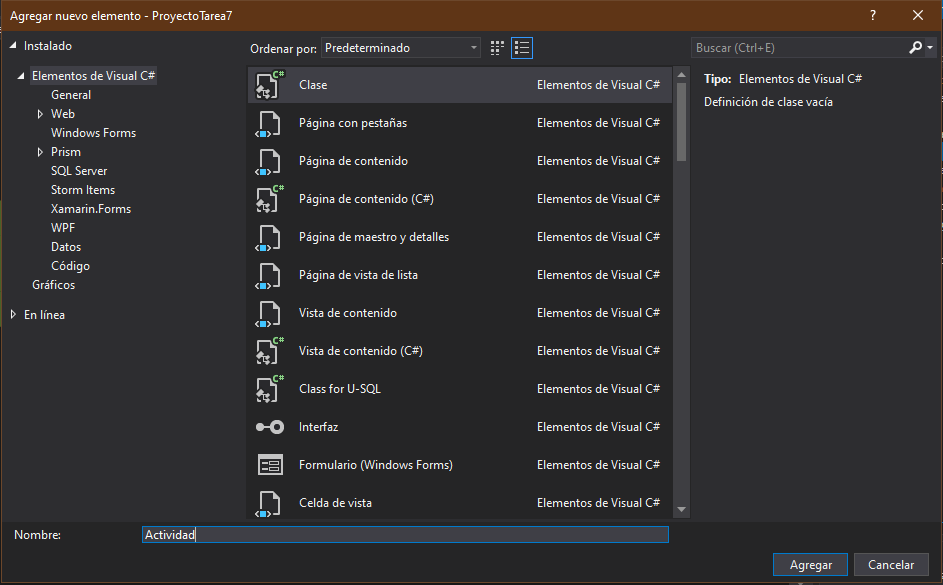


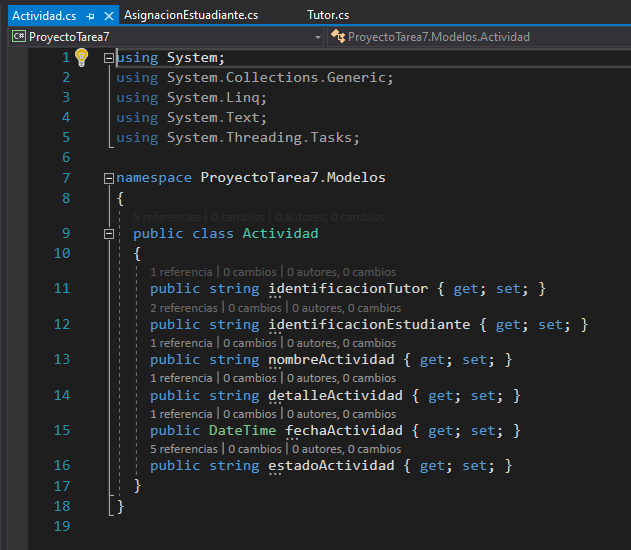


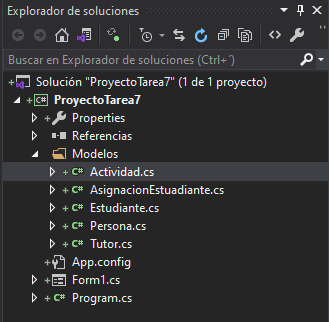


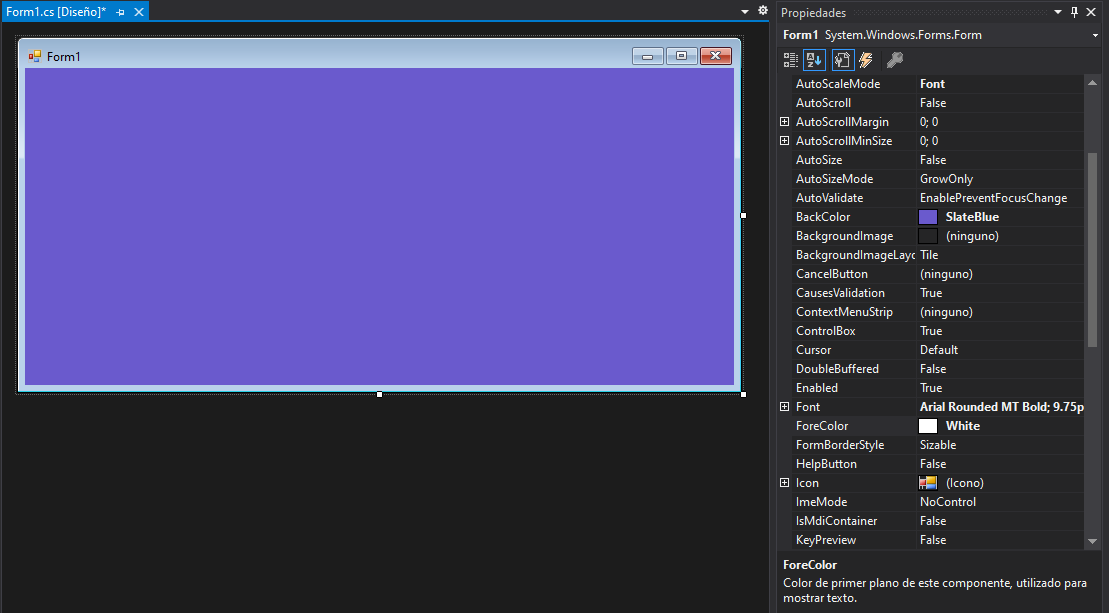


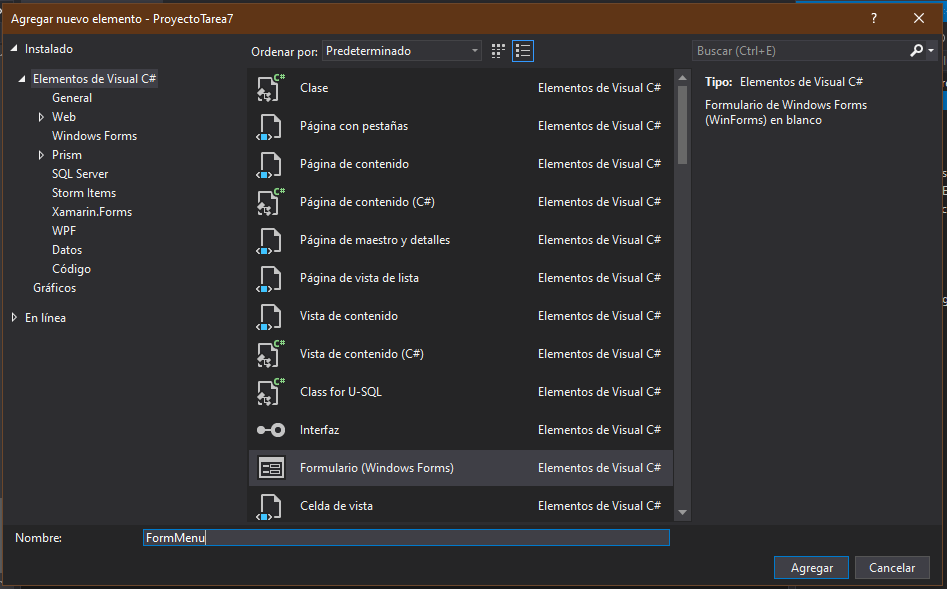


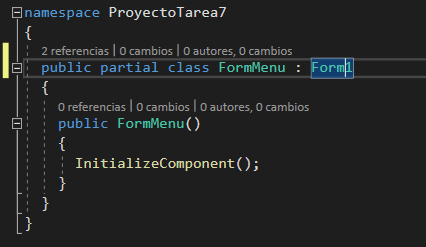


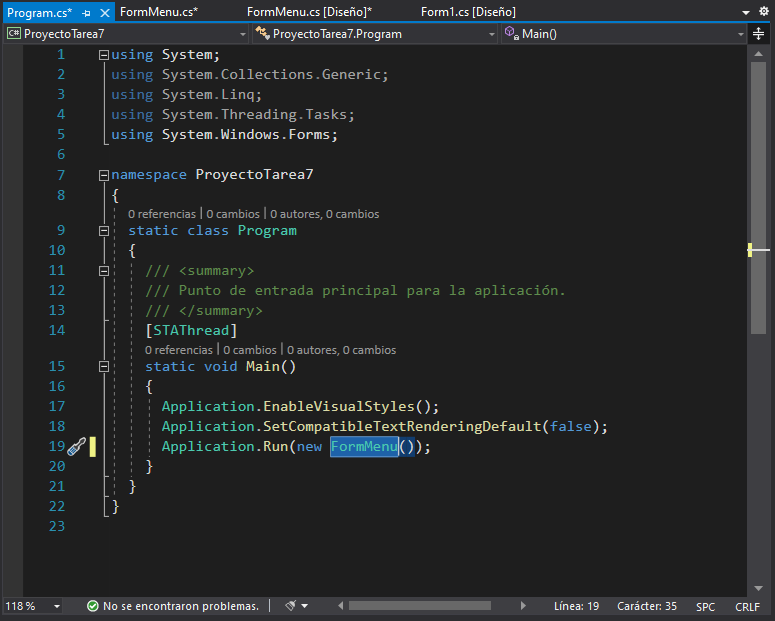




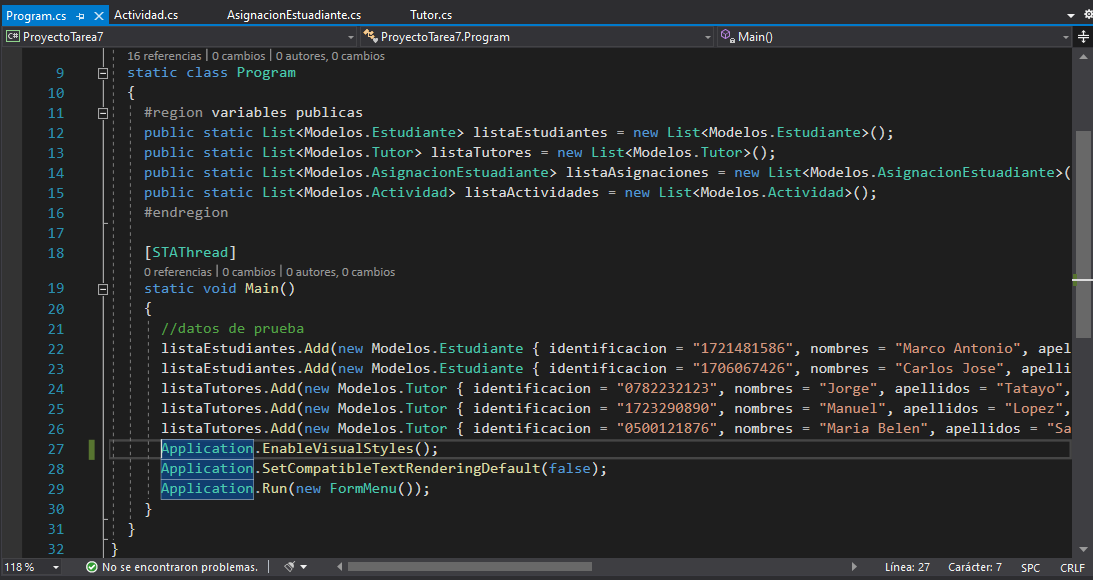


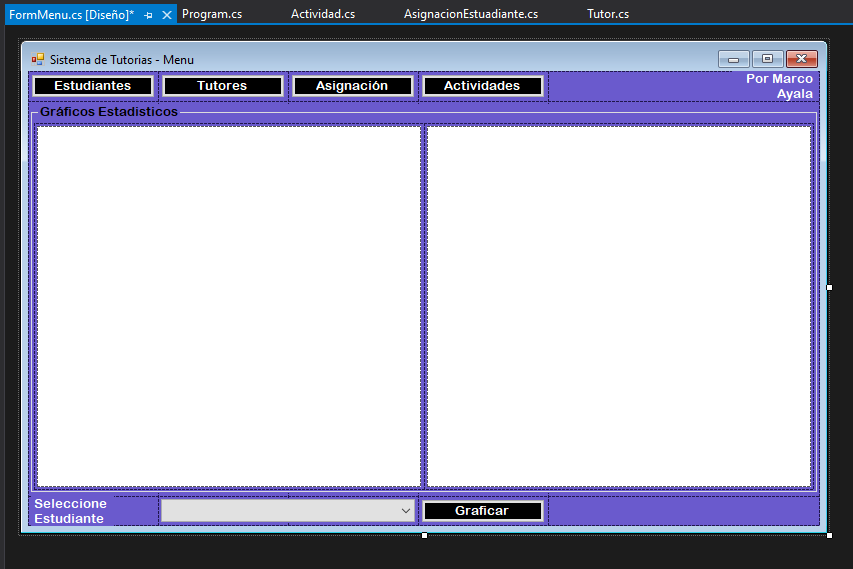


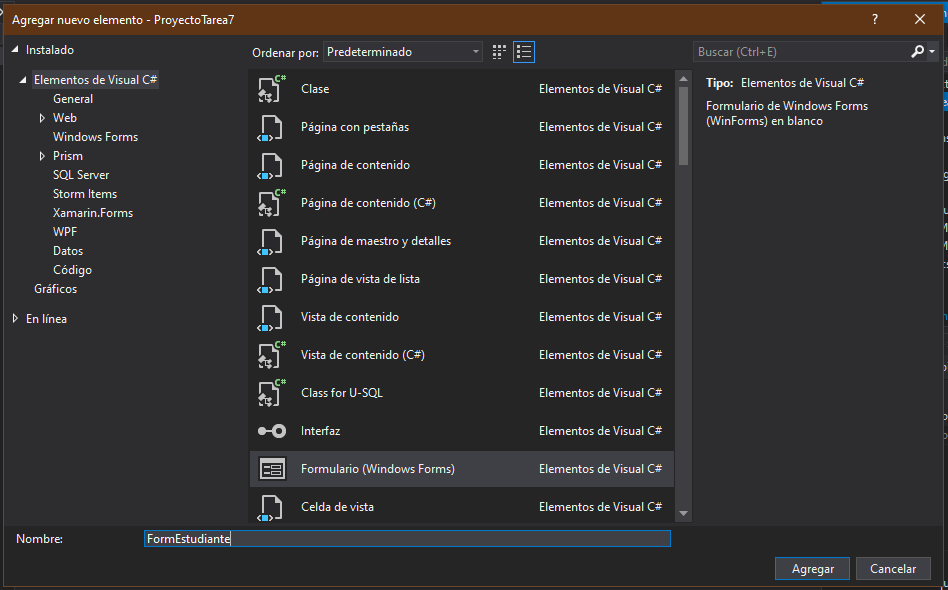


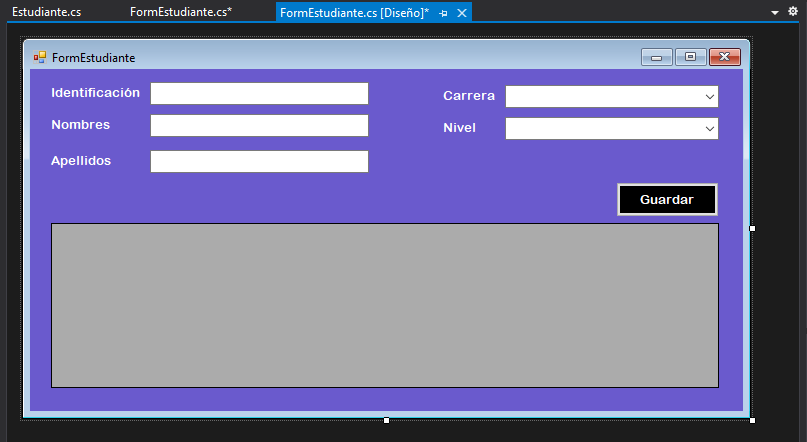


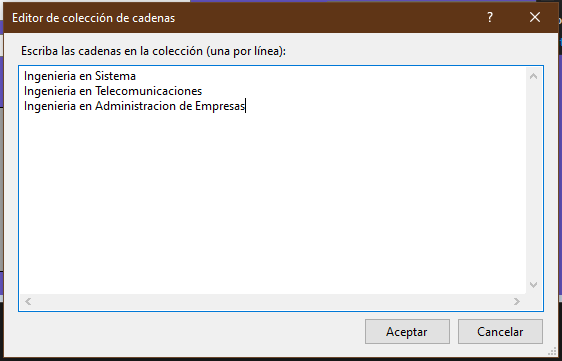
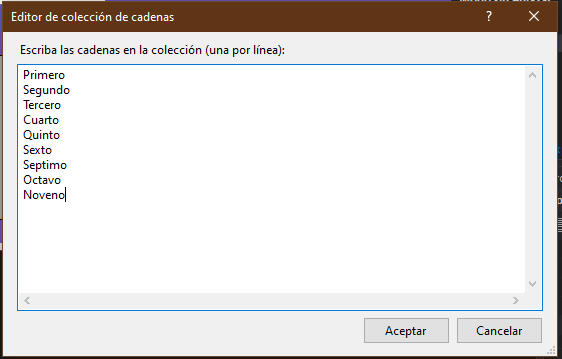
**Aquí también agregamos las variables globales y datos de prueba para tutores y estudiantes**

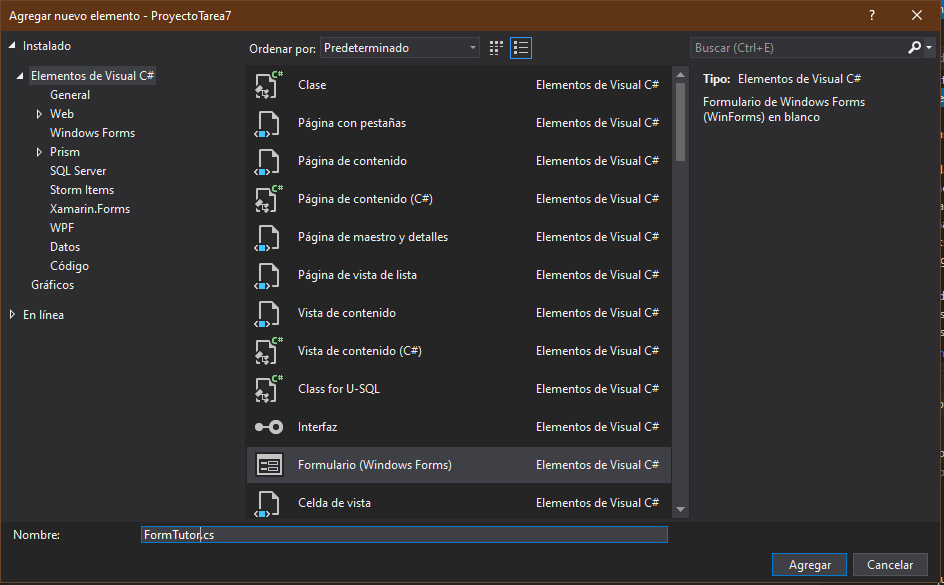


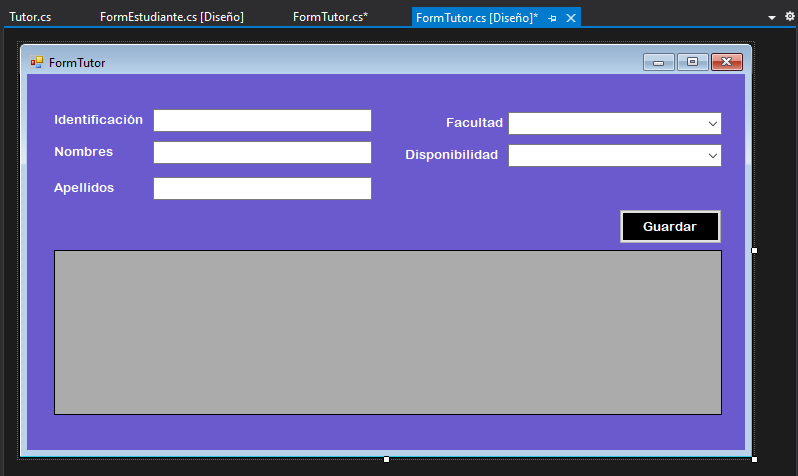


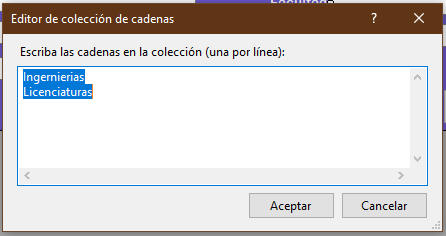


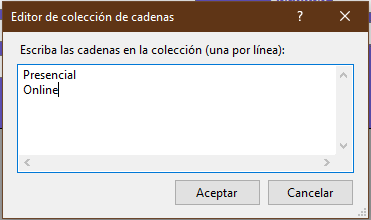


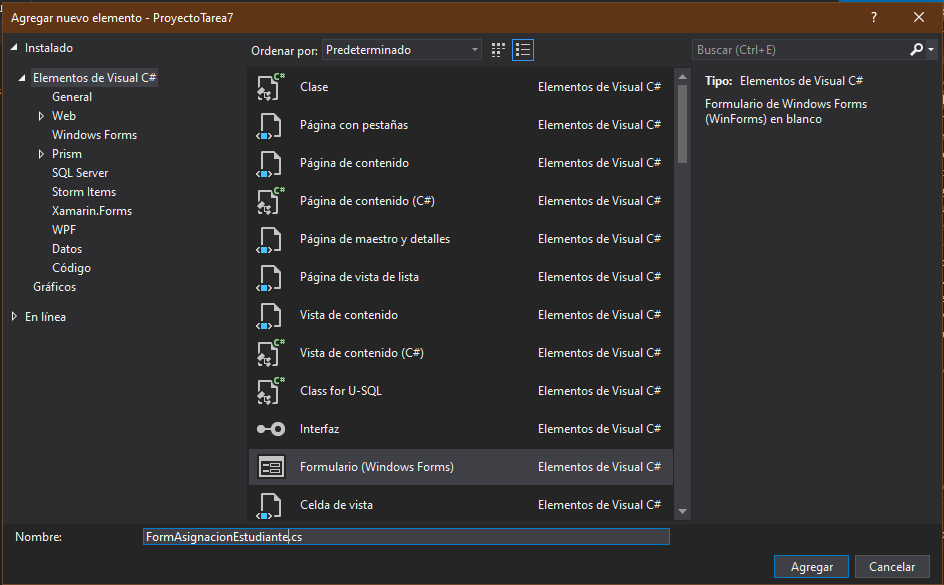


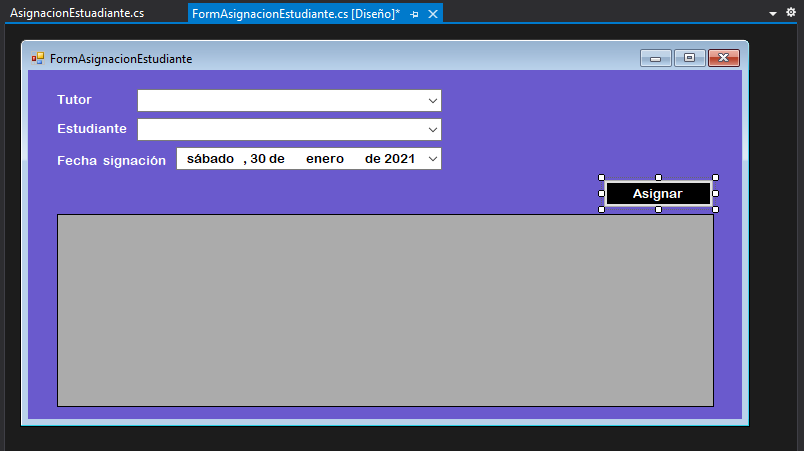


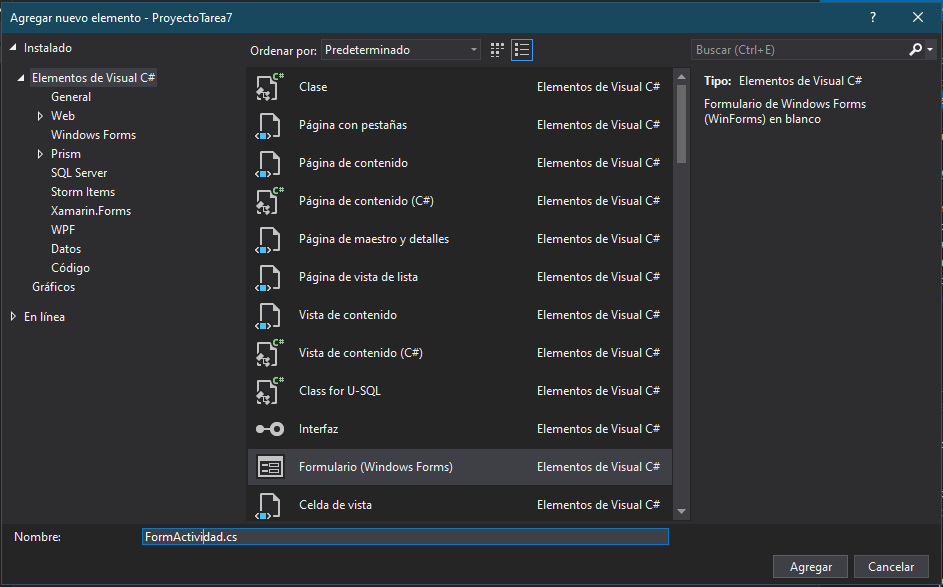


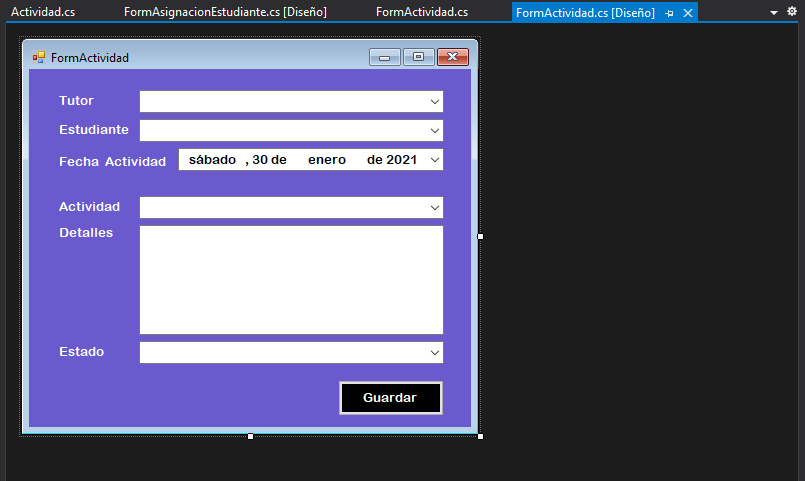












**La extracción del código de pantallas**

**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.SelectedValue.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.identificaciontTutor.Equals(cmbTutor.SelectedValue.ToString()) && x.identificacionEstudiante.Equals(cmbEstudiante.SelectedValue.ToString()))==null)

{

Program.listaAsignaciones.Add(

new Modelos.AsignacionEstuadiante

{

identificaciontTutor = 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.identificaciontTutor.Equals(cmbTutor.SelectedValue.ToString())).ToList().Count > 0)

{

cmbEstudiante.DataSource = Program.listaAsignaciones.Where(x => x.identificaciontTutor.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;

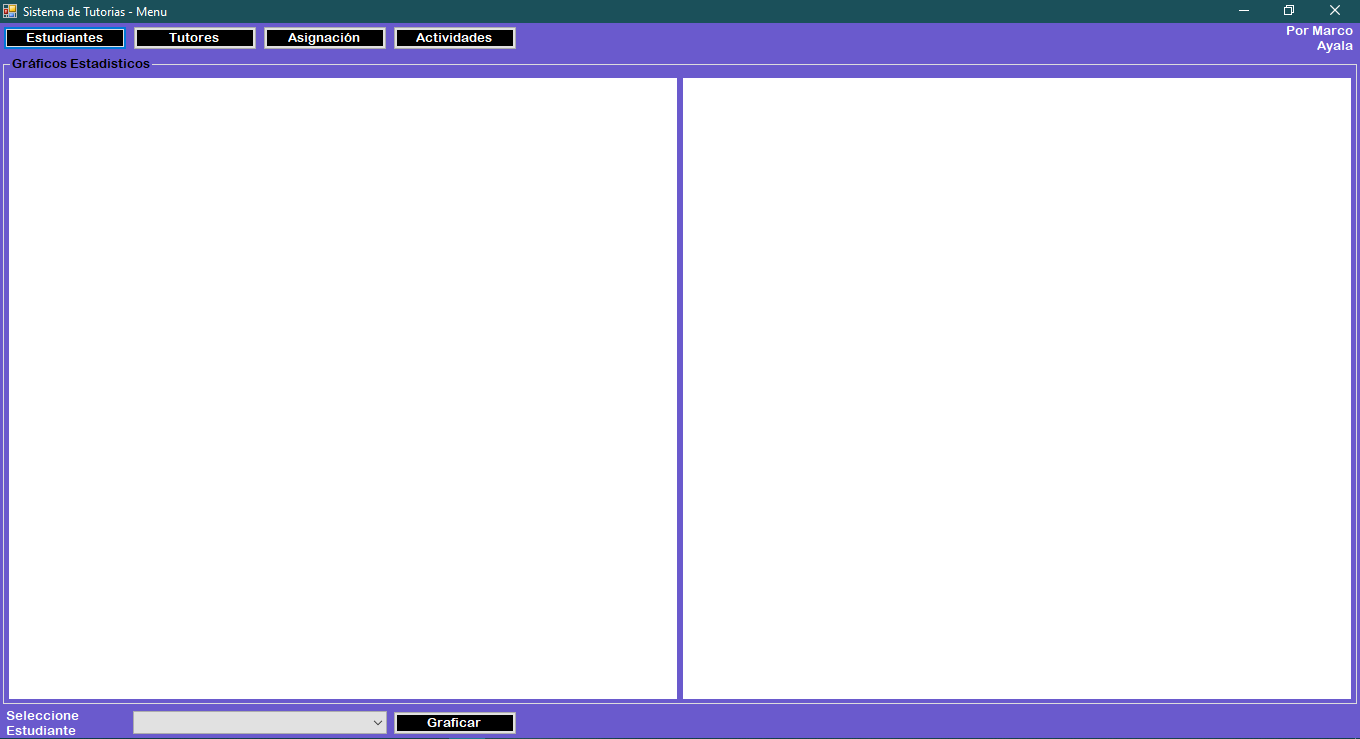
txtDetalles.Text = string.Empty;

}

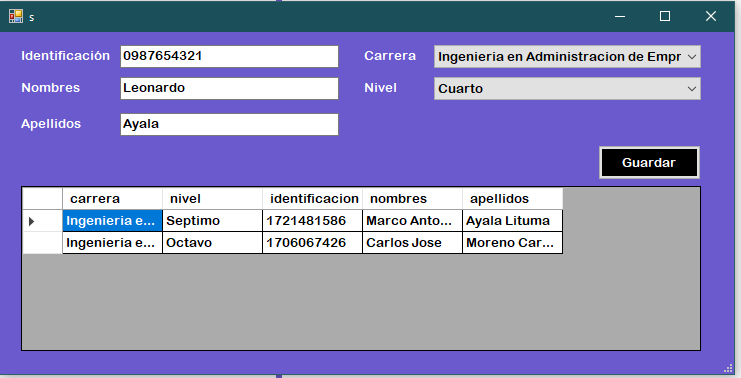
}

**Ejecución del Proyecto**

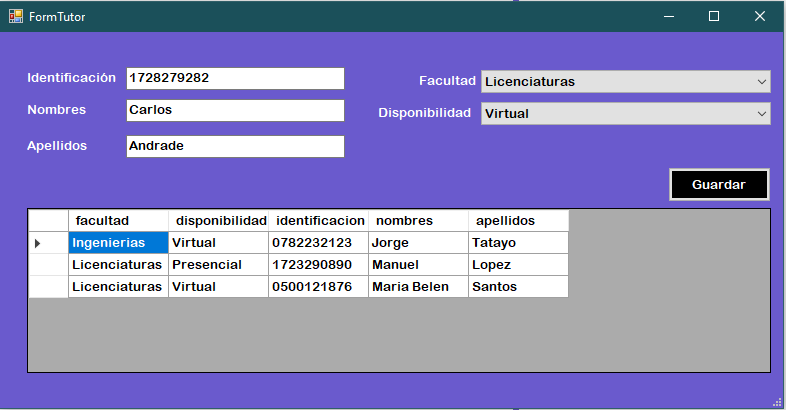
**En esta pantalla será el menú de opciones y también se visualizará los gráficos estadisticos**



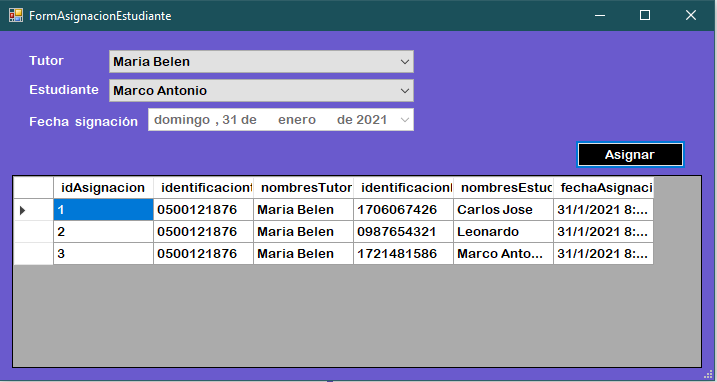
**En esta pantalla se crea los estudiantes con carrera y nivel**



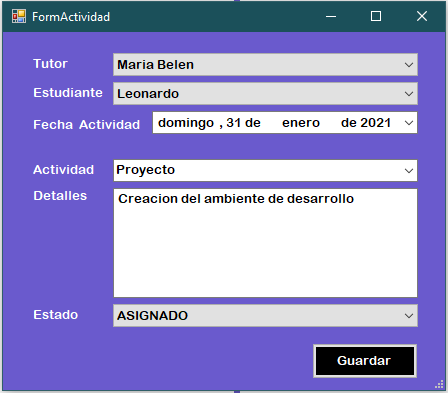
**En esta pantalla se crear los tutores y la modalidad que imparte la tutoría**



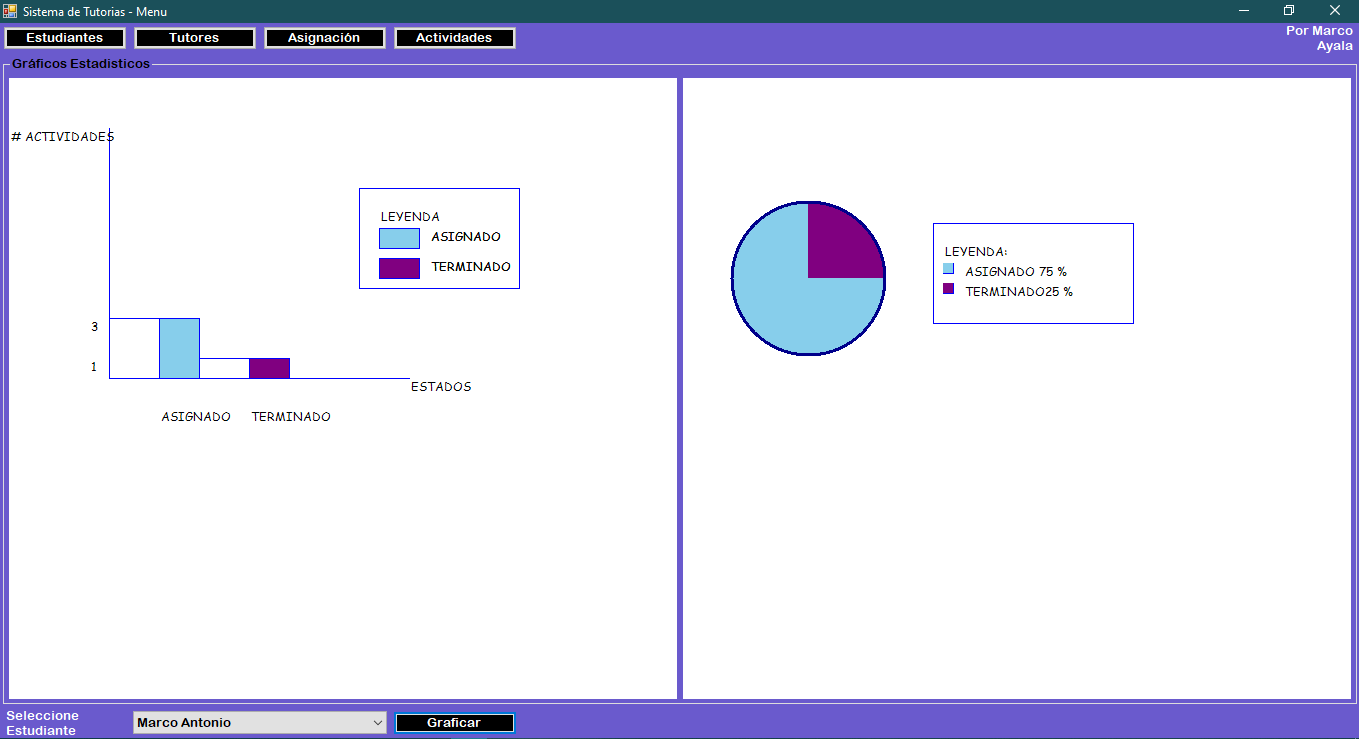
**En esta pantalla se asigna los estudiantes al tutor**



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



**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/62458> |
| Empezar a programar usando Java (3ra. Ed) | Prieto Saez, N y Casanova Faus A. | 2016 | Edittorial de la Univerdad Politecnica de Valencia | <https://elibro.net/es/lc/uisrael/titulos/57434> |