UNIVERSIDAD ISRAEL

CIENCIAS DE LA INGENIERÍA

CARRERA DE SISTEMAS DE INFORMACIÓN

PLATAFORMAS DE DESARROLLO 1

SEMESTRE 2020 B

INFORME DE LABORATORIO S4 – S5

TEMA: Aplicaciones C# (Diagramas de barras y de pastel)

ESTUDIANTE(S): Marco Ayala
CURSO: Séptimo
PARALELO: "A"

PROFESOR: Mg. Luis Fernando Aguas Bucheli

QUITO, 2020

1. **TEMA:** Aplicaciones C# (Diagramas de barras y de pastel)

2. **OBJETIVOS:**
   - Adquirir los conceptos básicos relacionados con C#
   - Reconocer las características de C#

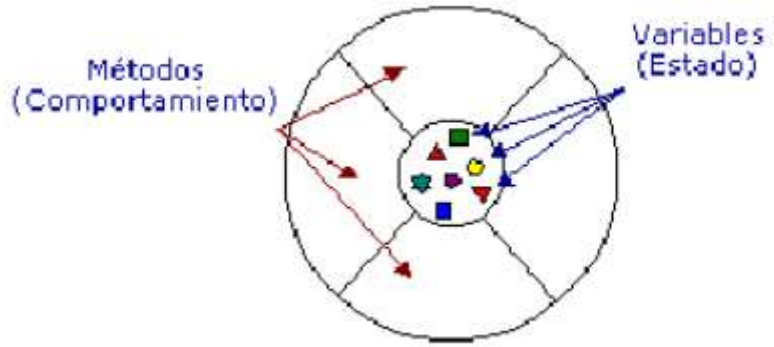
3. **INTRODUCCION:**

Clases y Objetos

Objeto

Un objeto es una encapsulación genérica de datos y de los procedimientos para manipularlos. Al igual que los objetos del mundo real, los objetos de software tienen un estado y un comportamiento. El estado de los objetos se determina a partir de una o más variables y el comportamiento con la implementación de métodos.

La siguiente figura muestra la representación común de los objetos de software.

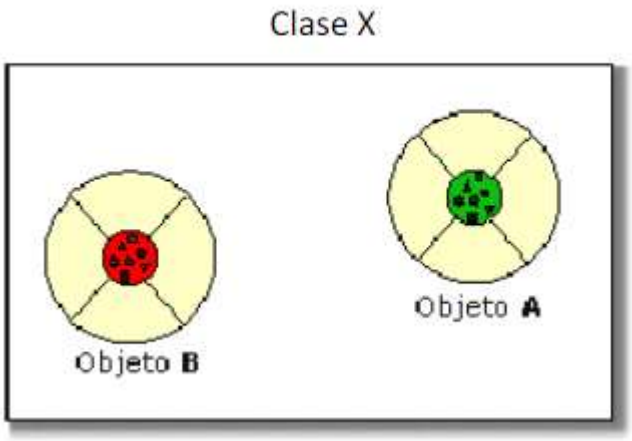


Como se observa en la figura, todos los objetos tienen una parte pública (su comportamiento) y una parte privada (su estado). En este caso, hicimos una vista transversal, pero desde el mundo exterior, el objeto se observará como una esfera.

Clase

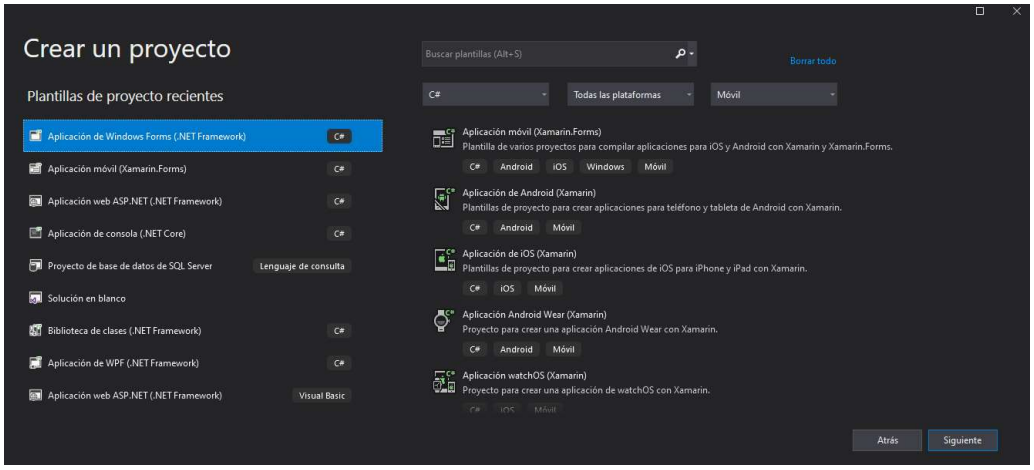
Una clase está formada por los métodos y las variables que definen las características comunes a todos los objetos de esa clase. Precisamente la clave de la OOP está en abstraer los métodos y los datos comunes a un conjunto de objetos y almacenarlos en una clase.

Una clase equivale a la generalización de un tipo específico de objetos. Una instancia es la concreción de una clase.

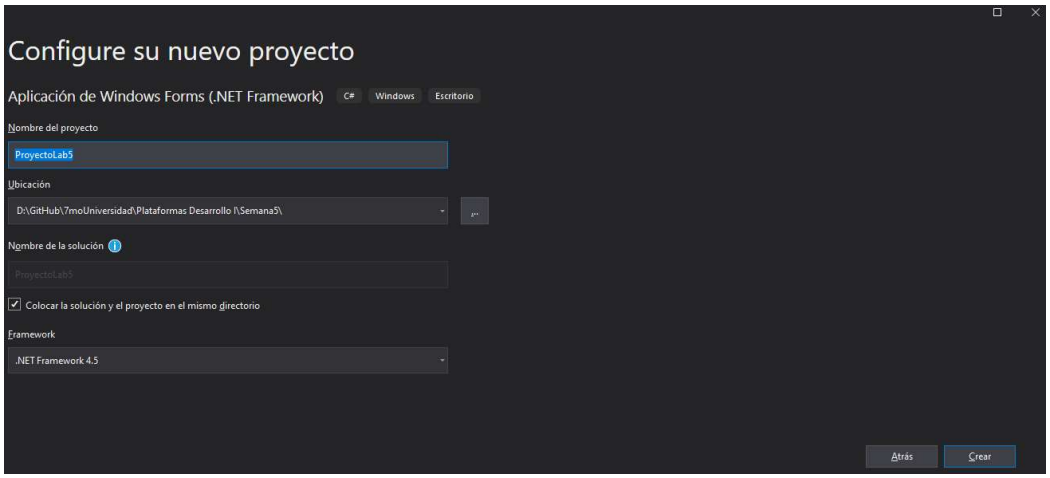


4. DESARROLLO:

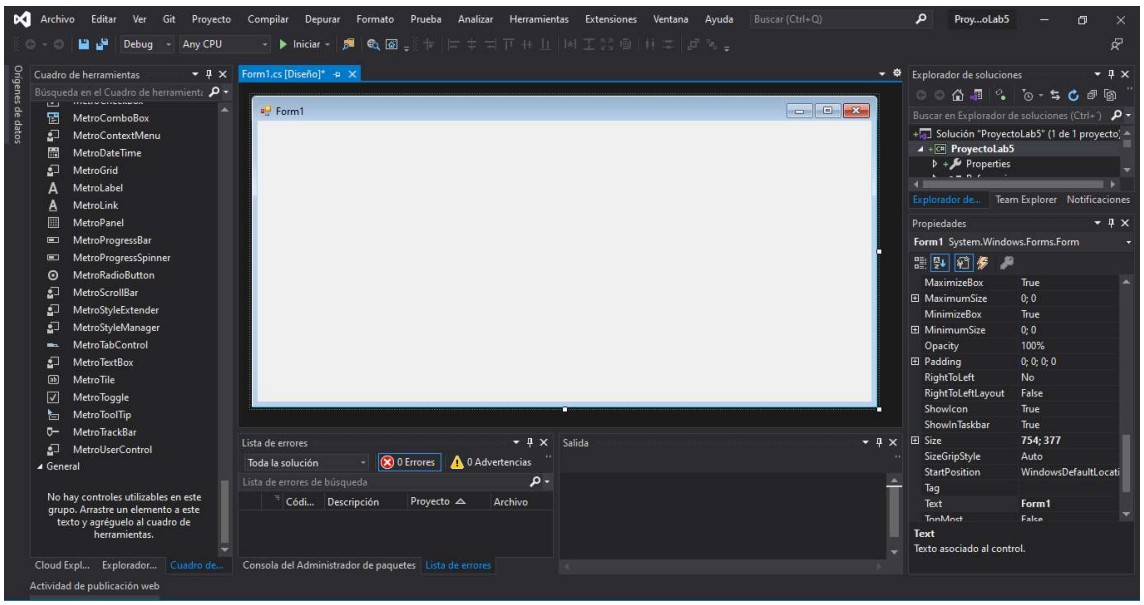
Seleccionamos proyecto de Windows Forms



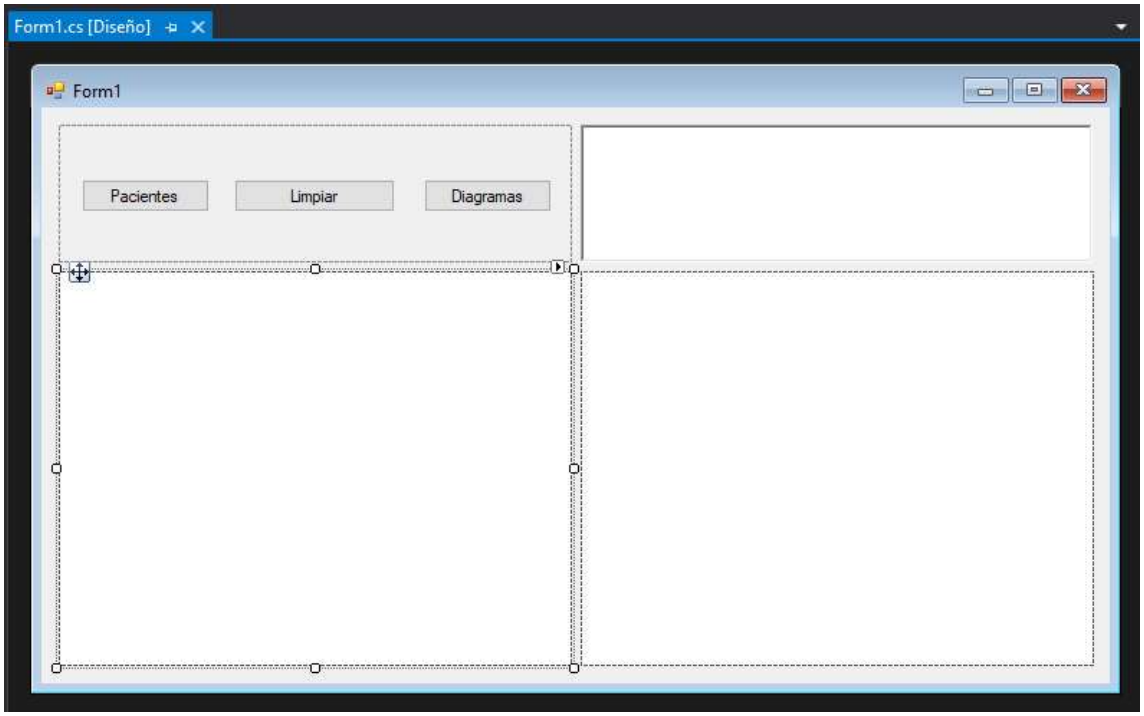
Le colocamos el nombre del proyecto y le ubicamos en carpeta



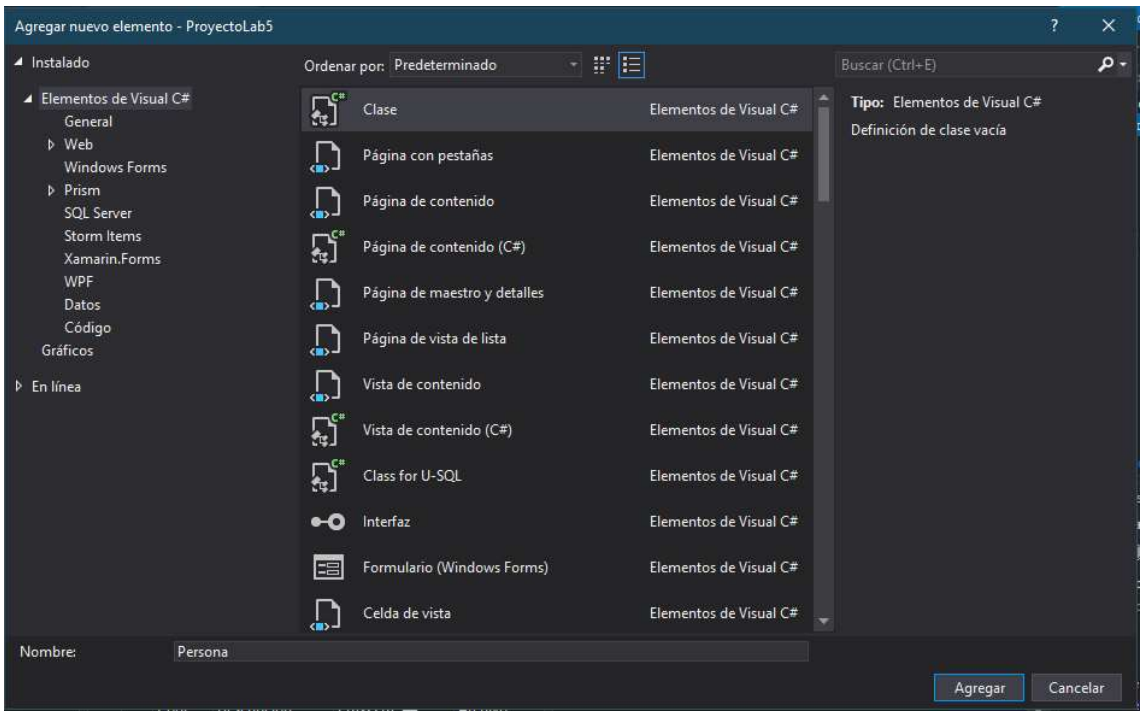
Así queda el proyecto inicial



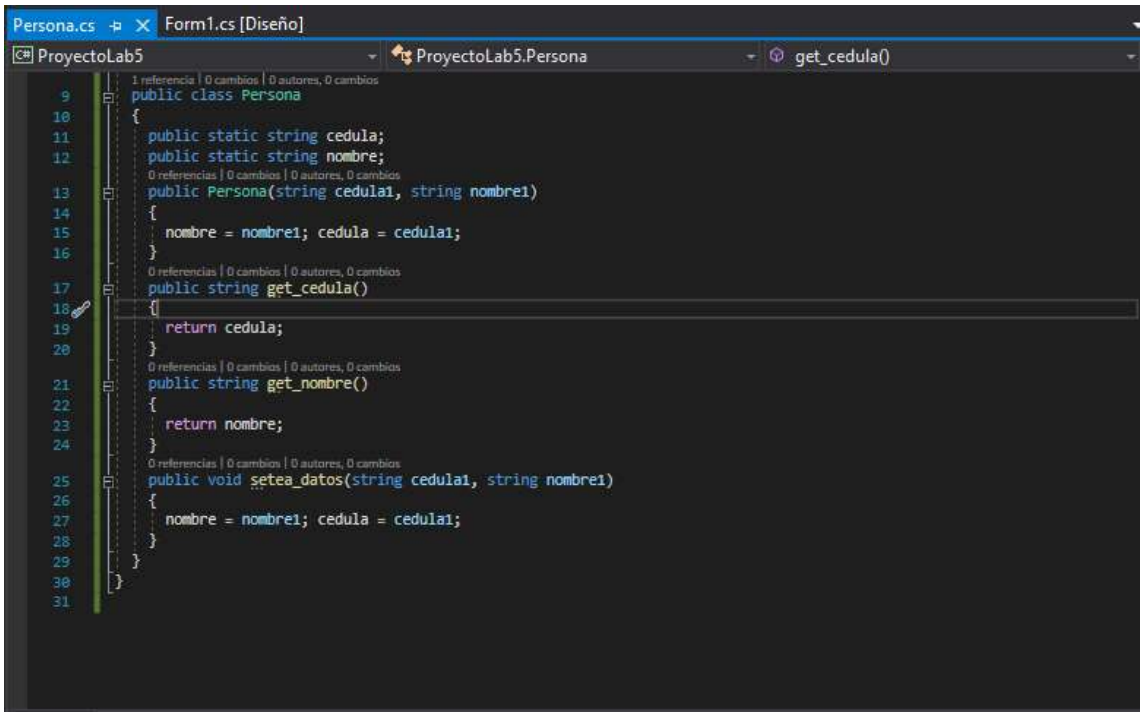
Le Diseñamos la pantalla



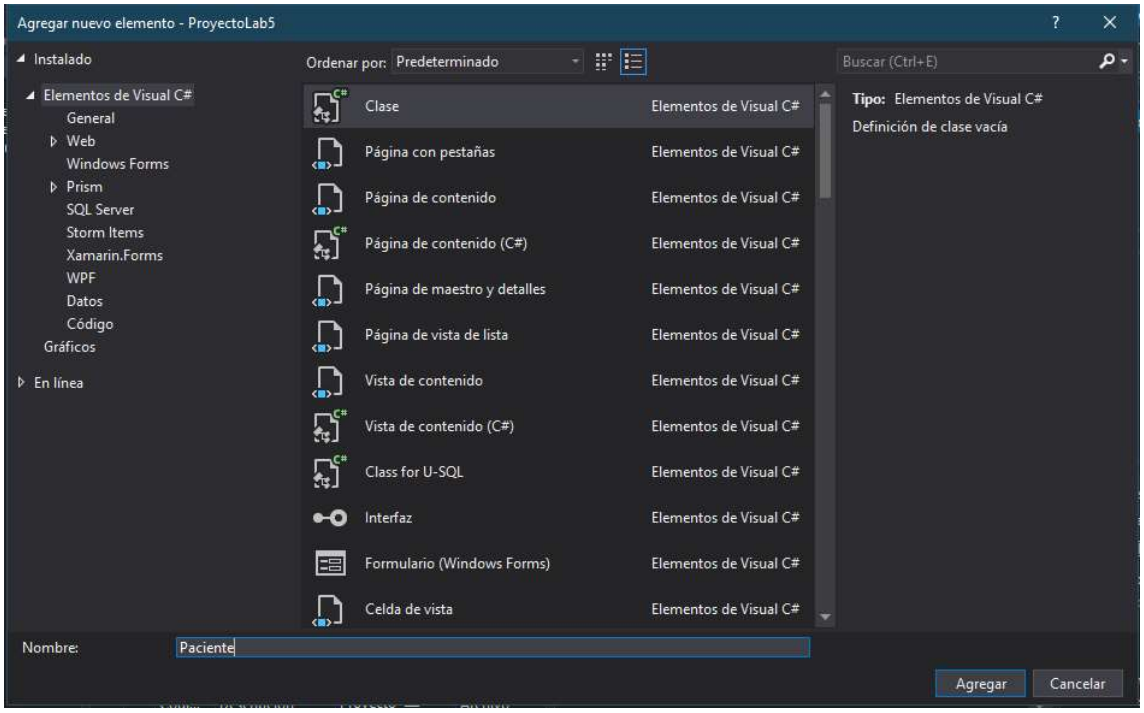
Creamos la clase Persona



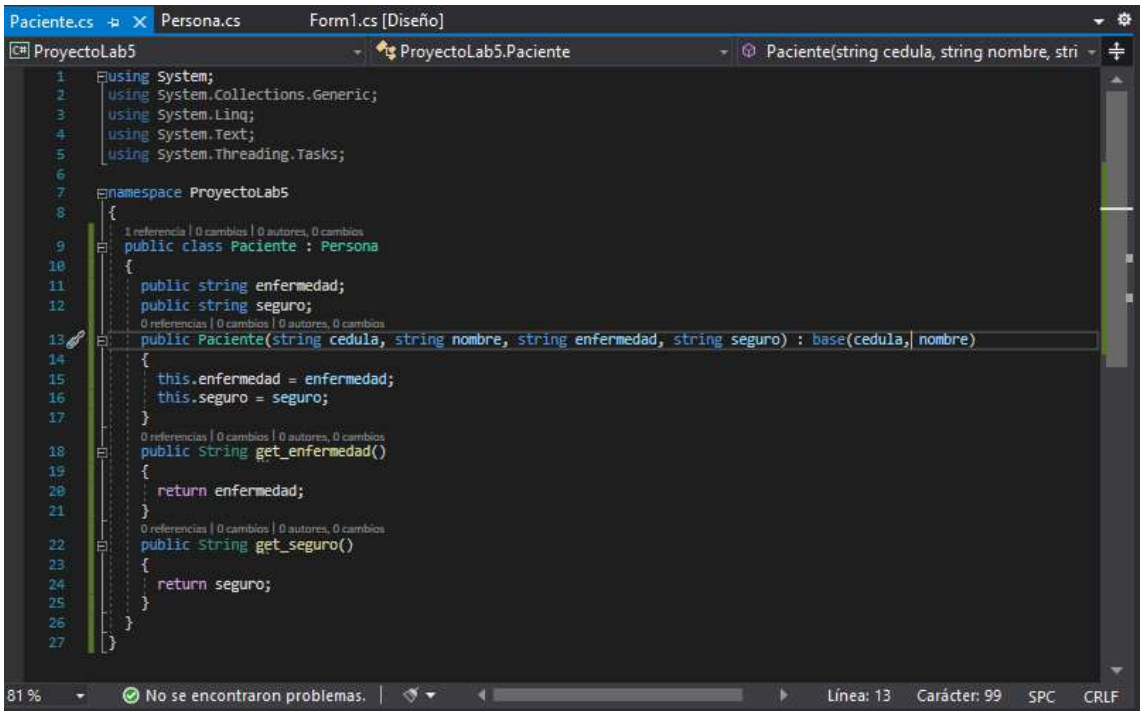
Agregamos el siguiente código de la clase Persona



Creamos Clase Paciente



Agregamos el siguiente código en la clase.



Agregamos el siguiente de en el botón paciente

```
1 referencia | 0 cambios | 0 autores, 0 cambios
private void btnPacientes_Click(object sender, EventArgs e)
{
  A[0] = new Paciente("1721481586", "Marco Ayala", "Ninguna", "Seguro");
  A[1] = new Paciente("1719343020", "Belen Narvaez", "Ninguna", "Seguro");
  richTextBox1.AppendText("\nCédula - Apellidos y Nombres - Enfermedad- Seguro");
  richTextBox1.AppendText("\n----------------------");
  for (i = 0; i < A.Length; i++)
  {
    richTextBox1.AppendText("\n" + A[0].get_cedula() + "-" +
                                   A[1].get_nombre() + "-" +
                                   A[i].get_enfermedad() + "-" +
                                   A[i].get_seguro());
    richTextBox1.AppendText("\n");
  }
}
```

Agregamos las siguientes métodos.

```
1 referencia | 0 cambios | 0 autores, 0 cambios
public void graficoPastel(Graphics g, int x, int y)
{
  int ptot = 0; int s = 0, ns = 0;
  for (int i = 0; i < A.Length; i++)
  {
    if (A[i].get_seguro().Equals("Seguro")) s++;
    if (A[i].get_seguro().Equals("No seguro")) ns++;
  }
  ptot = A.Length; if (ptot != 0)
  {
    g.DrawRectangle(p1, x + 200, y + 20, 100, 100);
    g.DrawString("Leyenda:", tipo2, Brushes.Black, x + 210, y + 40);
    g.DrawString("S " + (double)(s * 100 / ptot) + " %", tipo2, Brushes.Black, x + 230, y + 60);
    g.DrawString("NS " + (double)(ns * 100 / ptot) + " %", tipo2, Brushes.Black, x + 230, y + 80);
    g.DrawRectangle(p1, x + 210, y + 60, 10, 10);
    g.DrawRectangle(p1, x + 210, y + 80, 10, 10);
    g.FillEllipse(Brushes.Blue, x - 3, y - 3, 156, 156);
    g.FillPie(Brushes.LightBlue, x, y, 150, 150, 0, (int)(s * 360 / ptot));
    g.FillRectangle(Brushes.LightBlue, x + 210, y + 60, 10, 10);
    g.FillPie(Brushes.Yellow, x, y, 150, 150, (int)(s * 360 / ptot), (int)(ns * 360 / ptot));
    g.FillRectangle(Brushes.Yellow, x + 210, y + 80, 10, 10);
  }
}
```

```
1 referencia | 0 cambios | 0 autores, 0 cambios
public void graficoBarras(Graphics t, Paciente[] A, int x, int y)
{
  int contS = 0, contC = 0, a = 20, ancho = 40;
  for (int i = 0; i < A.Length; i++)
  {
    if (A[i].get_seguro().Equals("Seguro")) contS++;
    if (A[i].get_seguro().Equals("No Seguro")) contC++;
  }
  t.DrawLine(p1, x, y, x + 300, y); t.DrawLine(p1, x, y, x, y - 250);
  t.DrawString("Tipo", tipo2, Brushes.Black, x + 300, y);
  t.DrawString("# Asegurados", tipo2, Brushes.Black, x - 100, y - 250);
  t.DrawRectangle(p1, x + 250, y - 190, 160, 100); t.DrawString("Leyenda", tipo2, Brushes.Black, x + 270, y - 170);
  t.FillRectangle(Brushes.LightBlue, x + 270, y - 150, ancho, ancho / 2);
  t.FillRectangle(Brushes.Blue, x + 270, y - 120, ancho, ancho / 2);
  t.DrawRectangle(p1, x + 270, y - 150, ancho, ancho / 2);
  t.DrawRectangle(p1, x + 270, y - 120, ancho, ancho / 2);
  t.DrawString("Seguro", tipo2, Brushes.Black, x + 320, y - 150);
  t.DrawString("No Seguro", tipo2, Brushes.Black, x + 320, y - 120);
  t.DrawRectangle(p1, x + 270, y - 150, ancho, ancho / 2);
  t.DrawRectangle(p1, x + 270, y - 120, ancho, ancho / 2);
  t.DrawString("Seguro", tipo2, Brushes.Black, x + 320, y - 150);
  t.DrawString("No Seguro", tipo2, Brushes.Black, x + 320, y - 120);
  t.FillRectangle(Brushes.Blue, x + 140, y - (contC * a), ancho, contC * a);
  t.DrawRectangle(p1, x + 140, y - (contC * a), ancho, contC * a);
  t.DrawLine(p1, x + 50 + ancho, y - (contC * a), x + 140 + ancho, y - (contC * a));
  t.DrawString("No Seguro", tipo2, Brushes.Black, x + 140, y + 30);
  t.DrawString("" + contC, tipo2, Brushes.Black, x - 20, y - (contC * a));
}
```

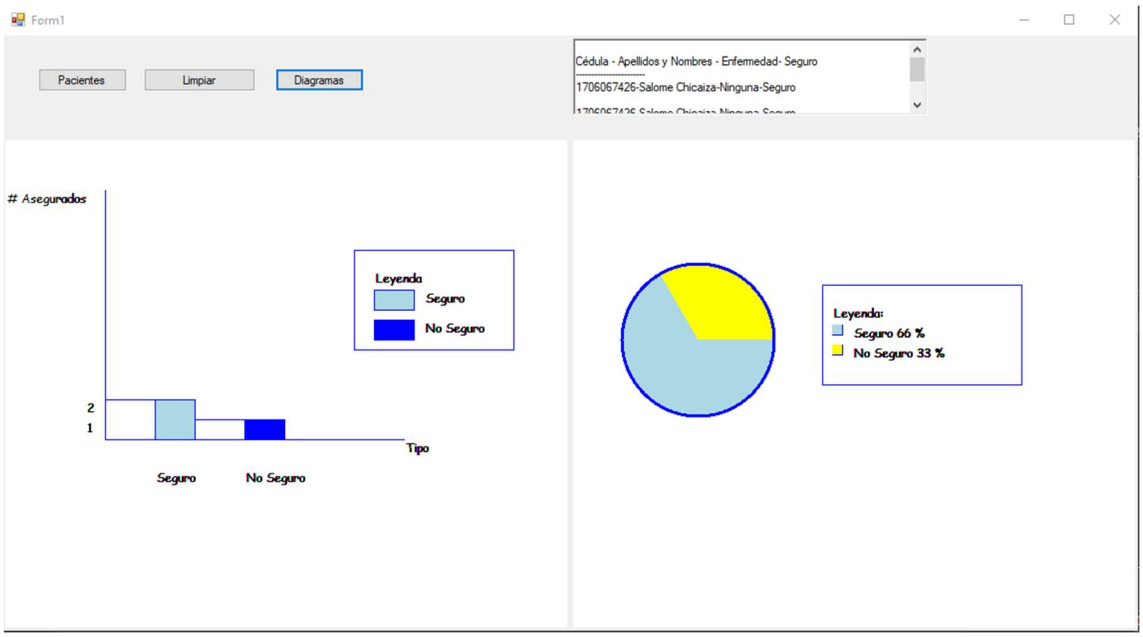
Ahora agregamos el siguiente código en el panel 2 y panel 3

```
1 referencia | 0 cambios | 0 autores, 0 cambios
private void panel2_Paint(object sender, PaintEventArgs e)
{
  if (band)
  {
    graficoBarras(panel2.CreateGraphics(),A,100,300);
  }
}

1 referencia | 0 cambios | 0 autores, 0 cambios
private void panel3_Paint(object sender, PaintEventArgs e)
{
  if (band)
  {
    graficoPastel(panel3.CreateGraphics(), 50,125);
  }
}
```

Agregamos el siguiente código en el botón limpiar y el código Diagrama

```
1 referencia | 0 cambios | 0 autores, 0 cambios
private void button2_Click(object sender, EventArgs e)
{
  richTextBox1.Clear();
}

1 referencia | 0 cambios | 0 autores, 0 cambios
private void button3_Click(object sender, EventArgs e)
{
  band = true;
  this.Refresh();
}
```

Compilación y revisamos la aplicación.



5. CONCLUSIONES:

- Se utilizo la programación orientada a objetos para manejar la información.
- Se utilizo propiedades nativas de .NET
- Se utiliza posiciones estáticos.

6. RECOMENDACIONES:

- La programación orientada a objetos nos ayuda la manipulación de información.
- La graficación del diagrama nos ayuda a visualizar los valores estadísticos.
- Los eventos de dibujado son personalizables.

7. BIBLIOGRAFIA:

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