



UNIVERSIDAD ISRAEL

CIENCIAS DE LA INGENIERÍA

CARRERA DE SISTEMAS DE INFORMACIÓN

PLATAFORMAS DE DESARROLLO 1

SEMESTRE 2020 B

INFORME DE LABORATORIO 1-S2

TEMA: Aplicaciones Visual C# (Gráficas)

ESTUDIANTE(S): Marco Ayala

CURSO: Séptimo

PARALELO: "A"

PROFESOR: Mg. Luis Fernando Aguas Bucheli

QUITO, 2020

1. TEMA: Aplicaciones Visual C# (Gráficas)

2. OBJETIVOS:

- Adquirir los conceptos básicos relacionados con Visual Studio
- Reconocer las características de Visual Studio

3. INTRODUCCION:

Windows es un entorno gráfico, por lo que se podría suponer que deberíamos tener la posibilidad de trabajar en "modo gráfico" desde dentro de Windows, dibujando líneas, círculos y demás figuras básicas. En efecto, podemos usar las posibilidades de "System.Drawing" para crear una ventana gráfica dentro de nuestro formulario (ventana de programa). Deberemos preparar también las "plumas" ("Pen", para los contornos) y las "brochas" ("Brush", para los rellenos) que queramos usar. Un ejemplo que dibujara una línea roja y una elipse azul cuando pulsásemos un botón del formulario podría ser así:

```
void Button1Click(object sender, EventArgs e)
{
    // Creamos la pluma, el relleno y la ventana gráfica
    System.Drawing.Pen contornoRojo = new System.Drawing.Pen(
        System.Drawing.Color.Red);
    System.Drawing.SolidBrush rellenoAzul = new System.Drawing.SolidBrush(
        System.Drawing.Color.Blue);
    System.Drawing.Graphics ventanaGrafica;
    ventanaGrafica = this.CreateGraphics();

    // Dibujamos
    ventanaGrafica.DrawLine(contornoRojo, 200, 100, 300, 400);
    ventanaGrafica.FillEllipse(rellenoAzul, new Rectangle(0, 0, 200, 300));

    // Liberamos la memoria que habíamos reservado
    contornoRojo.Dispose();
    rellenoAzul.Dispose();
    ventanaGrafica.Dispose();
}
```

Los métodos para dibujar líneas, rectángulos, elipses, curvas, etc. son parte de la clase Graphics. Algunos de los métodos que ésta contiene y que pueden ser útiles para realizar dibujos sencillos son:

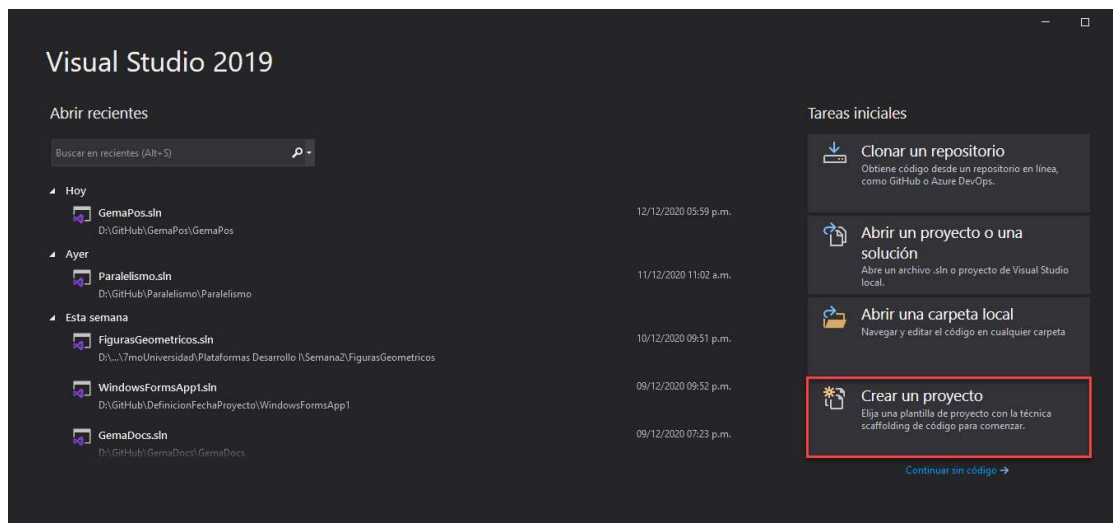
- DrawArc, para dibujar un arco.
- DrawBezier, para una curva spline de Bézier definida por cuatro puntos (estructuras Point).
- DrawClosedCurve, para una curva spline cerrada, a partir de un array de puntos.
- DrawCurve, para una curva.
- DrawEllipse, para dibujar una elipse, a partir del rectángulo que la contiene.
- DrawIcon, para dibujar una imagen representada por un icono (Icon).
- DrawImage, para mostrar una imagen (Image).
- DrawLine, para una línea.
- DrawPolygon, para un polígono, a partir de un array de puntos.



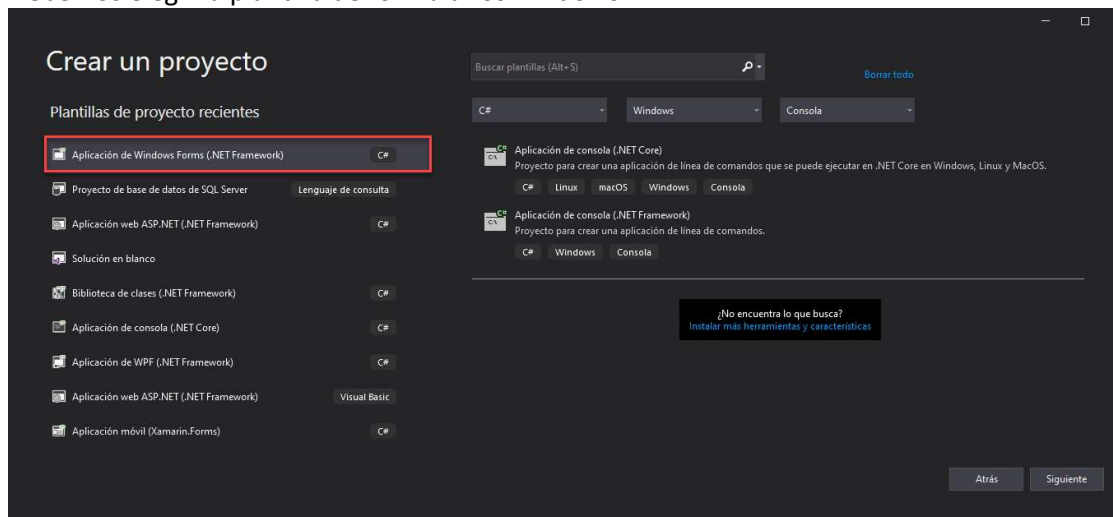
- DrawRectangle, para un rectángulo.
- DrawString, para mostrar una cadena de texto.
- FillEllipse, para rellenar el interior de una elipse.
- FillPolygon, para rellenar el interior de un polígono.
- FillRectangle, para rellenar el interior de un rectángulo.

4. DESARROLLO:

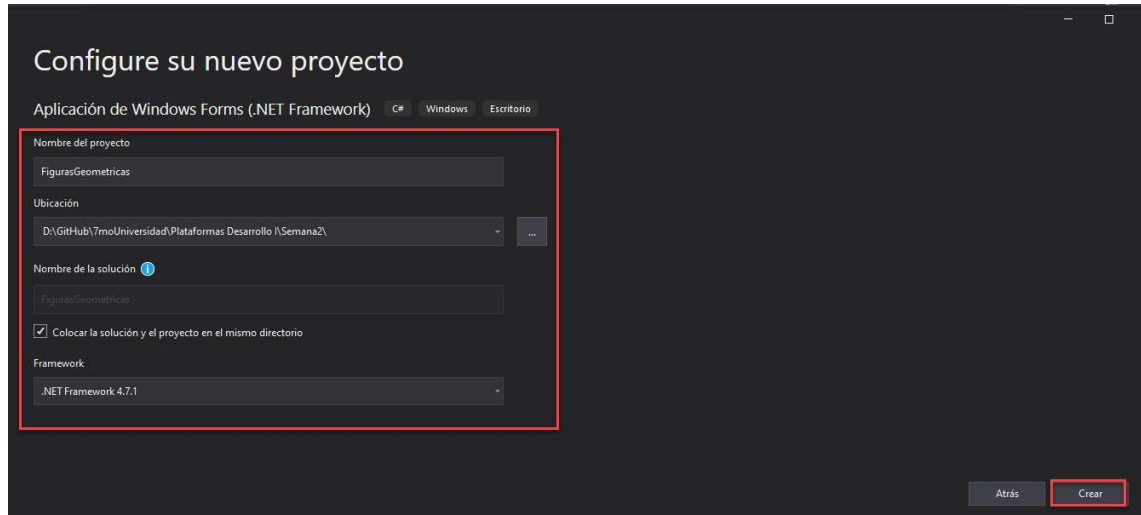
Abrir el Visual Studio 2019 y elegir la opción de crear proyecto.



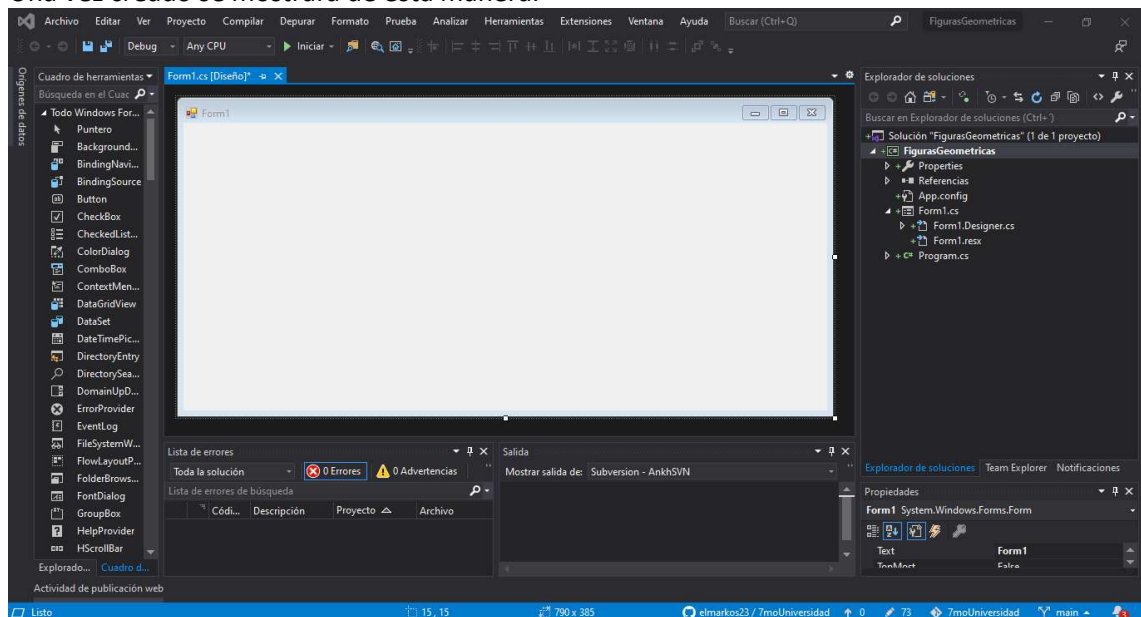
Debemos elegir la plantilla de formularios windows



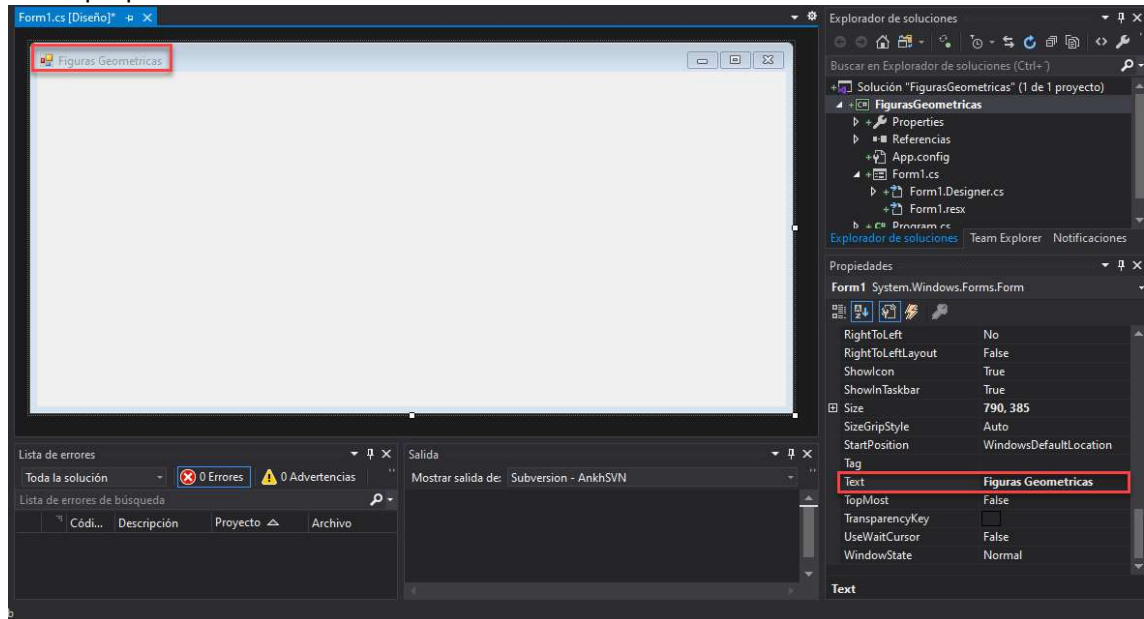
En esta pantalla Ingresamos el nombre del proyecto que vamos a crear, seleccionamos la ubicación donde vamos a grabar, seleccionamos el framework y seleccionamos finalmente boton Crear.



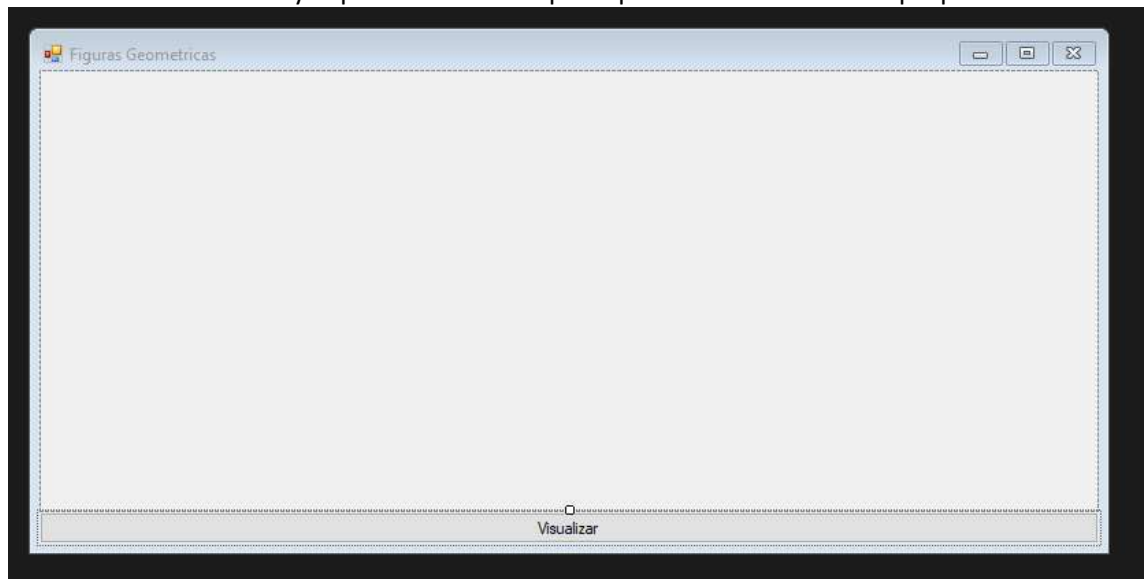
Una vez creado se mostrará de esta manera.



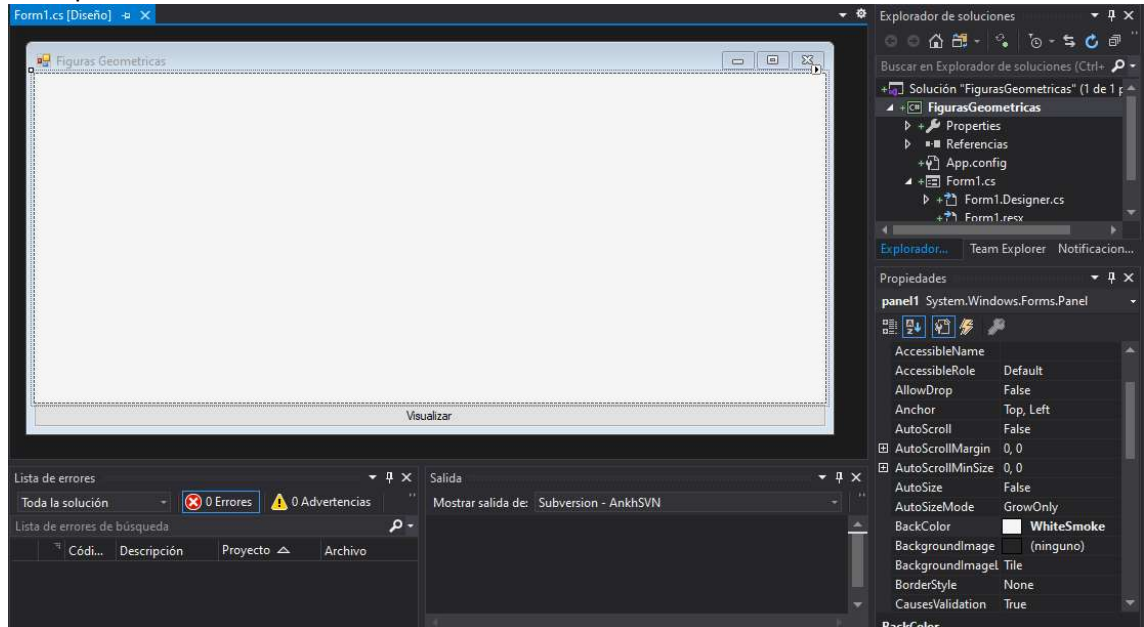
En las propiedades del formulario e cambiamos el nombre.



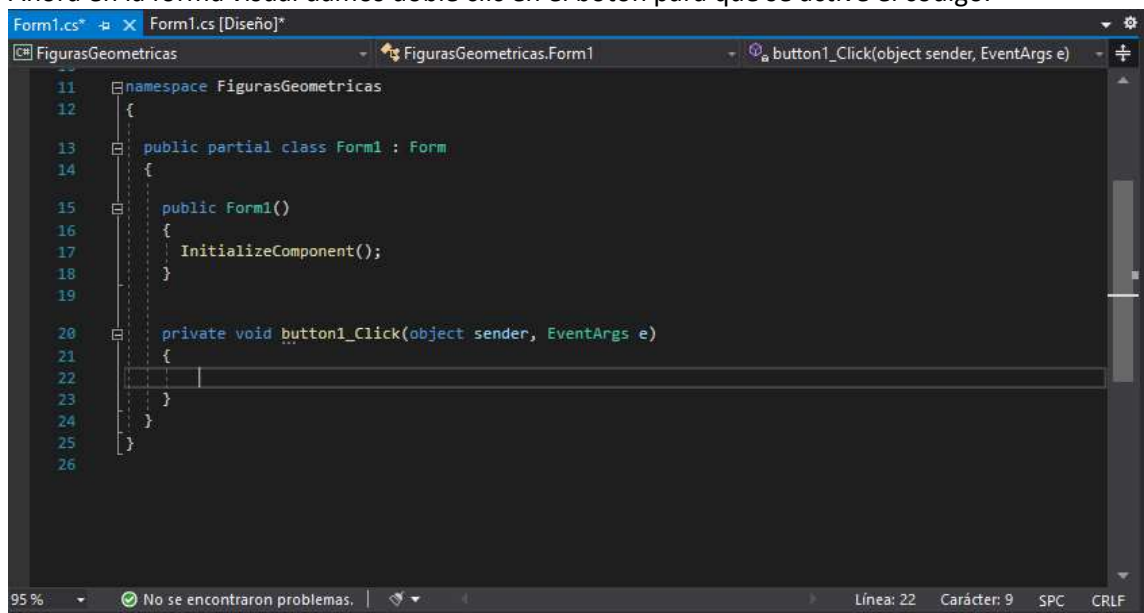
Ahora al formulario le agregamos 2 controles un panel y botón, en nuestro caso le ponemos al botón dock en botón y al panel en dock fill para que se rellene de forma proporcional.



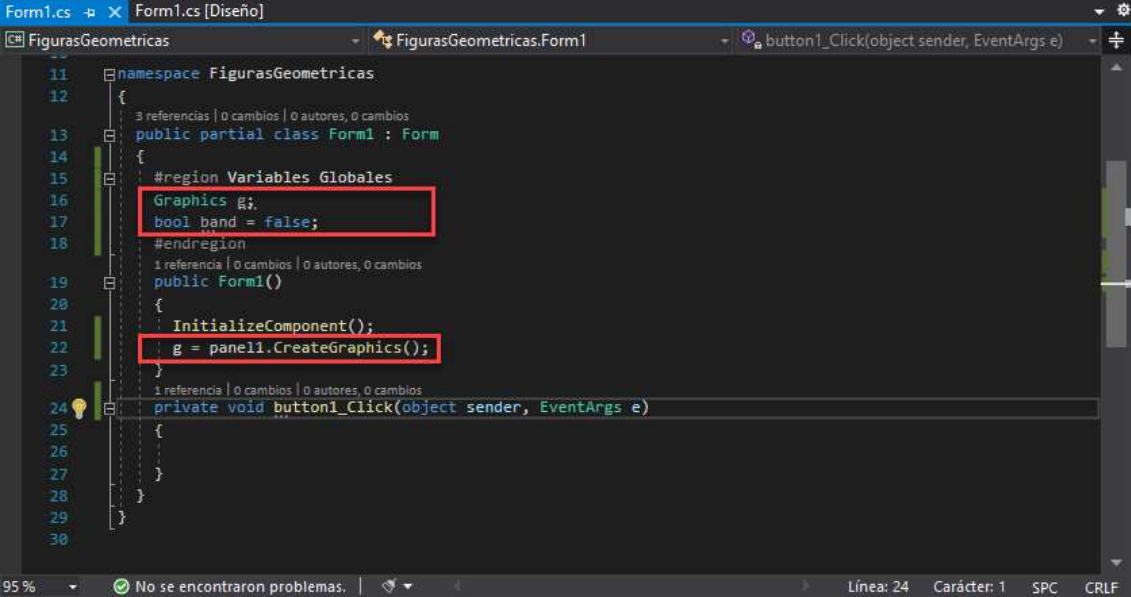
Seleccionamos el panel y en las propiedades le cambiamos el color de Backcolor a White o Transparent



Ahora en la forma visual damos doble clic en el botón para que se active el código.

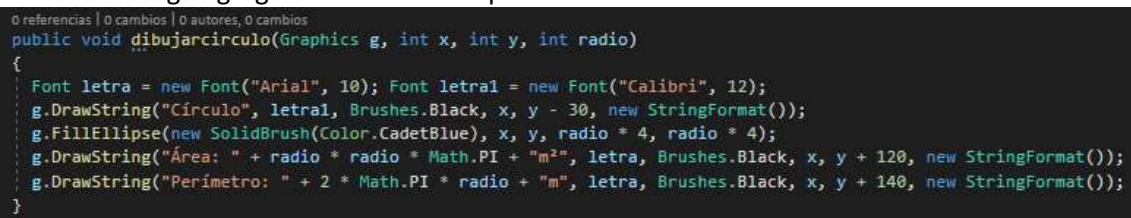


Ahora declaramos las variables globales e inicializamos la variable grafica g.



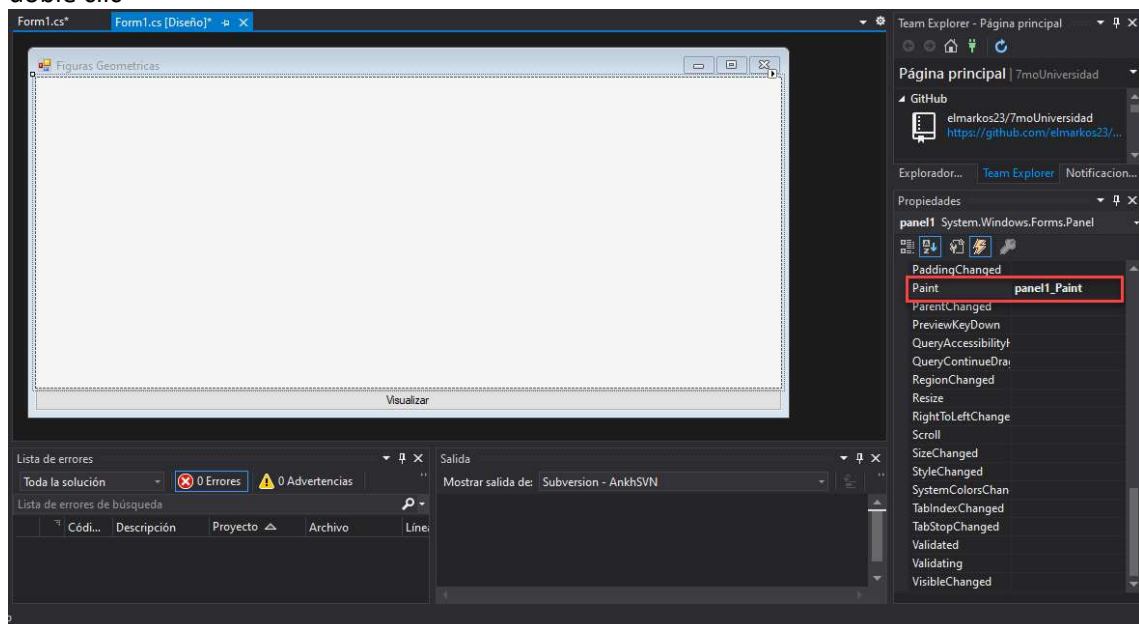
```
11 namespace FigurasGeometricas
12 {
13     3 referencias | 0 cambios | 0 autores, 0 cambios
14     public partial class Form1 : Form
15     {
16         #region Variables Globales
17         Graphics g;
18         bool band = false;
19         #endregion
20     }
21     1 referencia | 0 cambios | 0 autores, 0 cambios
22     public Form1()
23     {
24         InitializeComponent();
25         g = panel1.CreateGraphics();
26     }
27     1 referencia | 0 cambios | 0 autores, 0 cambios
28     private void button1_Click(object sender, EventArgs e)
29     {
30     }
```

Ahora en código agregamos una función para crear el círculo



```
0 referencias | 0 cambios | 0 autores, 0 cambios
public void dibujarcirculo(Graphics g, int x, int y, int radio)
{
    Font letra = new Font("Arial", 10); Font letra1 = new Font("Calibri", 12);
    g.DrawString("Círculo", letra, Brushes.Black, x, y - 30, new StringFormat());
    g.FillEllipse(new SolidBrush(Color.CadetBlue), x, y, radio * 4, radio * 4);
    g.DrawString("Área: " + radio * radio * Math.PI + "m²", letra, Brushes.Black, x, y + 120, new StringFormat());
    g.DrawString("Perímetro: " + 2 * Math.PI * radio + "m", letra, Brushes.Black, x, y + 140, new StringFormat());
}
```

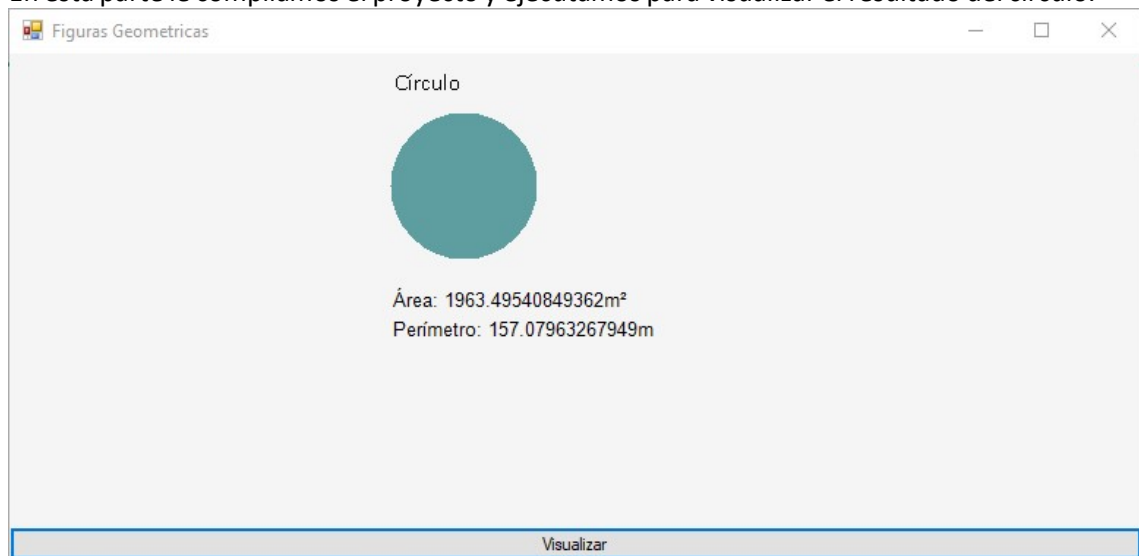
Ahora en la parte visual seleccionamos el panel y nos dirigimos a los eventos de Paint y hacer doble clic



Ahora agregamos el siguiente código en mismo que dibujará el círculo en el panel

```
1 referencia | 0 cambios | 0 autores, 0 cambios
private void panel1_Paint(object sender, PaintEventArgs e)
{
    if (band)
    {
        int x = 20, y = 40;
        dibujarcirculo(g, x + 240, y, 25);
    }
}
```

En esta parte le compilamos el proyecto y ejecutamos para visualizar el resultado del círculo.



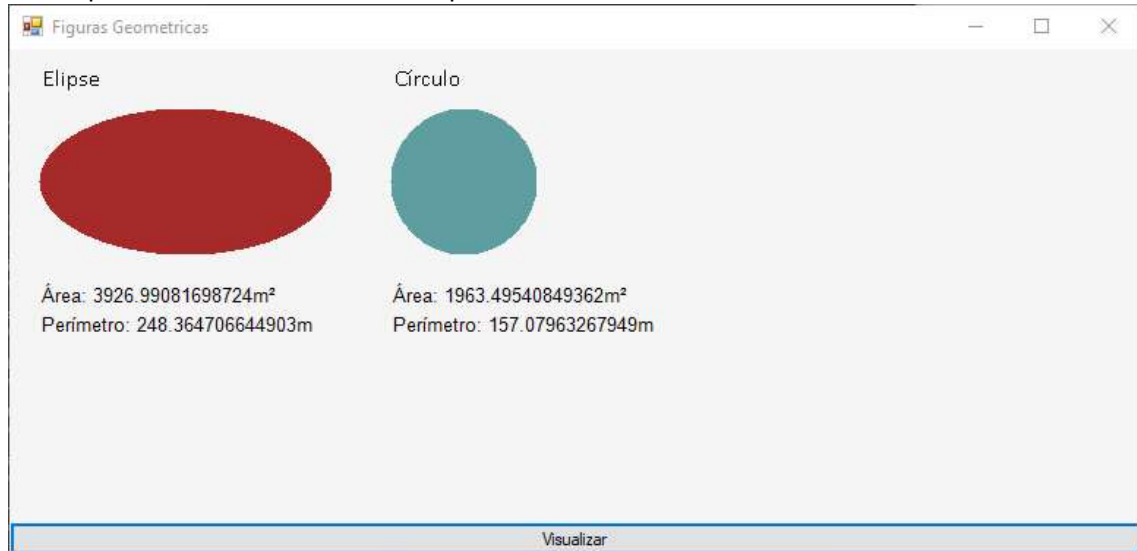
Ahora agregamos otra función para dibujar la elipse.

```
1 referencia | 0 cambios | 0 autores, 0 cambios
public void dibujarelipse(Graphics g, int x, int y, int radio1, int radio2)
{
    Font letra = new Font("Arial", 10); Font letra1 = new Font("Calibri", 12);
    g.DrawString("Elipse", letra1, Brushes.Black, x, y - 30, new StringFormat());
    g.FillEllipse(new SolidBrush(Color.Brown), x, y, radio1 * 4, radio2 * 4);
    g.DrawString("Área: " + radio1 * radio2 * Math.PI + "m²", letra, Brushes.Black, x, y + 120, new StringFormat());
    g.DrawString("Perímetro: " + 2 * Math.PI * Math.Pow((Math.Pow(radio1, 2) + Math.Pow(radio2, 2)) / 2, 0.5) + "m", letra, Brushes.Black, x, y + 140, new StringFormat());
}
```

A continuación nos dirigimos al método de Paint y le agregamos la función con los valores

```
1 referencia | 0 cambios | 0 autores, 0 cambios
private void panel1_Paint(object sender, PaintEventArgs e)
{
    if (band)
    {
        int x = 20, y = 40;
        dibujarcirculo(g, x + 240, y, 25);
        dibujarelipse(g, x, y, 50, 25);
    }
}
```


Ahora podemos visualizar nuestra elipse.



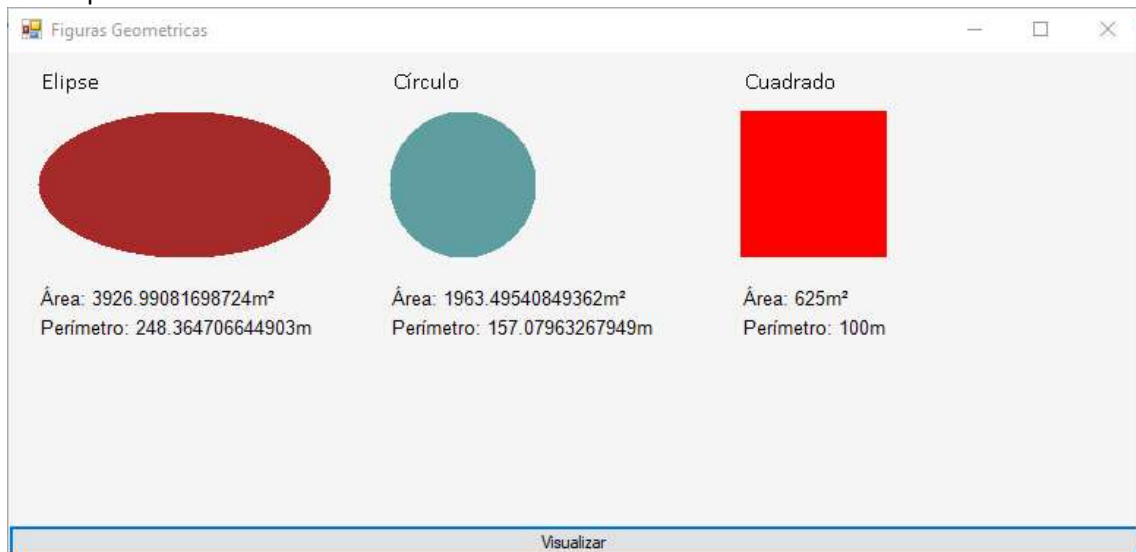
Ahora agregamos otra función para dibujar el cuadrado.

```
0 referencias | 0 cambios | 0 autores, 0 cambios
public void dibujarcuadrado(Graphics g, int x, int y, int lado)
{
    Font letra = new Font("Arial", 10);
    Font letra1 = new Font("Calibri", 12);
    g.DrawString("Cuadrado", letra1, Brushes.Black, x, y - 30, new StringFormat());
    g.FillRectangle(new SolidBrush(Color.Red), x, y, lado * 4, lado * 4);
    g.DrawString("Área: " + lado * lado + "m²", letra, Brushes.Black, x, y + 120, new StringFormat());
    g.DrawString("Perímetro: " + lado * 4 + "m", letra, Brushes.Black, x, y + 140, new StringFormat());
}
```

A continuación nos dirigimos al método de Paint y le agregamos la función con los valores

```
1 referencia | 0 cambios | 0 autores, 0 cambios
private void panell1_Paint(object sender, PaintEventArgs e)
{
    if (band)
    {
        int x = 20, y = 40;
        dibujarcirculo(g, x + 240, y, 25);
        dibujarelipse(g, x, y, 50, 25);
        dibujarcuadrado(g, x + 480, y, 25);
    }
}
```

Ahora podemos visualizar nuestro cuadrado.



Ahora agregamos otra función para dibujar el rectángulo.

```
0 referencias | 0 cambios | 0 autores, 0 cambios
public void dibujarrectagulo(Graphics g, int x, int y, int a, int b)
{
    Font letra = new Font("Arial", 10); Font letra1 = new Font("Calibri", 12);
    g.DrawString("Rectángulo", letra1, Brushes.Black, x, y - 30, new StringFormat());
    g.FillRectangle(new SolidBrush(Color.Yellow), x, y, a * 4, b * 4);
    g.DrawString("Área: " + b * a + "m²", letra, Brushes.Black, x, y + 120, new StringFormat());
    g.DrawString("Perímetro: " + 2 * b + 2 * a + "m", letra, Brushes.Black, x, y + 140, new StringFormat());
}
```

A continuación nos dirigimos al método de Paint y le agregamos la función con los valores

```
1 referencia | 0 cambios | 0 autores, 0 cambios
private void panell1_Paint(object sender, PaintEventArgs e)
{
    if (band)
    {
        int x = 20, y = 40;
        dibujarcirculo(g, x + 240, y, 25);
        dibujarelipse(g, x, y, 50, 25);
        dibujarcuadrado(g, x + 480, y, 25);
        dibujarrectagulo(g, x, y + 240, 50, 25);
    }
}
```

Ahora podemos visualizar nuestro rectángulo.



5. CONCLUSIONES:

- En Visual Studio Soporta librerías gráficas que nos ayuda a crear objetos personalizados.
- Con C# es un lenguaje super robusto.
- Con C# se puede crear aplicaciones de diferentes tipos.

6. RECOMENDACIONES:

- Investigar las opciones avanzadas de visual studio.

7. BIBLIOGRAFIA:

Libro electrónico: C Sharp (2000). C Sharp. Recuperado de https://es.wikipedia.org/wiki/C_Sharp

Libro electrónico: Desarrollo de Aplicaciones .NET
<https://channel9.msdn.com/Series/CSharp-101/What-is-C/player>

Video: Que es C#? <https://channel9.msdn.com/Series/CSharp-101/What-is-C/player>

Libro electrónico: Clase Gráfica Recuperado de <https://docs.microsoft.com/en-us/dotnet/api/system.drawing.graphics?view=dotnet-plat-ext-5.0>