



UNIVERSIDAD ISRAEL

CIENCIAS DE LA INGENIERÍA

CARRERA DE SISTEMAS DE INFORMACIÓN

PLATAFORMAS DE DESARROLLO 1

SEMESTRE 2020 B

INFORME DE LABORATORIO S6

TEMA: Aplicaciones C# (Funciones)

ESTUDIANTE(S): Marco Antonio Ayala Lituma

CURSO: Séptimo

PARALELO: "A"

PROFESOR: Mg. Luis Fernando Aguas Bucheli

QUITO, 2020

1. TEMA: Aplicaciones C# (Funciones)

2. OBJETIVOS:

- Adquirir los conceptos básicos relacionados con C#
- Reconocer las características de C#
- Manipular controles básicos de la herramienta
- Tener acceso a pantallas externas

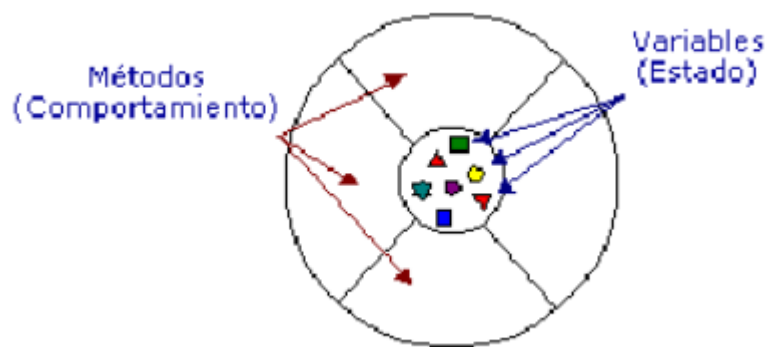
3. INTRODUCCION:

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

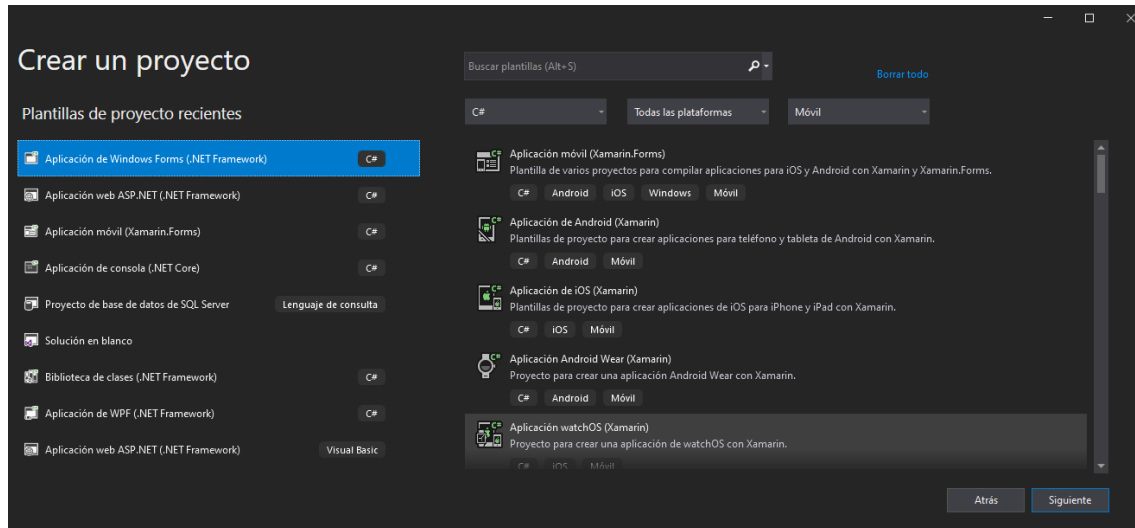
Una clase está formada por los métodos y las variables que definen las características comunes a todos los objetos de esa clase. Precisamente la clave de la OOP está en abstraer los métodos y los datos comunes a un conjunto de objetos y almacenarlos en una clase.

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

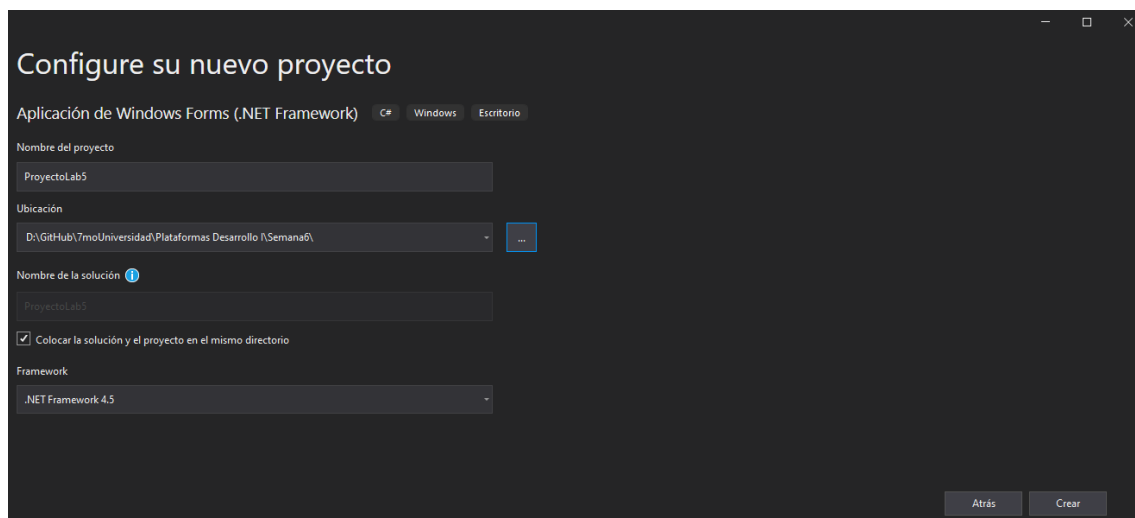


4. DESARROLLO:

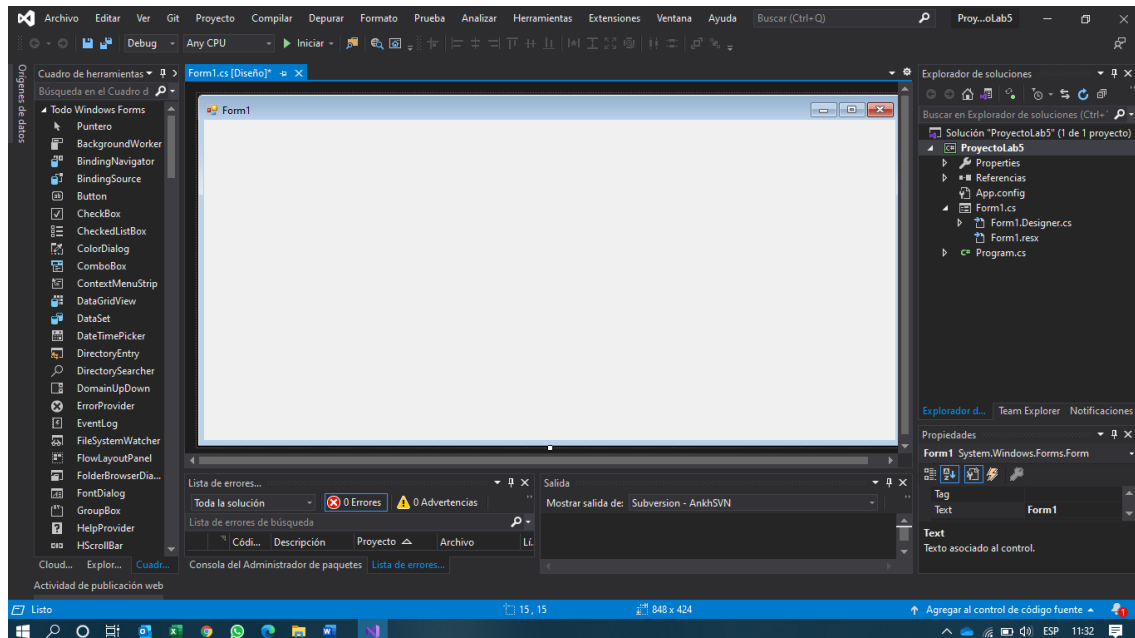
Seleccionamos el tipo de proyecto de Windows forms



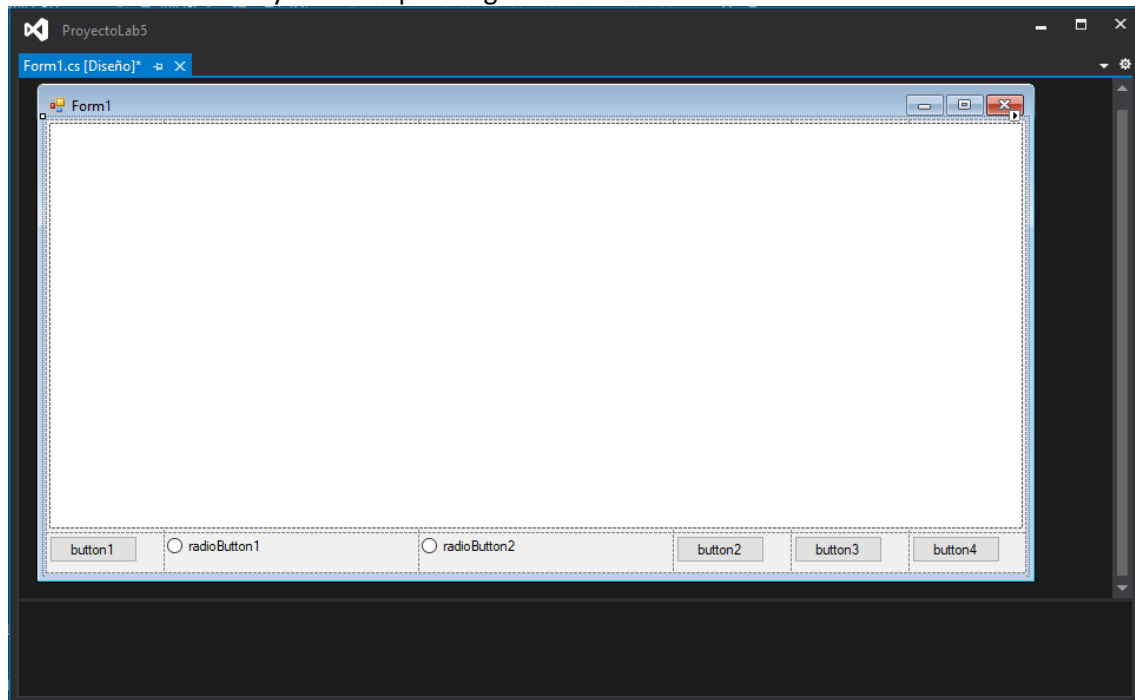
Le colocamos el nombre y le ubicamos en la carpeta para grabar



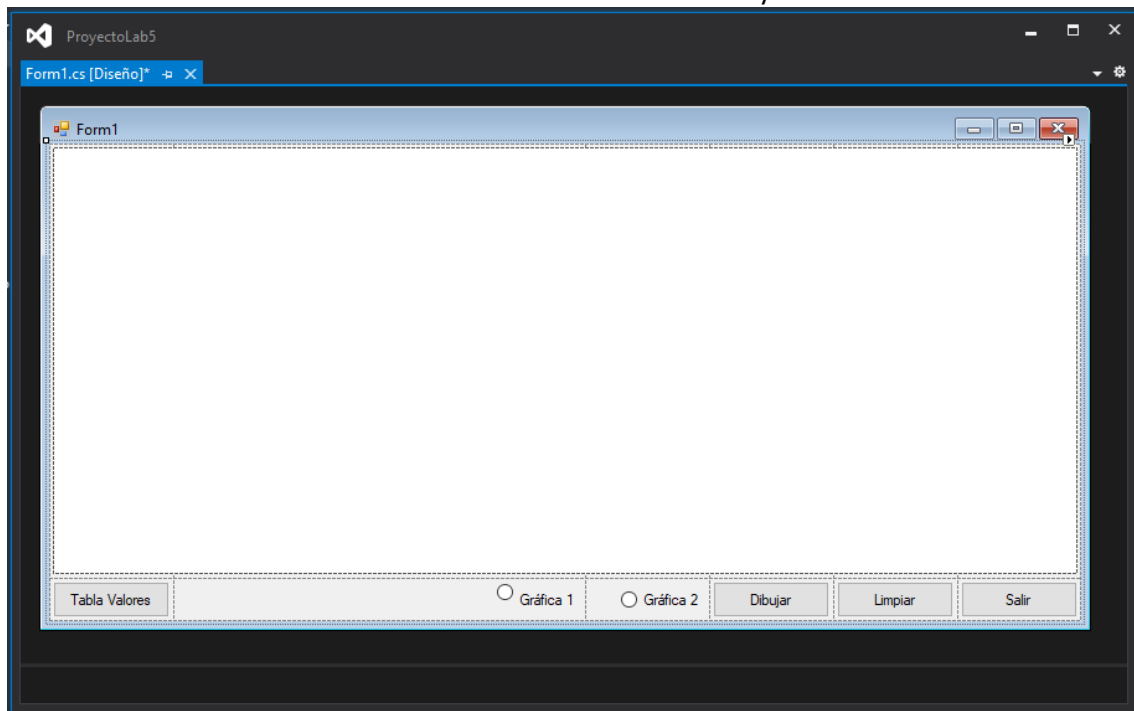
El proyecto creado quedaría así.



Ahora agregamos los siguientes controles en el formulario form1, en este caso también colocamos un `TableLayoutPanel` para organizar de forma automática los controles.



Ahora le colocamos los nombres a los controles como botones y radio botones



Ahora en el Form1 agregamos estas variables globales.

```
1 referencia | 0 cambios | 0 autores, 0 cambios  
public Form1()  
{  
    InitializeComponent();  
}  
double[] valores = new double[20000];  
double[] valores1 = new double[20000];
```

Le agregamos el siguiente código al botón dibujar

```
1 referencia | 0 cambios | 0 autores, 0 cambios
private void button2_Click(object sender, EventArgs e)
{
    Graphics dibujo = pictureBox1.CreateGraphics();//parte grafica del picture
    Pen lapiz = new Pen(Color.Black);
    Pen lapiz1 = new Pen(Color.Olive, 3);
    Pen lapiz2 = new Pen(Color.Aqua, 3);
    int xcentro = pictureBox1.Width / 2;
    int ycentro = pictureBox1.Height / 2;
    int cortes = 30;//numero de cortes en el eje x

    double n1 = 1.50;//cambio de valores de n1
    double n2 = 1.01;//cambio de valores de n2
    dibujo.TranslateTransform(xcentro, ycentro);//trasladamos las coordenadas al centro
    dibujo.ScaleTransform(1, -1);//Convertimos a coordenadas normales
    //dibujamos ejes x-y
    dibujo.DrawLine(lapiz, xcentro * -1, 0, xcentro * 2, 0);//ejex
    dibujo.DrawLine(lapiz, 0, ycentro, 0, ycentro * -1);//ejey
    //dibujamos puntos +x
    int a;
    for (a = 0; a <= xcentro * 2; a += (pictureBox1.Width / cortes * 2))
    {
        dibujo.DrawLine(lapiz, a, 5, a, -5);
    }
    // dibujamos puntos -x
    for (a = 0; a <= xcentro * -1; a += (-1 * pictureBox1.Width / cortes * 2))
    {
        dibujo.DrawLine(lapiz, a, 5, a, -5);
    }
    double xx, puntoX1, puntoY1, puntoX2, puntoY2;
    int con = 0;
    //dibujamos la gráfica 1
    if (radioButton1.Checked == true)
    {
        for (xx = (pictureBox1.Width / 2) * -1; xx <= (pictureBox1.Width / 2) * 2; xx += 0.1)
        {
            //Escribimos la función a graficar
            valores[con] = System.Math.Abs((n1 * System.Math.Cos(xx) - System.Math.Sqrt(n2 * n2 - n1 * n1 *
                System.Math.Sin(xx) * System.Math.Sin(xx))) / (n1 * System.Math.Cos(xx) +
                System.Math.Sqrt(n2 * n2 - n1 * n1 * System.Math.Sin(xx) * System.Math.Sin(xx))));
            if (valores[con].ToString().Contains("N") == true)//en casode que existan números imaginarios
                valores[con] = 1.0; con++;
        }

        con = 1;
        for (xx = (xcentro * -1) + 0.1; xx <= xcentro * 2; xx += 0.1)
        {
            //iniciamos una cordenada X adelante
            //sacamos coordenadas 1
            puntoX1 = (xx - 0.1) * (pictureBox1.Width / (cortes * 2));
            puntoY1 = valores[con - 1] * ycentro;
            puntoY2 = valores[con] * ycentro;
            //sacamos coordenadas 2
            puntoX2 = xx * (pictureBox1.Width / (cortes * 2));
            dibujo.DrawLine(lapiz1, Convert.ToSingle(puntoX1), Convert.ToSingle(puntoY1), Convert.ToSingle(puntoX2), Convert.ToSingle(puntoY2));
            con++;
        }
    }
}
```

Continuamos con el código

```
if (radioButton2.Checked == true)
{
    //grafica 2
    con = 0;
    for (double xx1 = (pictureBox1.Width / 2) * -1; xx1 <= (pictureBox1.Width / 2) * 2; xx1 += 0.1)
    {
        //Escribimos la función a graficar
        valores1[con] = Math.Cos(xx1);
        if (valores1[con].ToString().Contains("N") == true)
        {
            valores1[con] = 1.0;
        }
        con = con + 1;
    }
    con = 1;
    for (double xx1 = (pictureBox1.Width / 2) * -1; xx1 <= (pictureBox1.Width / 2) * 2; xx1 += 0.1)
    {
        //iniciamos una cordenada X adelante
        //sacamos coordenadas 1
        puntoX1 = (xx1 - 0.1) * (pictureBox1.Width / (cortes) * 2);
        puntoY1 = valores1[con - 1] * ycentro;
        //sacamos coordenadas 2
        puntoX2 = xx1 * (pictureBox1.Width / (cortes) * 2);
        puntoY2 = valores1[con] * ycentro;
        dibujo.DrawLine(lapiz2, Convert.ToSingle(puntoX1), Convert.ToSingle(puntoY1), Convert.ToSingle(puntoX2), Convert.ToSingle(puntoY2));
        con = con + 1;
    }
}
```

Ahora creamos la siguiente funcion

```
1 referencia | 0 cambios | 0 autores, 0 cambios
public void EncerarPctGraf()
{
    Graphics Lienzo = pictureBox1.CreateGraphics(); Lienzo.Clear(pictureBox1.BackColor); //Se borra el gráfico
    Lienzo.Dispose();
}
```

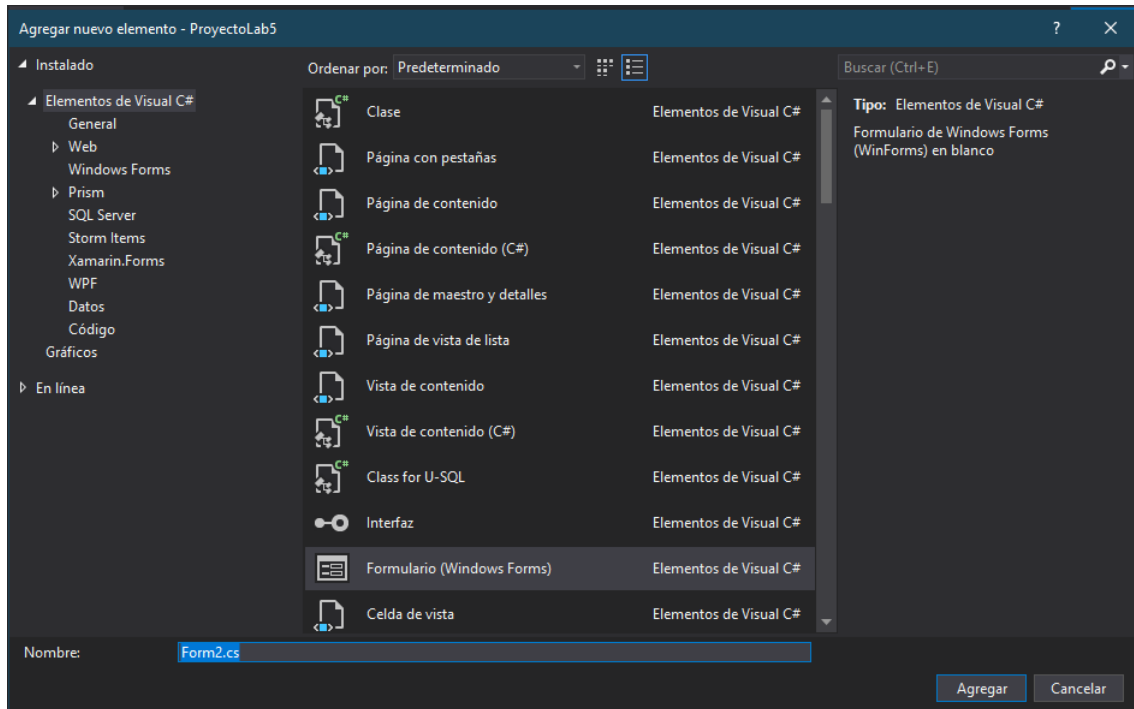
La función creada le colocamos en el botón limpiar

```
1 referencia | 0 cambios | 0 autores, 0 cambios
private void button3_Click(object sender, EventArgs e)
{
    EncerarPctGraf();
}
```

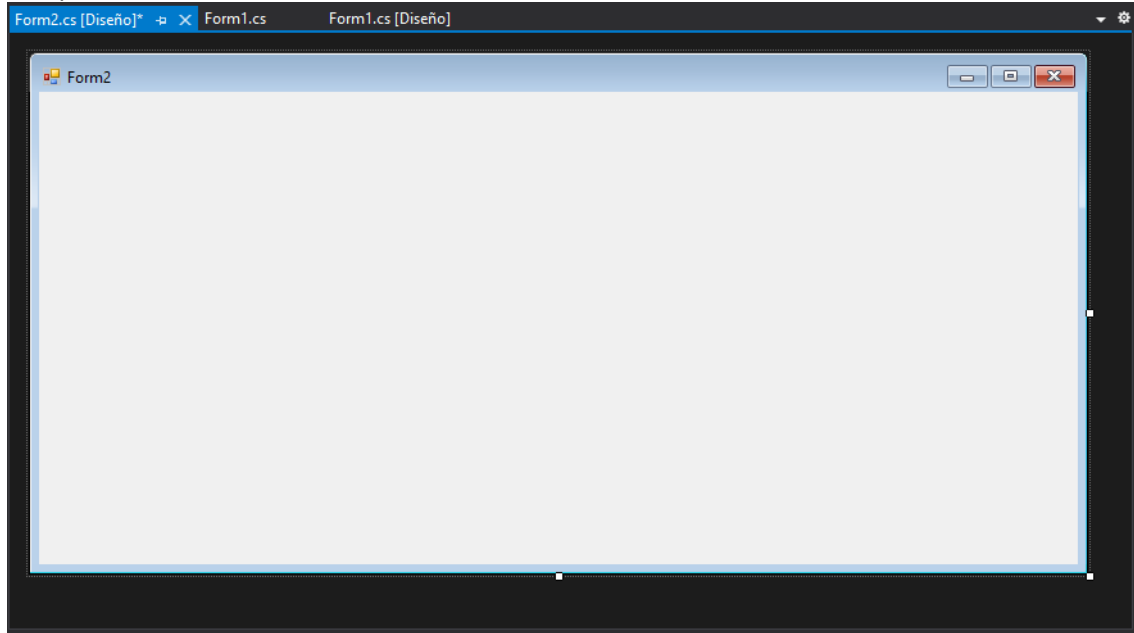
Ahora Botón Salir le agregamos el siguiente código.

```
1 referencia | 0 cambios | 0 autores, 0 cambios
private void button4_Click(object sender, EventArgs e)
{
    this.Visible = false;
}
```

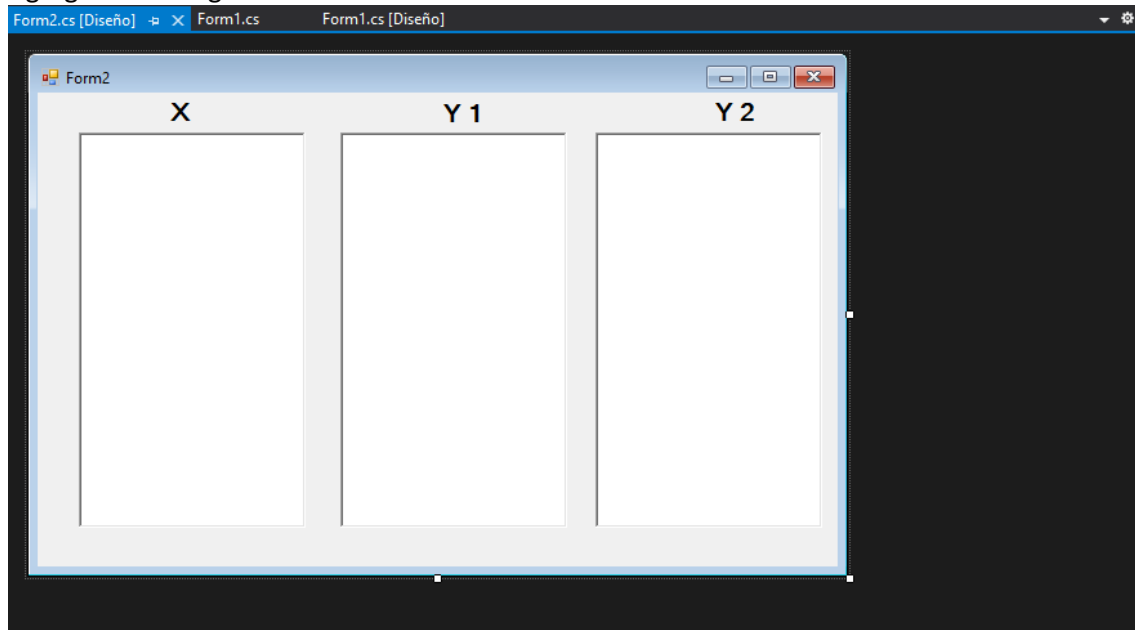
Ahora creamos un nuevo Form2



Así queda nuestro formulario vacío



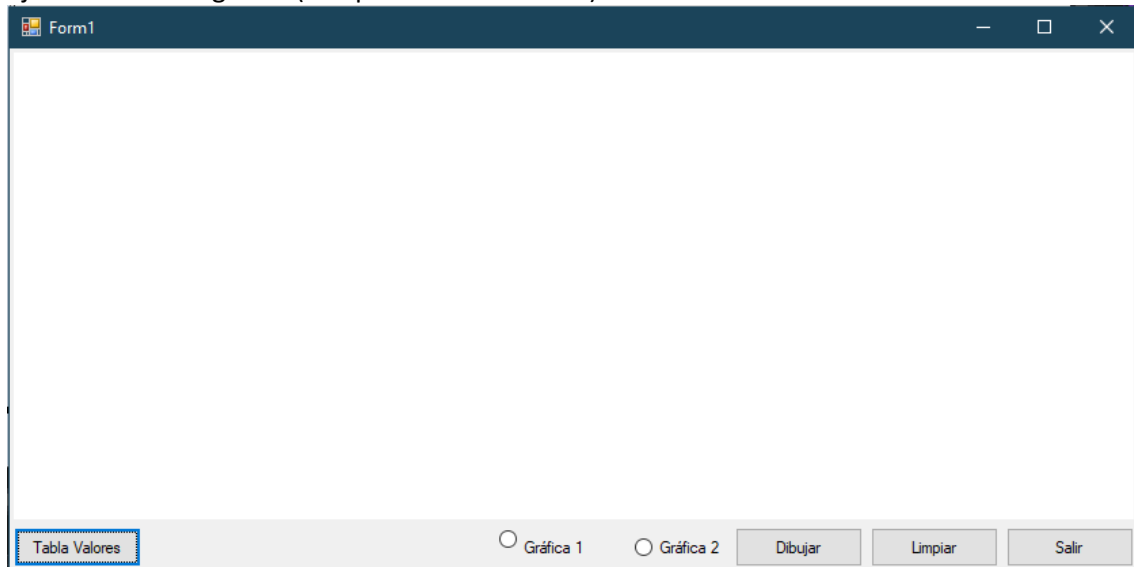
Agregamos los siguientes controles RichTextBox



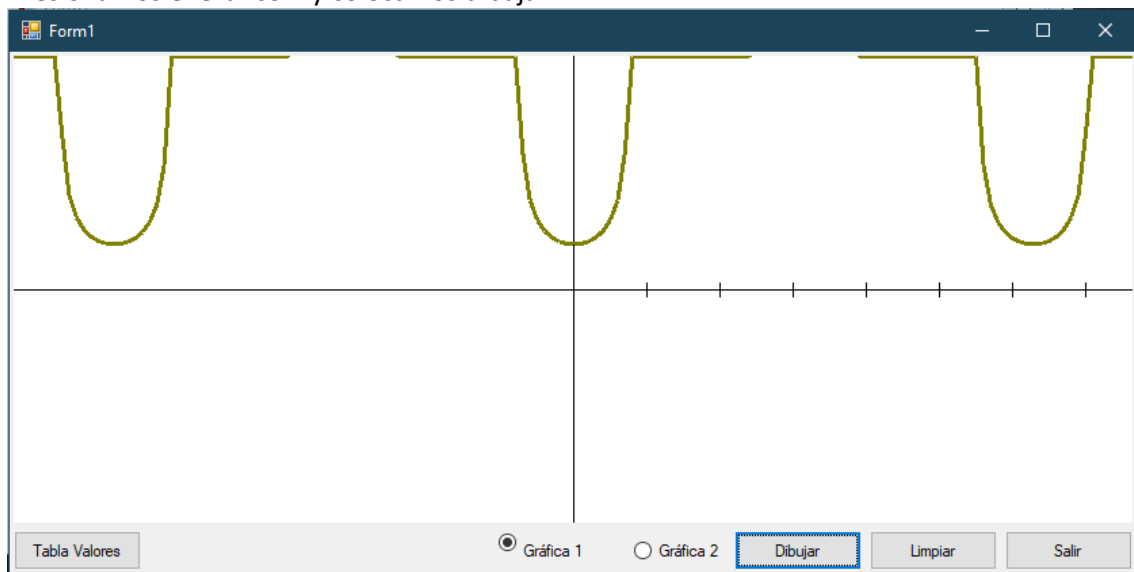
Ahora en el botón Tabla de Valores del Form1 le agregamos del código .

```
1 referencia | 0 cambios | 0 autores, 0 cambios
private void button1_Click(object sender, EventArgs e)
{
    Form2 m = new Form2(); for (int c = 0; c < 90; c++)//visualizacion de los primero valores
    {
        m.richTextBox1.AppendText("\n" + c);
        m.richTextBox2.AppendText("\n" + Math.Round(valores[c], 5));
        m.richTextBox3.AppendText("\n" + Math.Round(valores1[c], 5));
    }
    m.Show();
}
```

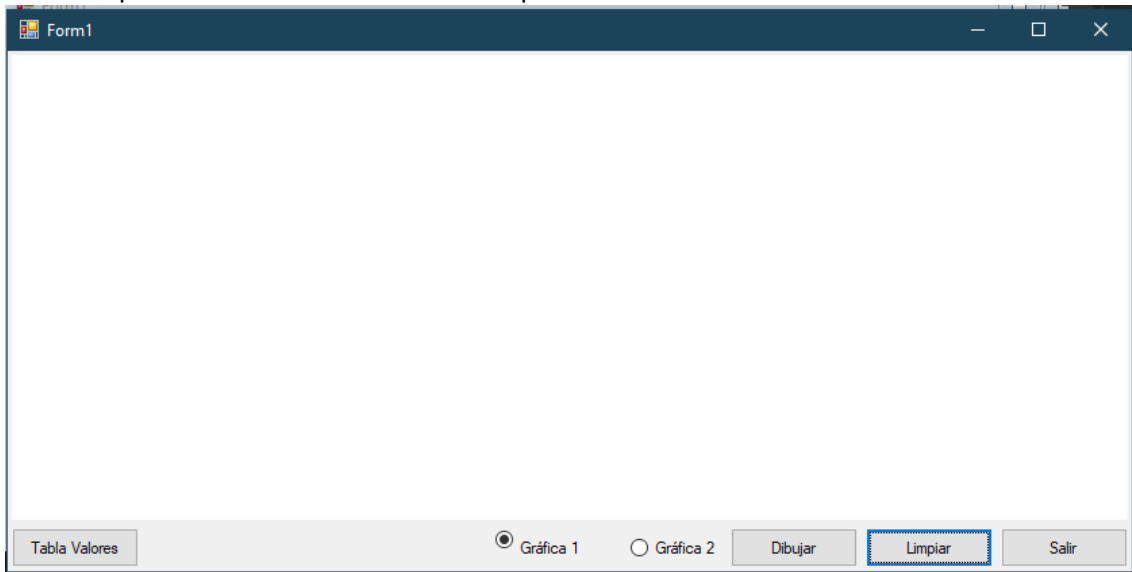
Ejecución del Programa (compilamos e iniciamos)



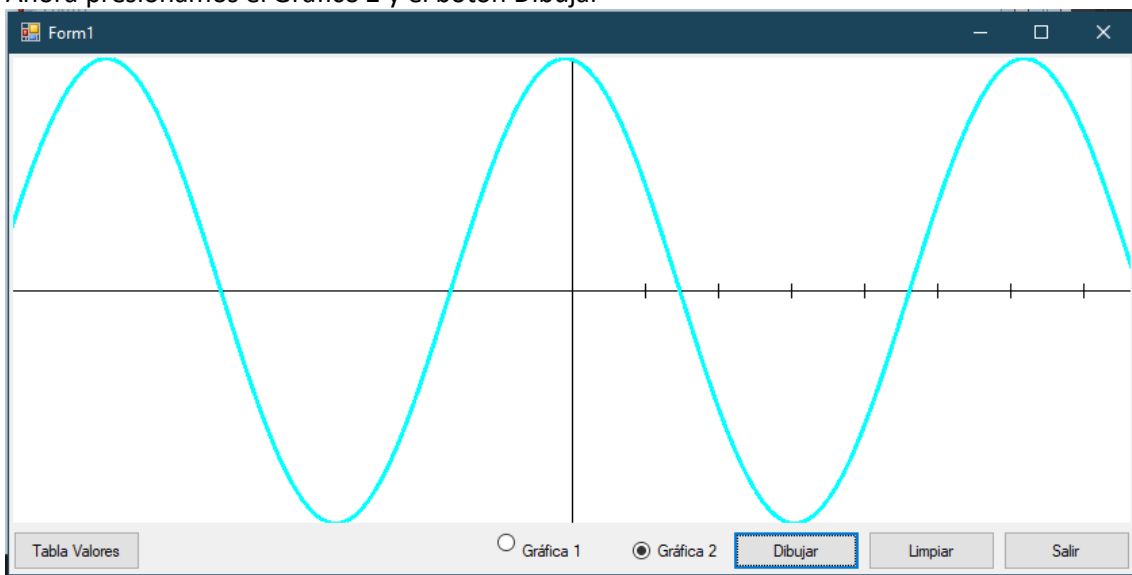
Presionamos el Grafico 1 y colocamos dibujar



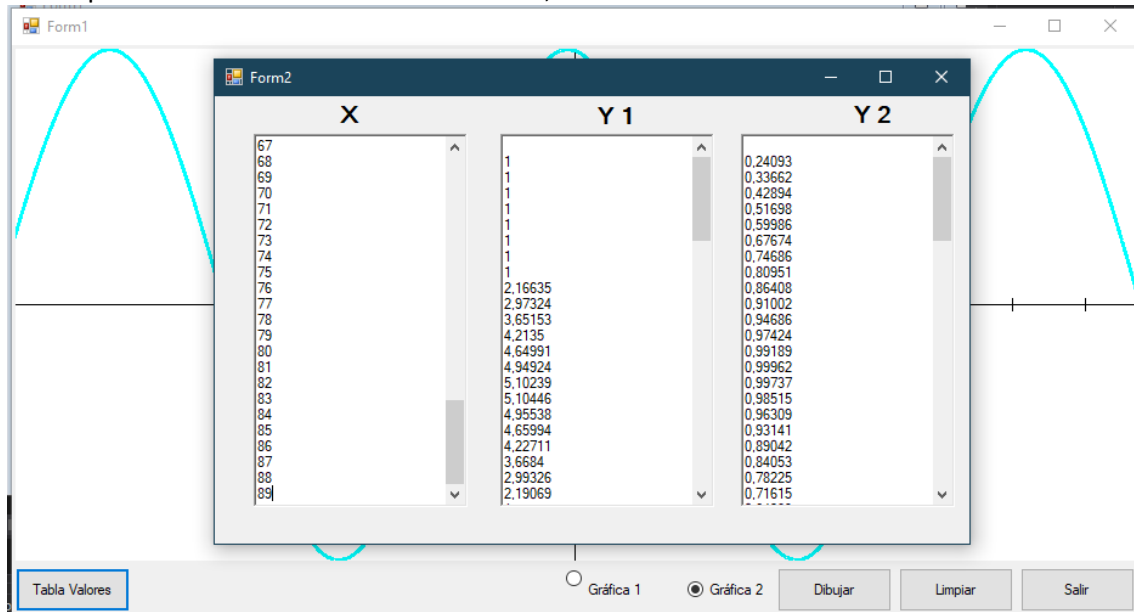
Ahora la prueba Presionamos el Botón Limpiar



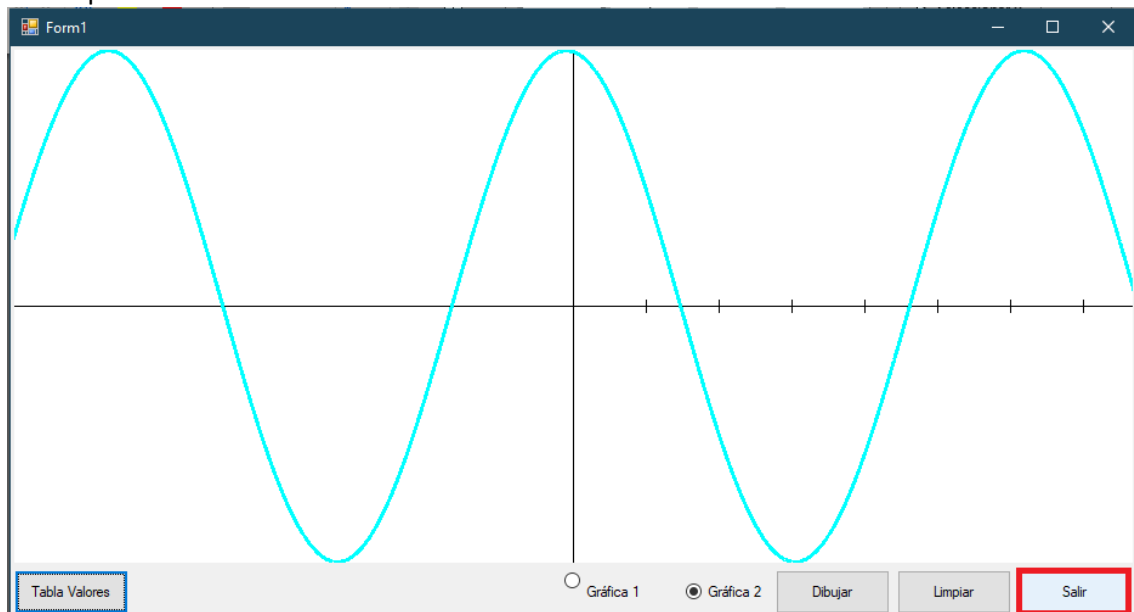
Ahora presionamos el Grafico 2 y el botón Dibujar



Ahora podemos Visualizar la tabla de Valores, Presionando el Botón



Ahora podemos Probar el Botón Salir



5. CONCLUSIONES:

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

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