LENGUAJE VISUAL C# (Sharp)
Realizado por: Manuel Flores

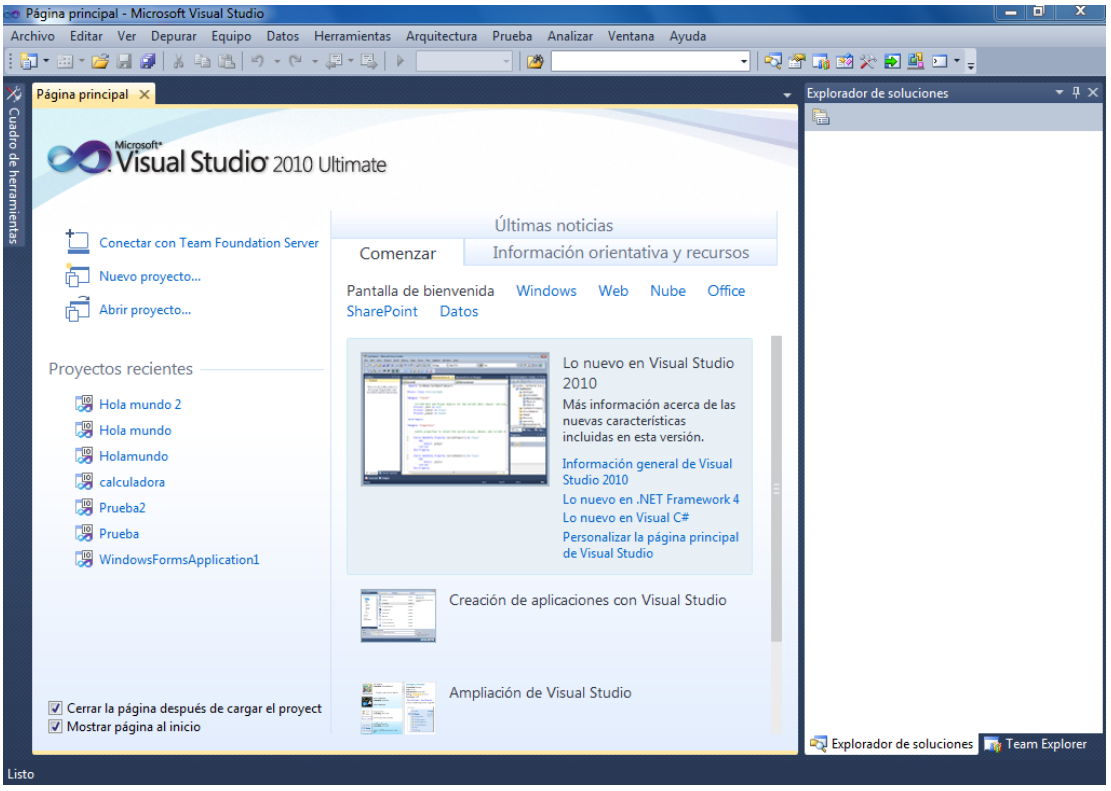
1. Introducción

El lenguaje visual C# es la evolución de Microsoft a su lenguaje C++ y fue creado en torno al año 2000. Es un lenguaje de programación orientado a objetos y esto significa que trabajaremos con entidades, como botones, etiquetas, entre otras, que llamaremos objetos.

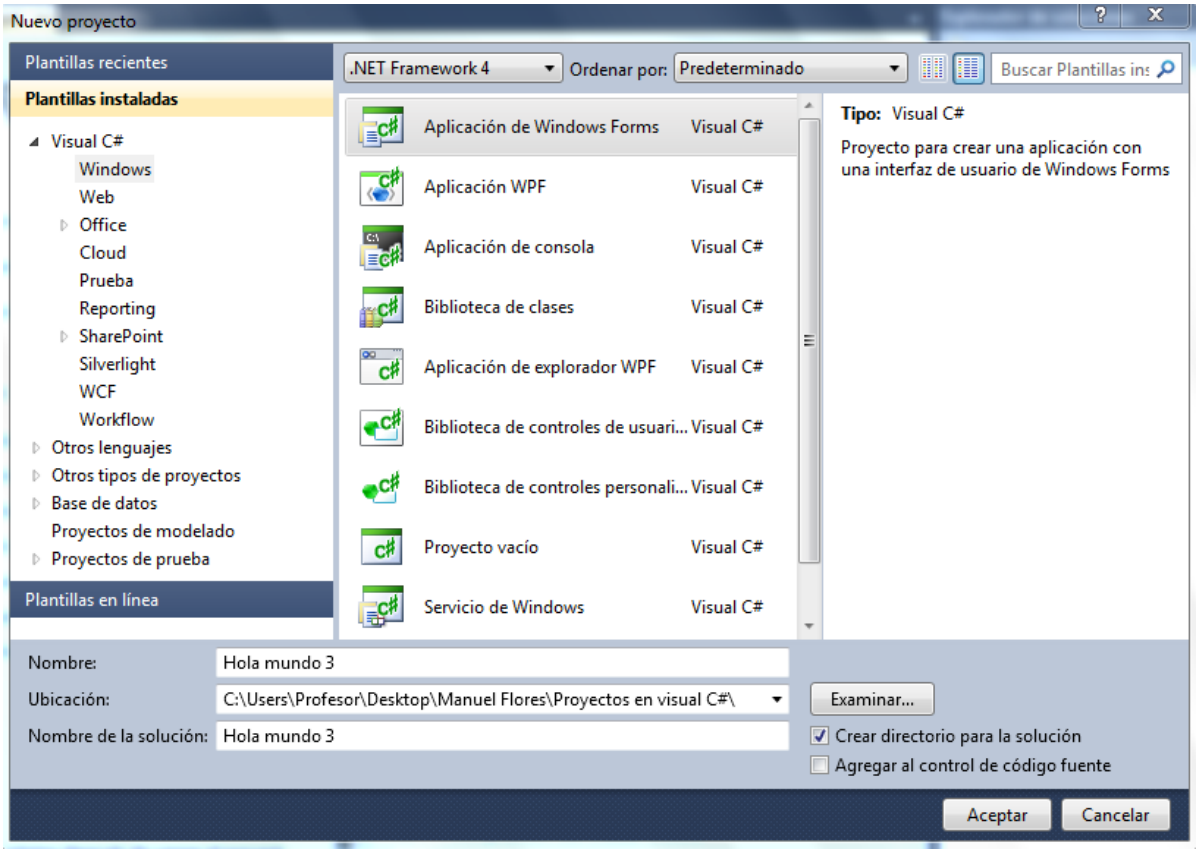
Utilizaremos la aplicación Microsoft Visual Studio 2010 para la realización de los programas.

2. Primer programa en visual c#. Hola mundo.

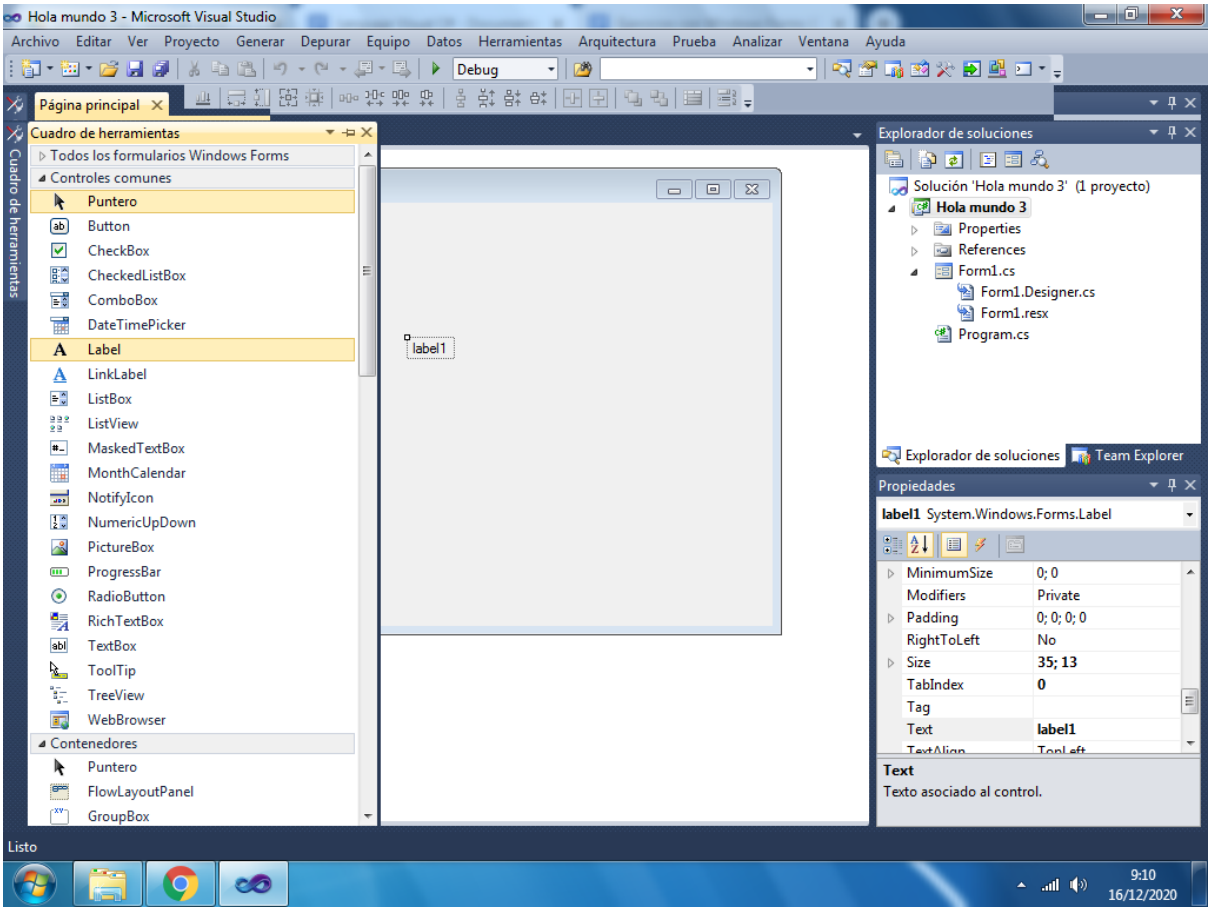
Abrimos Visual Studio 2010:



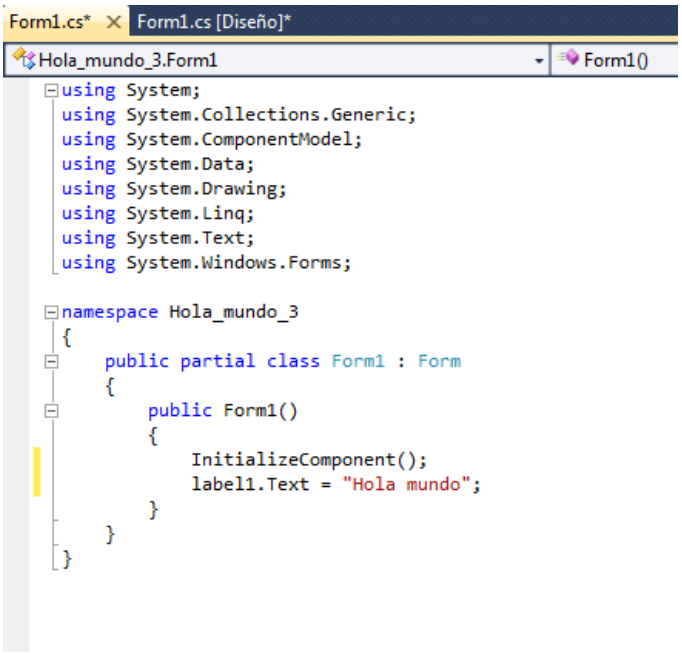
Pulsamos Archivo -> Nuevo -> Proyecto y le damos nombre a nuestro primer proyecto que será un proyecto Aplicación Windows Forms Visual C#.



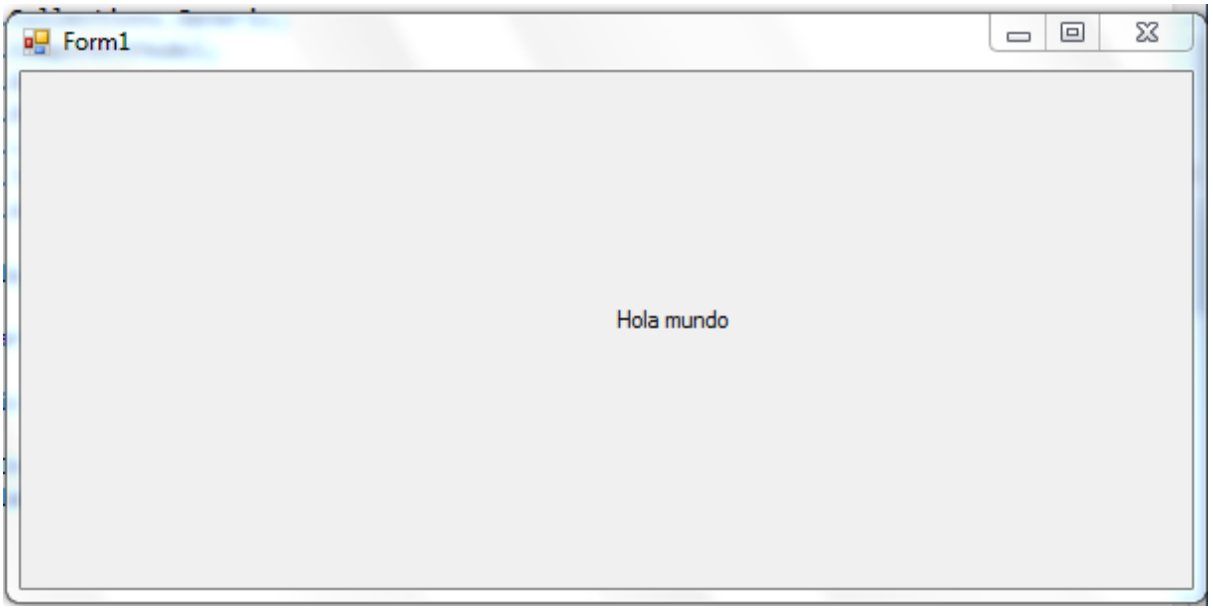
Nos aparece un formulario de diseño donde podemos agregar el objeto Label1 desde el cuadro de herramientas:



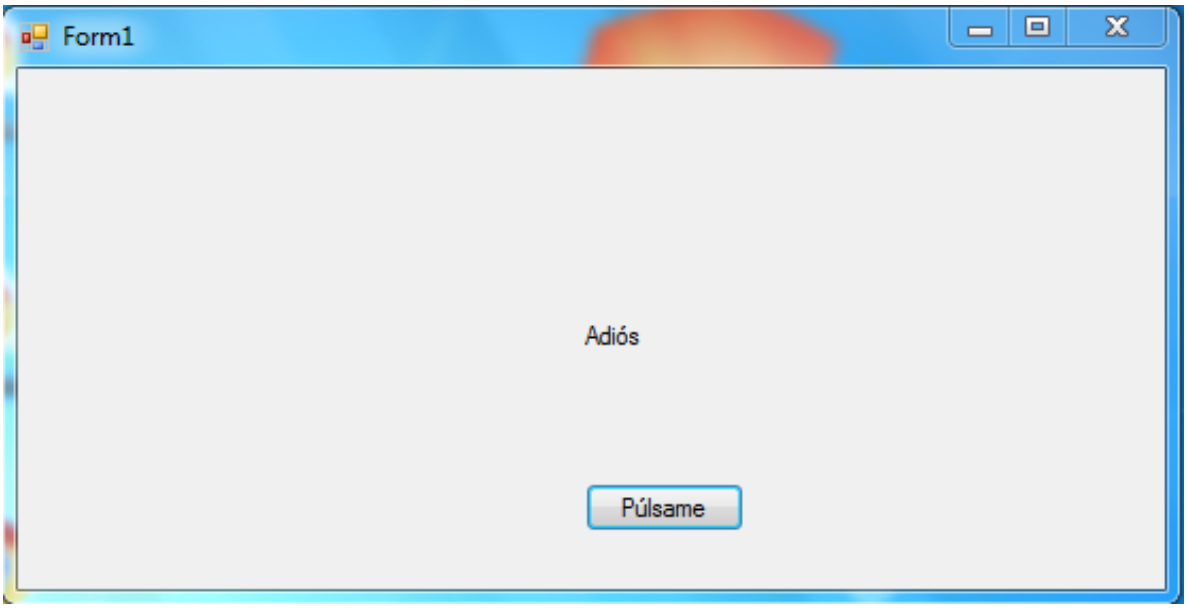
Pulsamos sobre el objeto label1 con el botón derecho del ratón y seleccionamos Ver código. Cuando nos aparezca el código del programa, introduciremos la instrucción: label1.text = “Hola mundo”;



Para iniciar la ejecución del programa pulsaremos Depurar -> Iniciar depuración.



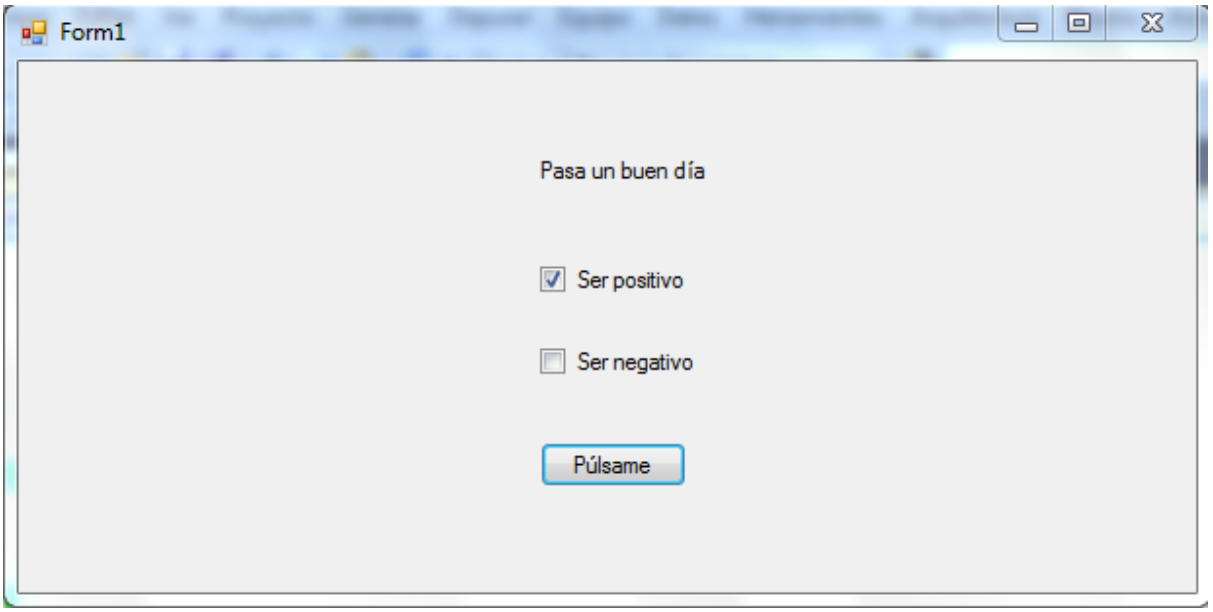
Ahora vamos a introducir un objeto nuevo que será un botón, y haremos que al pulsarlo, en la etiqueta label1, aparezca el mensaje “Adiós”.




```
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            label1.Text = "Hola mundo 2";
        }

        private void button1_Click(object sender, EventArgs e)
        {
            label1.Text = "Adiós";
        }
    }
}
```

3. Objeto CheckBox (Casillas de verificación)

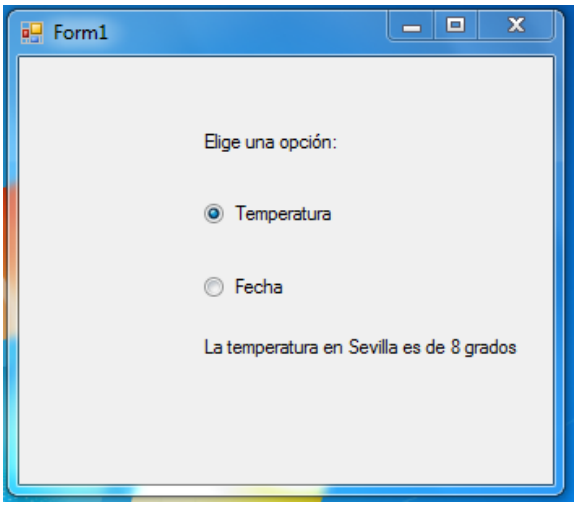
Las casillas de verificación o checkboxes son otro objeto de visual c# que podemos utilizar para nuestros programas. Un ejemplo de uso sería el siguiente:




```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        label1.Text = "Adiós";
        if (checkBox1.Checked == true && checkBox2.Checked==false) label1.Text = "Pasa un buen día";
        if (checkBox2.Checked == true && checkBox1.Checked==false) label1.Text = "Que tengas un mal día";
        if (checkBox1.Checked == true && checkBox2.Checked == true) label1.Text = "Sé coherente y elige una sola opción";
    }
}
```

4. Objeto radiobutton (opciones)

El objeto radiobutton permite elegir una opción única de las distintas opciones que hayamos creado. Un ejemplo de programa sería:



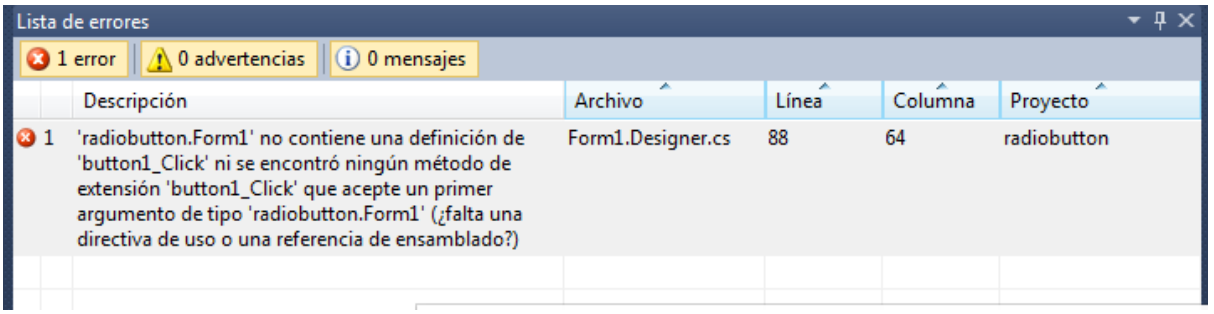

```
        public Form1()
        {
            InitializeComponent();
        }

        private void radioButton1_CheckedChanged(object sender, EventArgs e)
        {
            label2.Text = "La temperatura en Sevilla es de 8 grados";
        }

        private void radioButton2_CheckedChanged(object sender, EventArgs e)
        {
            label2.Text = "Hoy es Viernes 15 de enero de 2021";
        }
    }
}
```

5. Borrar el código de un evento que se ha escrito por error.

Cuando por error hacemos doble click sobre un objeto en el formulario de diseño nos aparecerá el código del evento:

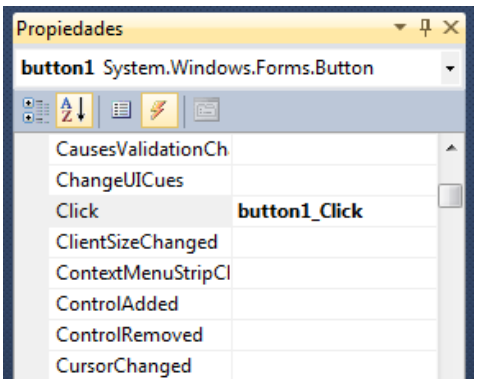
```
private void button1_Click(object sender, EventArgs e)
{

}
```

Si queremos borrar ese código, al hacerlo el programa nos da error. Para eliminar el error debemos hacer doble click sobre él y en el código que nos aparezca borrar la referencia a ese evento.



6. **Cómo desencadenar distintos eventos sobre un objeto.**

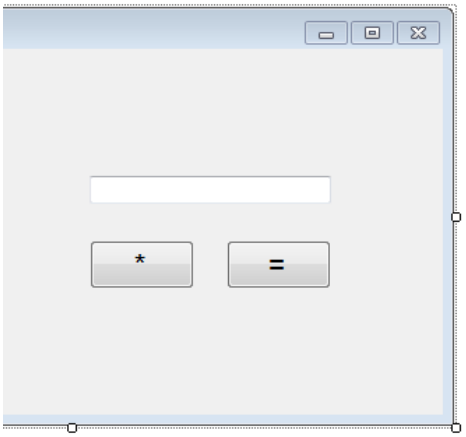
Por defecto, cuando hacemos doble click sobre un objeto en el formulario de diseño, nos aparece el evento objeto_Click (al hacer click sobre el objeto). Si quiero desencadenar otro evento, lo haría de la siguiente forma. Me sitúo sobre el objeto y al lado de las propiedades veo un rayo que son los distintos eventos disponibles. Ahi podré elegir el que me interesa desencadenar.



7. **Objeto Textbox**

Este objeto nos permite introducir información mediante el teclado, para poder guardarla en alguna variable (normalmente un string). Equivale a la instrucción cin que usábamos en C++.

Veamos un ejemplo de uso:



El código que tendría nuestra aplicación después de crear el diseño del formulario sería algo como esto:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace textboxnew
{
   public partial class Form1 : Form
   {
      public String numero1;
      public Int64 num1;
```

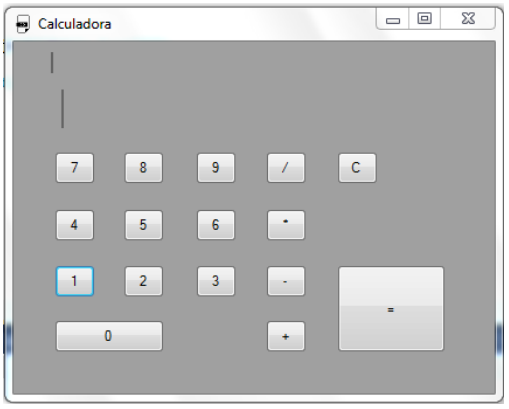
```
        public String numero2;
        public Int64 num2;
        public String resultado;
        public Int64 result;
        public Form1()
        {
            InitializeComponent();
        }

        private void botonpor_Click(object sender, EventArgs e)
        {
            numero1 = textBox1.Text;
            num1 = Convert.ToInt64(numero1);
            textBox1.Clear();
        }

        private void botonigual_Click(object sender, EventArgs e)
        {
            numero2 = textBox1.Text;
            num2 = Convert.ToInt64(numero2);
            result = num1 * num2;
            resultado = Convert.ToString(result);
            textBox1.Text = resultado;
        }

        private void botonpor_MouseUp(object sender, MouseEventArgs e)
        {
            textBox1.Focus();
        }
    }
}
```

Nota aclaratoria: Hemos tenido que convertir los string en int64 (enteros de 64 bits) para poder operar con ellos, pues recordemos que en un objeto textbox se almacenan datos de tipo string (cadena de caracteres). Esto lo hemos logrado con la instrucción Convert.

Curiosidad: Si queremos que después de darle al botón * nos aparezca el cursor en el textbox1 sin tener que darle con el ratón, solo tenemos que disparar el evento MouseUp sobre el objeto botonpor y añadir el código textBox1.Focus();

8. Diseño de una calculadora en Visual C#.

Con los objetos que ya conocemos podemos diseñar una calculadora básica. Empecemos con el diseño del formulario:



El código de nuestro programa sería el siguiente:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace calculator
{
    public partial class Form1 : Form
    {
        float num1,num2,result;
        string resultado;
        bool flagsuma = false;
        bool flagresta = false;
        bool flagmultiplica = false;
        bool flagdivide = false;
        public Form1()
        {
            InitializeComponent();
        }

        private void button0_Click(object sender, EventArgs e)
        {
            if (label1.Text == "") { }
            else label1.Text = label1.Text + "0";
            if (label1.Text == "") { }
            else label2.Text = label2.Text + "0";
        }
        private void button1_Click(object sender, EventArgs e)
        {
            label1.Text = label1.Text + "1";
            label2.Text = label2.Text + "1";
        }
```

```
private void button2_Click(object sender, EventArgs e)
{
   label1.Text = label1.Text + "2";
   label2.Text = label2.Text + "2";
}

private void button3_Click(object sender, EventArgs e)
{
   label1.Text = label1.Text + "3";
   label2.Text = label2.Text + "3";
}

private void button4_Click(object sender, EventArgs e)
{
   label1.Text = label1.Text + "4";
   label2.Text = label2.Text + "4";
}

private void button5_Click(object sender, EventArgs e)
{
   label1.Text = label1.Text + "5";
   label2.Text = label2.Text + "5";
}

private void button6_Click(object sender, EventArgs e)
{
   label1.Text = label1.Text + "6";
   label2.Text = label2.Text + "6";
}

private void button7_Click(object sender, EventArgs e)
{
   label1.Text = label1.Text + "7";
   label2.Text = label2.Text + "7";
}

private void button8_Click(object sender, EventArgs e)
{
   label1.Text = label1.Text + "8";
   label2.Text = label2.Text + "8";
}

private void button9_Click(object sender, EventArgs e)
{
   label1.Text = label1.Text + "9";
   label2.Text = label2.Text + "9";
}

private void buttonmas_Click(object sender, EventArgs e)
{
   num1 = Convert.ToSingle(label1.Text);
   label1.Text = "";
```

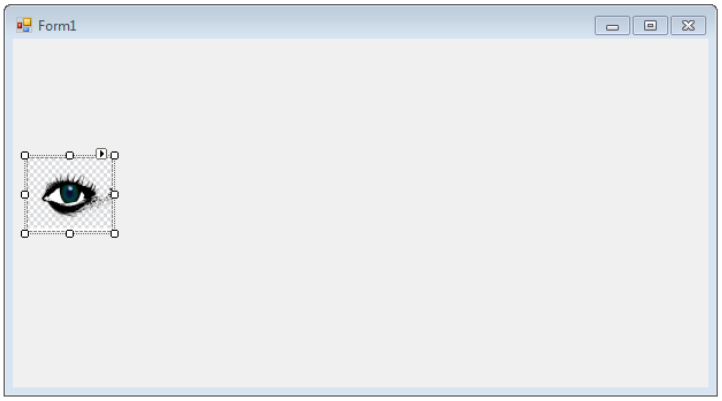
```
    label2.Text = label2.Text + " + ";
    flagsuma = true;
}

private void buttonmenos_Click(object sender, EventArgs e)
{
    num1 = Convert.ToSingle(label1.Text);
    label1.Text = "";
    label2.Text = label2.Text + " - ";
    flagresta = true;
}

private void buttonmultiplicar_Click(object sender, EventArgs e)
{
    num1 = Convert.ToSingle(label1.Text);
    label1.Text = "";
    label2.Text = label2.Text + " x ";
    flagmultiplica = true;
}

private void buttondividir_Click(object sender, EventArgs e)
{
    num1 = Convert.ToSingle(label1.Text);
    label1.Text = "";
    label2.Text = label2.Text + " / ";
    flagdivide = true;
}

private void buttonigual_Click(object sender, EventArgs e)
{
    num2 = Convert.ToSingle(label1.Text);
    if (flagsuma == true)
    {
        result = num1 + num2;
        flagsuma = false;
    }
    if (flagresta == true)
    {
        result = num1 - num2;
        flagresta = false;
    }
    if (flagmultiplica == true)
    {
        result = num1 * num2;
        flagmultiplica = false;
    }
    if (flagdivide == true)
    {
        result = num1 / num2;
        flagdivide = false;
    }
    resultado = Convert.ToString(result);
```

```
            label1.Text = resultado;

        }

        private void buttonc_Click(object sender, EventArgs e)
        {
            label1.Text = "";
            label2.Text = "";
            flagsuma = false;
            flagresta = false;
            flagmultiplica = false;
            flagdivide = false;
        }

    }
}
```

9. Objeto Picturebox

Este objeto nos permitirá poner imágenes en nuestro formulario. Como ejemplo, hemos realizado una aplicación que mueve una imagen de izquierda a derecha del formulario a una velocidad determinada.

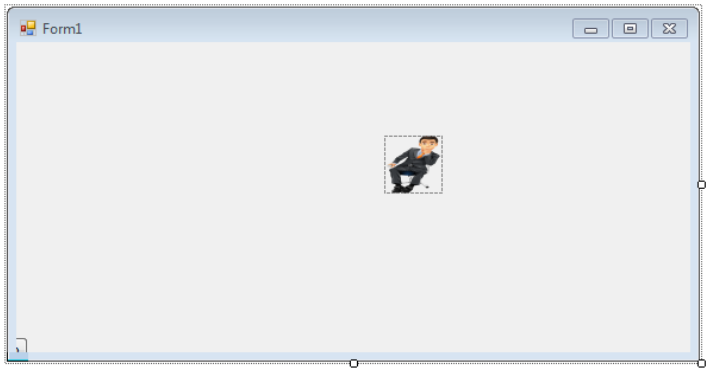
Si la imagen insertada en el picturebox no se ajusta a su tamaño, podemos poner el Modo de tamaño a StrechImage para lograrlo.




```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Threading; // Hemos introducido esta instrucción para usar Sleep()

namespace imagenes
{
    public partial class Form1 : Form
    {
        public Form1()
```

```
        {
            InitializeComponent();
        }

        private void pictureBox1_Click(object sender, EventArgs e)
        {
            for (int x = 0; x < 500; x++)
            {
                pictureBox1.Location = new Point(x, 100);
                Thread.Sleep(10);
            }
        }
    }
}
```

10. Mover un objeto utilizando el teclado

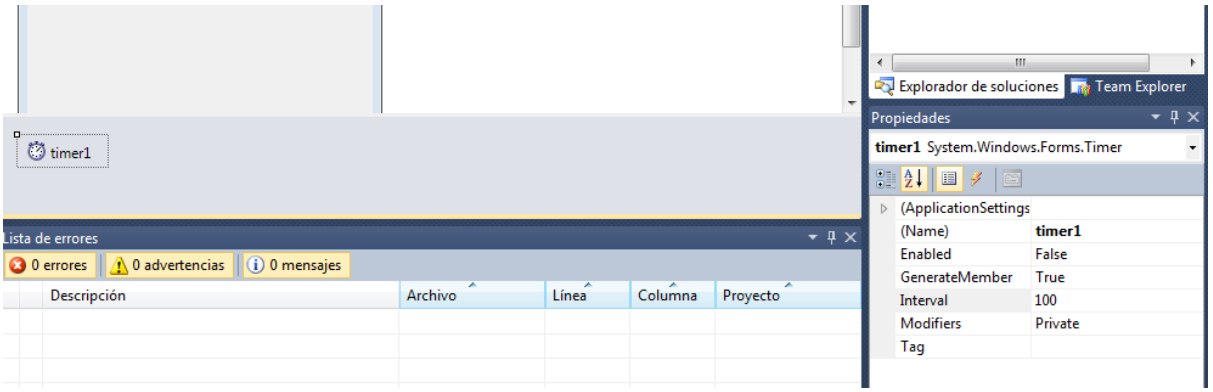



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace movimientoconteclado
{
    public partial class Form1 : Form
    {
        public int x = 300;
        public int y = 75;
        public Form1()
        {
            InitializeComponent();
            pictureBox1.Location = new Point(x, y);
        }
        private void button1_KeyDown(object sender, KeyEventArgs e)
        {
            if (e.KeyData == Keys.D)
            {
                x=x+5;
```

```
            pictureBox1.Location = new Point(x, y);
        }
        if (e.KeyData == Keys.A)
        {
            x = x - 5;
            pictureBox1.Location = new Point(x, y);
        }
        if (e.KeyData == Keys.W)
        {
            y = y - 5;
            pictureBox1.Location = new Point(x, y);
        }
        if (e.KeyData == Keys.S)
        {
            y = y + 5;
            pictureBox1.Location = new Point(x, y);
        }

    }
  }
}
```

11. Objeto Timer

Cuando necesitamos que ocurra un evento al cabo de un tiempo, podemos usar el objeto Timer, en el cual podemos modificar la propiedad Interval a los milisegundos que pasan para que ocurra el evento repetidamente.

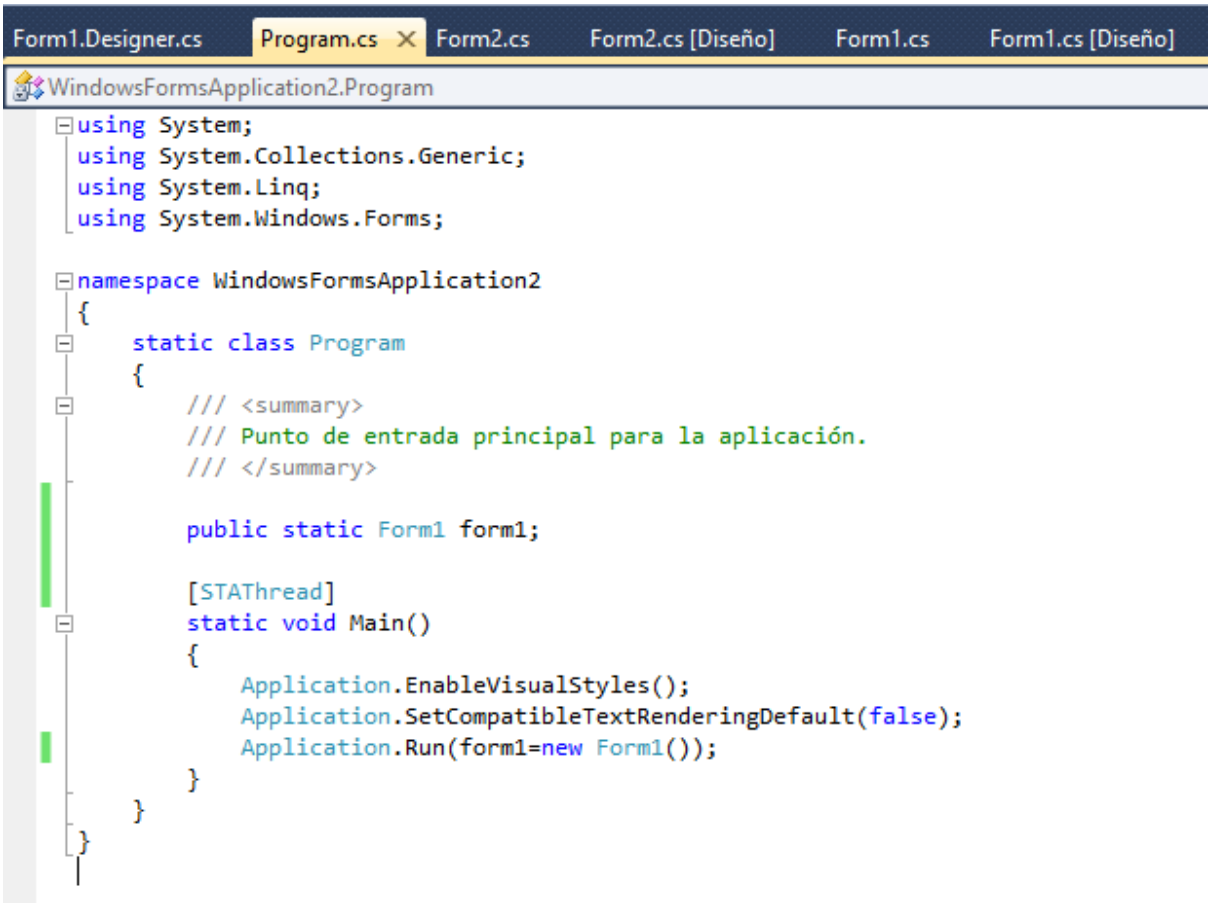


También tendremos que modificar la propiedad Enabled a true para que el timer funcione.

12. Agregar nuevos formularios al programa

Habrá ocasiones en que necesitemos usar varios formularios dentro de un programa.

Si queremos que nada más empezar la ejecución del programa, Form1 desaparezca y aparezca Form2, haremos la siguiente modificación en Program.cs



Luego en Form1.cs(Diseño) creamos un objeto timer y añadimos este código en Form1.cs:

Form1.Designer.cs | Program.cs | Form2.cs* | Form2.cs [Diseño]* | Form1.cs*

WindowsFormsApplication2.Form1

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication2
{
    public partial class Form1 : Form
    {
        Form2 frm2 = new Form2();
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            timer1.Enabled = true;
        }

        private void timer1_Tick(object sender, EventArgs e)
        {

            frm2.Show();
            timer1.Enabled = false;
        }
    }
}
```

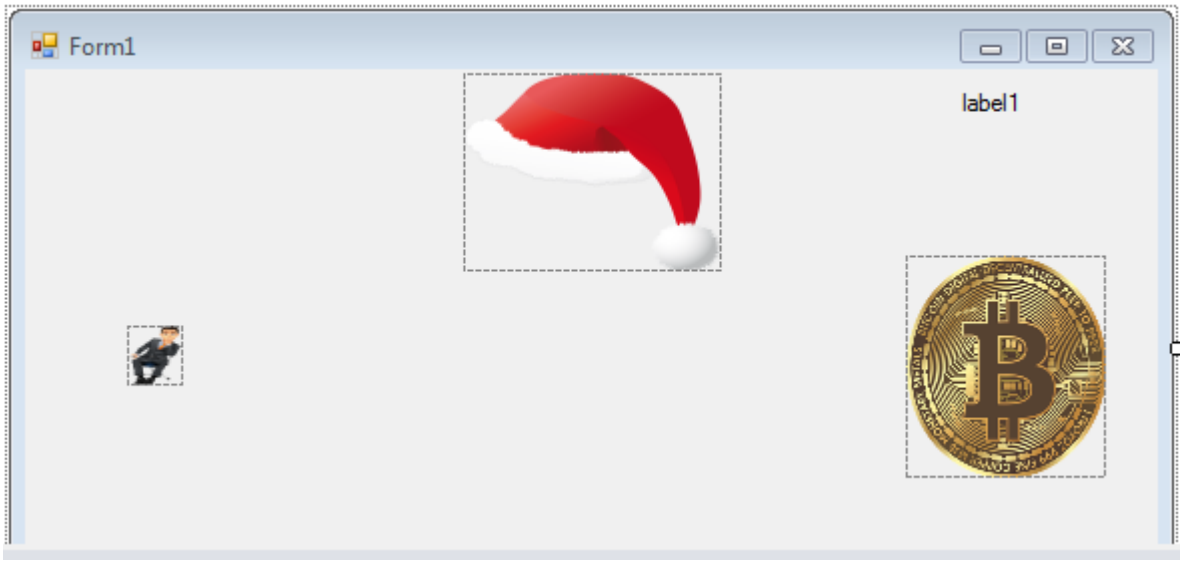
Cuando queramos volver al formulario 1, dentro del formulario 2, podemos poner un botón de comenzar en el formulario 2 y añadir este código:

Form1.Designer.cs | Program.cs | Form2.cs* | Form2.cs [Diseño]* | Form1.cs*

WindowsFormsApplication2.Form2

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication2
{
    public partial class Form2 : Form
    {

        public Form2()
        {
            InitializeComponent();
        }

        private void Form2_Load(object sender, EventArgs e)
        {
            Program.form1.Hide();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
            Program.form1.Show();
        }

    }
}
```

13. Colisión entre objetos (picturebox)




```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace movimientoconteclado
{
    public partial class Form1 : Form
    {
        public int x1 = 20;
        public int y1 = 125;
        public int x2 = 250;
        public int y2 = 1;
        public bool flagsubiendo = false;
        public int vidas = 3;

        public Form1()
        {
            InitializeComponent();
            pictureBox1.Location = new Point(x1, y1);
            pictureBox2.Location = new Point(x2, y2);
            label1.Text = "Vidas: " + vidas;
        }
        private void button1_KeyDown(object sender, KeyEventArgs e)
        {
            if (e.KeyData == Keys.D)
            {
                x1=x1+5;
                pictureBox1.Location = new Point(x1, y1);
            }
            if (e.KeyData == Keys.A)
```

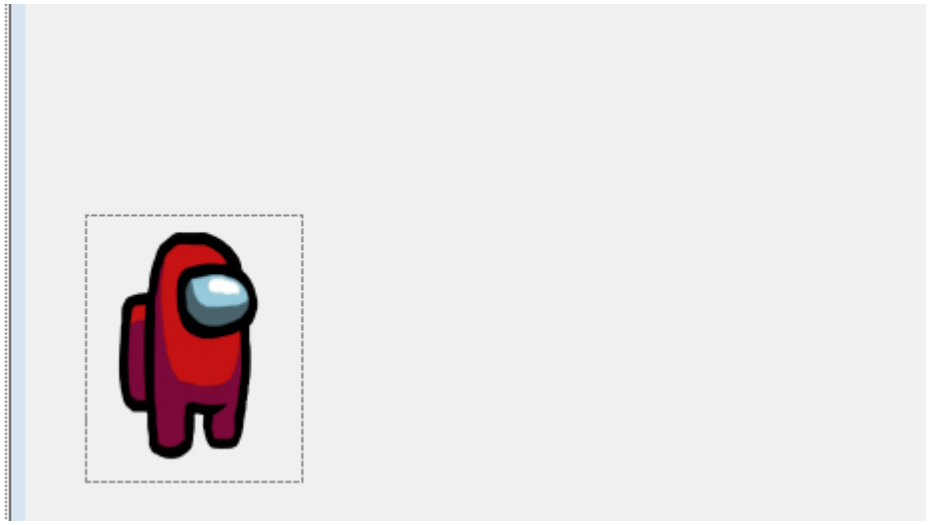
```
    {
        x1 = x1 - 5;
        pictureBox1.Location = new Point(x1, y1);
    }
    if (e.KeyData == Keys.W)
    {
        y1 = y1 - 5;
        pictureBox1.Location = new Point(x1, y1);
    }
    if (e.KeyData == Keys.S)
    {
        y1 = y1 + 5;
        pictureBox1.Location = new Point(x1, y1);
    }

}

private void timer1_Tick(object sender, EventArgs e)
{
    if (flagsubiendo == false)
    {
        y2++;
    }
    if (flagsubiendo == true)
    {
        y2--;
    }
    if (y2 == 300) flagsubiendo = true;
    if (y2 == 1) flagsubiendo = false;
    pictureBox2.Location = new Point(x2, y2);
}

private void timer2_Tick(object sender, EventArgs e)
{
    if (pictureBox2.Bounds.Contains(pictureBox1.Location))
    {
        x1 = 20;
        y1 = 125;
        pictureBox1.Location = new Point(x1, y1);
        vidas--;
        label1.Text = "Vidas: " + vidas;
        if (vidas == -1)
        {
            label1.Text = "GAME OVER";
            timer1.Enabled = false;
            timer2.Enabled = false;
            button1.Visible = false;
            MessageBox.Show("GAME OVER");
        }
    }
    if (pictureBox3.Bounds.Contains(pictureBox1.Location))
    {
```

```
            x1 = 20;
            y1 = 125;
            pictureBox1.Location = new Point(x1, y1);
            Form2 frm = new Form2();
            frm.Show();
        }

    }

  }
}
```

14. Lograr que un muñeco de un salto con efecto de gravedad.

Para lograrlo tendremos que recurrir a la ecuación del tiro parabólico e introducirla en nuestro programa. Esta ecuación tiene esta forma:
yinicial-y=(x - xinicial) - 0.01 * (x - xinicial)*(x-xinicial)

Nuestro formulario de diseño tendría este aspecto:



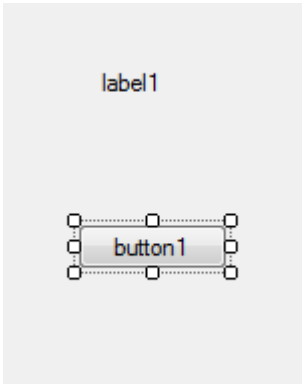
El programa que deberíamos usar para simular el salto con gravedad podría tener el siguiente código:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication2
{
   public partial class Form1 : Form
```

```
  {
    public int x = 10;
    public double xinicial;
    public int y = 300;
    public double ydouble;
    public double xdouble;
    public string xstring;
    public string ystring;
    public Form1()
    {
      InitializeComponent();
      pictureBox1.Location = new Point(x, y);
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
      pictureBox1.Location = new Point(x, y);
      x++;
      xdouble = Convert.ToDouble(x);
      ydouble = 300 - ((xdouble - xinicial) - 0.01 * (xdouble - xinicial)*(xdouble-xinicial));
      xstring = Convert.ToString(xdouble);
      label1.Text = xstring;
      ystring = Convert.ToString(ydouble);
      label2.Text = ystring;
      x = Convert.ToInt16(xdouble);
      y = Convert.ToInt16(ydouble);
      if (y > 300)
      {
        y = 300;
        timer1.Enabled = false;
      }
    }

    private void pictureBox1_Click(object sender, EventArgs e)
    {
      timer1.Enabled = true;
      xinicial = Convert.ToDouble(x);
    }


  }
}
```

15. Generar números aleatorios

Haremos un programa que al darle a un botón, nos genere un número aleatorio del 1 al 100 y lo muestre en una label.




```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace numerosaleatorios
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }

      private void button1_Click(object sender, EventArgs e)
      {
         Random rand1 = new Random();
         int numero = rand1.Next(1, 100);
         label1.Text = Convert.ToString(numero);
      }
   }
}
```

16. Mover un objeto picturebox con el ratón

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication2
{
   public partial class Form1 : Form
   {
```

```
        private bool btnDown;
        private int offsetX;
        private int offsetY;

        public Form1()
        {
            InitializeComponent();
        }

        private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
        {
            // el boton izquierdo esta pulsado
            if (e.Button == MouseButtons.Left)
            {
                btnDown = true;
                offsetX = e.X;
                offsetY = e.Y;
            }
        }

        private void pictureBox1_MouseMove(object sender, MouseEventArgs e)
        {
            if (btnDown)
            {
                // mover el pictureBox con el raton
                pictureBox1.Left += e.X - offsetX;
                pictureBox1.Top += e.Y - offsetY;
            }
        }
        private void pictureBox1_MouseUp(object sender, MouseEventArgs e)
        {
            // el boton izquierdo se libera
            if (e.Button == MouseButtons.Left)
            {
                btnDown = false;
            }
        }
    }
}
```

17. Superponer imágenes con objetos picturebox.

Por defecto, el último picturebox usado se superpone sobre los anteriores. Si queremos que esto cambie durante la ejecución del programa, podremos usar las funciones, pictureBox1.BringToFront(); y pictureBox2.SendToBack();

Un ejemplo de código sería:

```
private void Form1_Load(object sender, EventArgs e)
    {
      pictureBox1.BringToFront();
```

```
    pictureBox2.SendToBack();
}
```

18. Reiniciar o salir de una aplicación.

Para reiniciar una aplicación podemos usar la función Application.Restart();
Para salir de la aplicación podemos usar la función Application.Exit();