

Tema: “Entorno de Desarrollo Visual Studio en C#”.

Objetivos

- Construir aplicaciones básicas de Windows Forms para proyectos en C#.

Introducción

INTRODUCCIÓN A WINDOWS FORM

Esta guía tiene como objeto mostrar cómo construir aplicaciones básicas de Windows Forms usando varios de los componentes más comunes que son una característica de la mayoría de las aplicaciones GUI.

Se verá como establecer las propiedades de los formularios y de los componentes de Windows Forms usando Visual Designer y la ventana de propiedades. También tiene por objeto que se aprenda a cambiar o examinar los valores de estas propiedades dinámicamente usando el código de C#.

HERRAMIENTAS DE C#

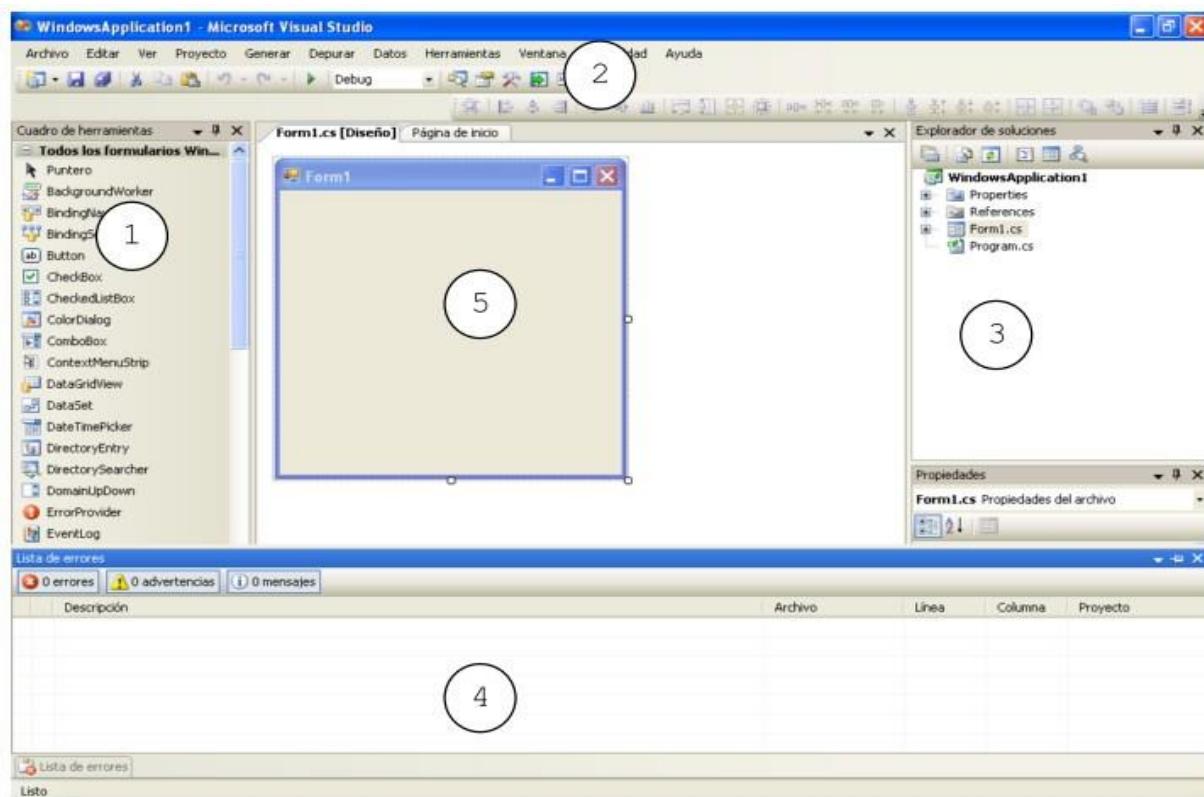
A continuación, se detallan las herramientas y ventanas más importantes de Visual C#. Las ventanas de la mayoría de herramientas pueden abrir desde el menú “Ver”.

1. El editor de código, para escribir código fuente
2. El compilador de C#, que se encarga de convertir el código fuente de C# en un programa ejecutable.
3. El depurador de Visual Studio, para probar el programa
4. El cuadro de herramientas y el Diseñador, para desarrollar rápidamente interfaces de usuario con el mouse.
5. El explorador de soluciones, para ver y administrar archivos de proyectos y configuraciones.
6. El diseñador de proyectos, para configurar opciones del compilador, rutas de implementación, recursos, entre otros.
7. La vista de clases, para desplazarse por el código fuente según los tipos, no los archivos.

8. La ventana “Propiedades”, para configurar propiedades y eventos en los controles de la interfaz de usuario.
9. El examinador de objetos, que ve los métodos y clases disponibles en las bibliotecas de vínculos dinámicos, incluidos los ensambladores de .NET Framework y los objetos COM
10. Document Explorer, para explorar y buscar la documentación del producto en su equipo local y en Internet.

ENTORNO

El entorno C# está compuesto con herramientas para interactuar a través de ventanas, páginas de propiedades y asistentes.



1. Cuadro de herramientas
2. Barra de herramientas y menú
3. Proyectos abiertos, propiedades y ayuda
4. Compilación de la aplicación, listado de errores.
5. Área de diseño y edición

EXPLORADOR DE SOLUCIONES Y DISEÑADOR DE PROYECTOS

La ventana de la parte superior derecha es el “Explorador de Soluciones”, que muestra todos los archivos del proyecto en una vista de árbol jerárquica.

Cuando se utiliza el menú “Proyecto” para agregar nuevos archivos al proyecto, se verán reflejados en el Explorador de Soluciones. Además de los archivos, el Explorador de soluciones también muestra la configuración del proyecto y las referencias a las bibliotecas externas que necesita la aplicación.

Para obtener acceso a las páginas de propiedades del Diseñador de proyectos, haga clic con el botón secundario del mouse en “Propiedades” del Explorador de soluciones y, a continuación, haga clic en “Abrir”.

Utilice estas páginas para modificar opciones de generación, requisitos de seguridad, detalles de implementación y muchas otras propiedades del proyecto.

Material y Equipo

- Guía de laboratorio No. 5.
- Computadora con Visual Studio 2013 o superior.
- Dispositivo de almacenamiento (USB).

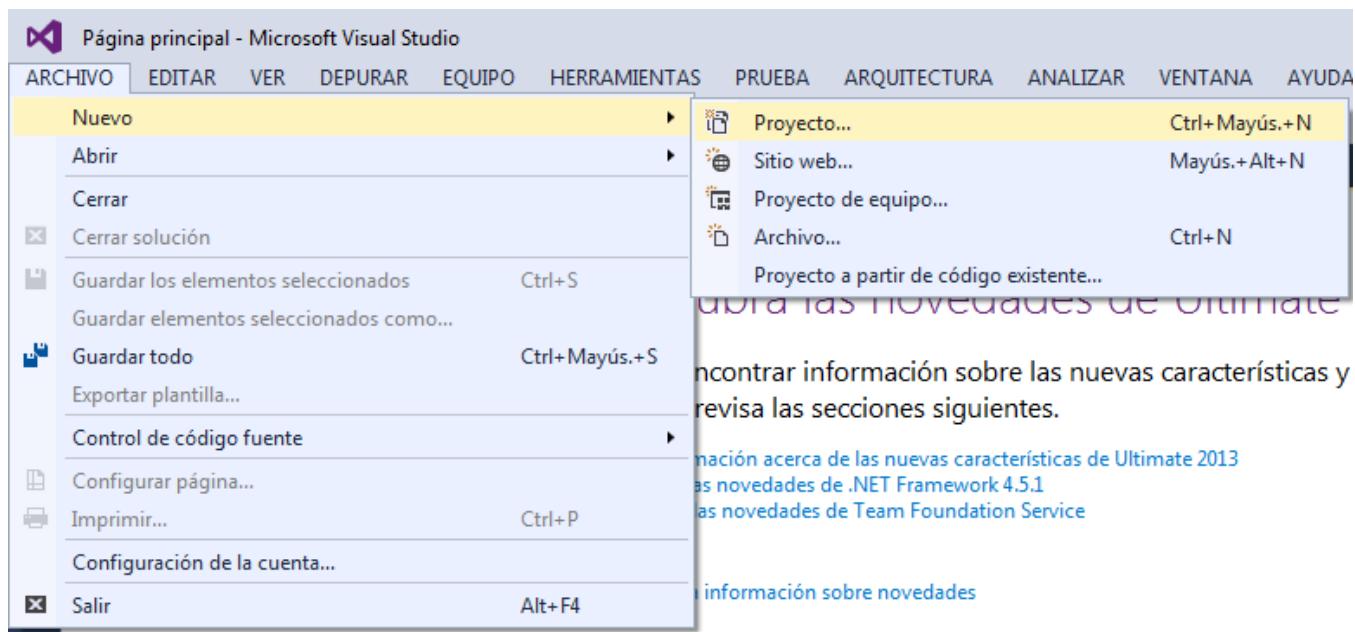
Procedimiento

Ejemplo1

Creación de primer proyecto.

Para la creación del proyecto, en Visual Studio

1. Inicie Visual Studio.NET
2. En el menú “Archivo (FILE)”, seleccione “Nuevo (NEW)” y después seleccione la opción “Proyecto (PROJECT)”



encontrar información sobre las nuevas características y revisa las secciones siguientes.

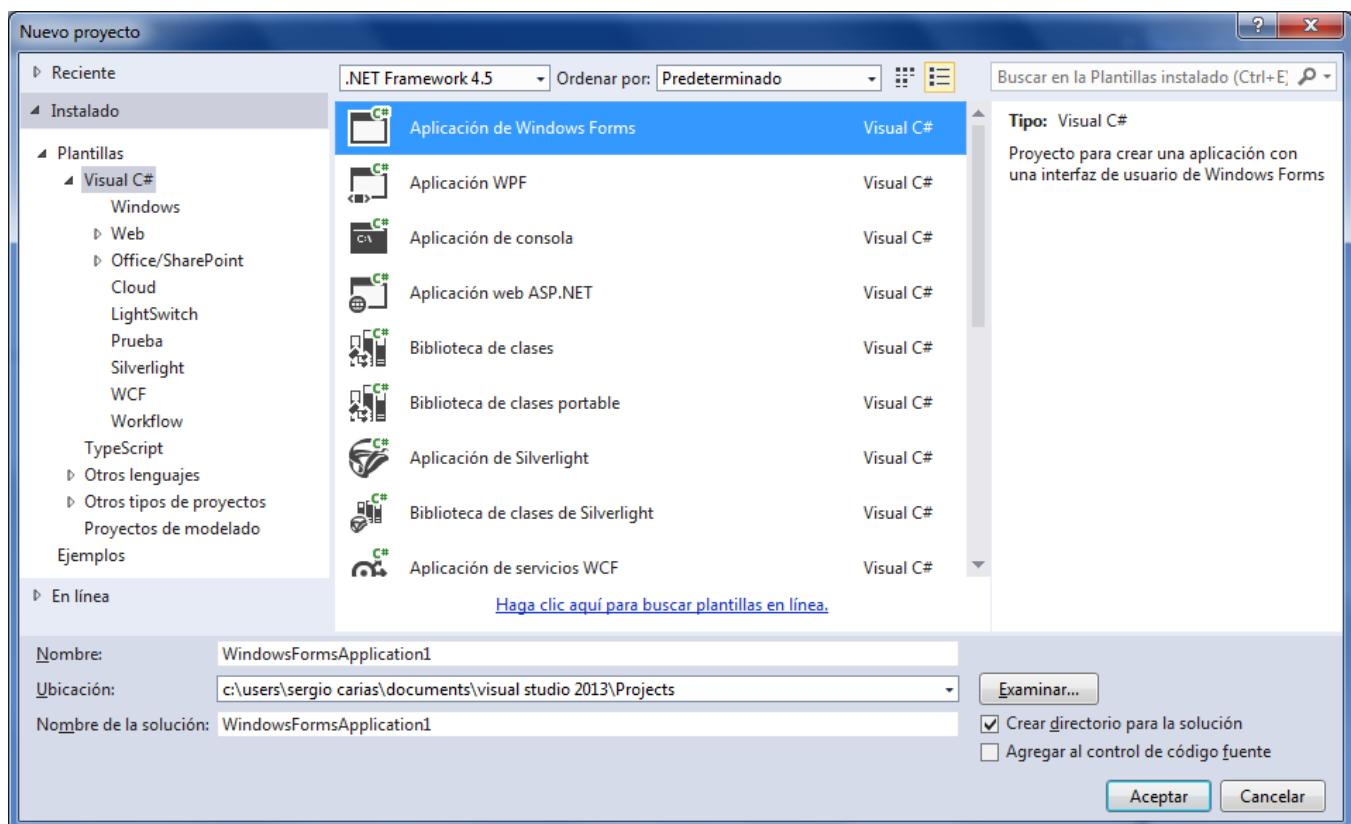
nación acerca de las nuevas características de Ultimate 2013

as novedades de .NET Framework 4.5.1

as novedades de Team Foundation Service

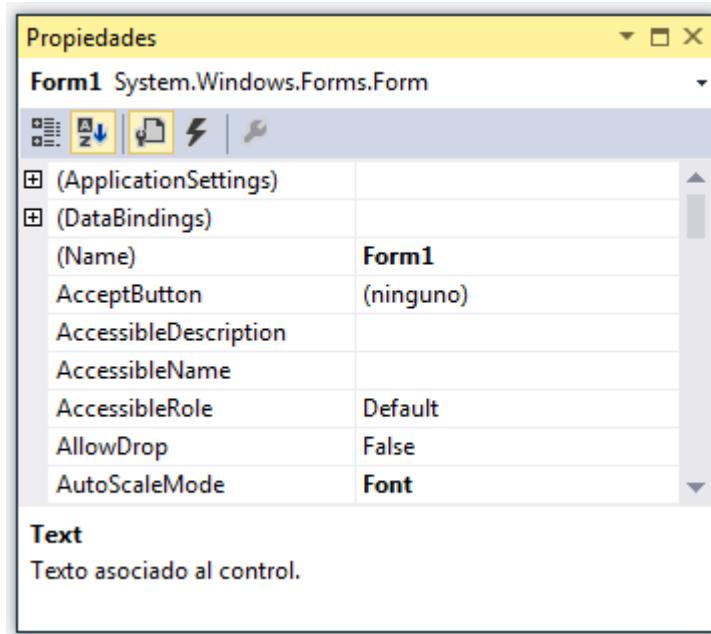
información sobre novedades

En el panel “Tipos de Proyecto”, seleccione proyectos de Visual C# y en el panel de plantillas, seleccione “Aplicación para Windows (Windows Forms Application)” y en el cuadro de texto “Nombre”, escribiremos “Ejemplo1”, después hacer clic en el botón “Aceptar (OK)”

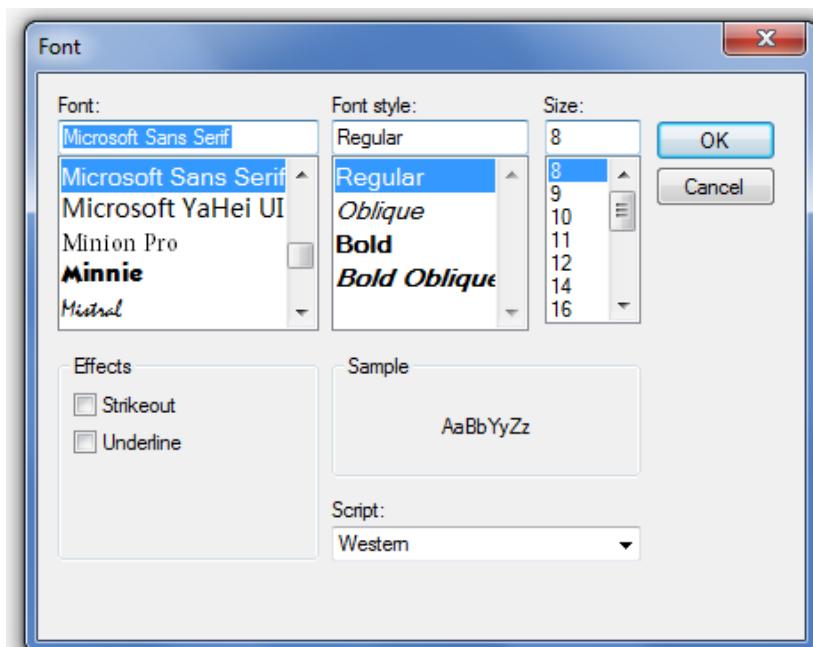


Pasos

1. En la ventana “Propiedades” (si no ve la ventana presione (F4), haga clic en la propiedad (Name), y después escriba Ejemplo1 en el cuadro de texto (Name) para cambiar el nombre del control del formulario (esto se hace así, porque con este nombre se hará referencia cuando se esté programando)



2. En la misma ventana “Propiedades”, seleccione la propiedad Text y después escriba “Este es un ejemplo”, para cambiar la barra de título del formulario.
3. Seleccione la propiedad Font y haga clic en el botón “Puntos suspensivos” que aparece al seleccionar la propiedad. Cuando se hace clic en el botón “Puntos suspensivos”, se abre el cuadro de diálogo “Font” y se puede seleccionar la fuente y los efectos que se desean.

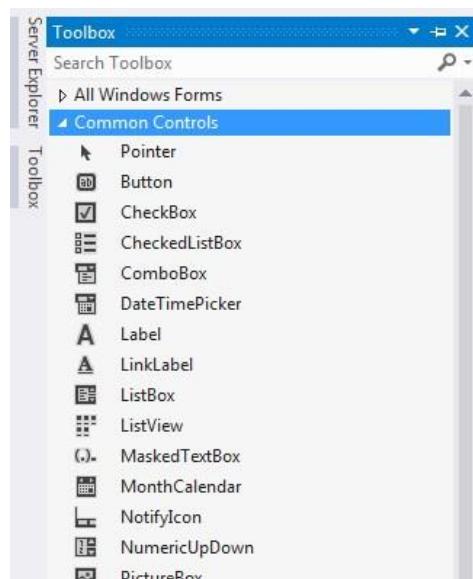


Añadir controles a los formularios

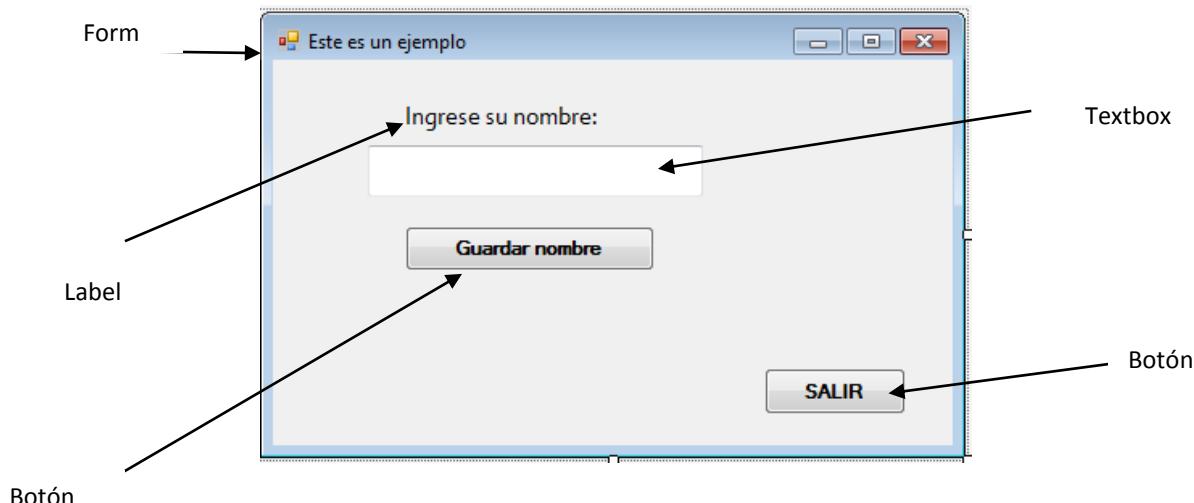
Hasta ahora se ha creado un formulario, se han establecido algunas de sus propiedades. Para que el formulario sea útil, se necesita agregar los controles y escribir algo de código propio.

Agregar controles de Windows Forms

En la siguiente figura se muestra el cuadro de herramientas, donde podemos arrastrar los controles que necesitamos para nuestra aplicación, (si no aparece la venta presionar Ctrl + w +x)



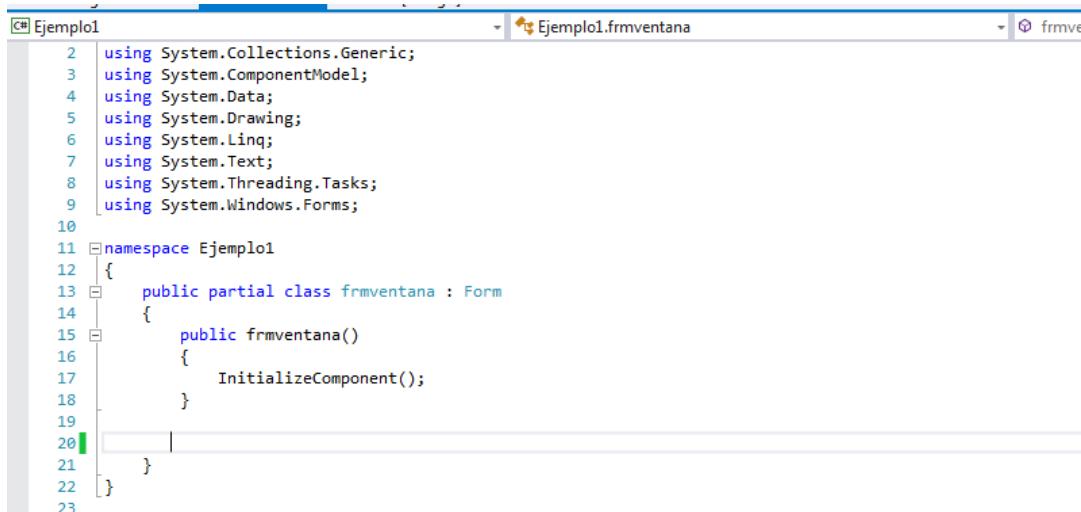
Arrastre los controles necesarios, para construir un formulario similar al mostrado en la figura siguiente:



Control	Propiedad	Valor
form1	Text	Este es un ejemplo
	Name	frmventana
label1	Text	Ingrese su nombre:
	Name	lbmensaje
textbox1	Text	(dejar vacío)
	Name	txtnombre

button1	Text	Guardar nombre
	Name	btnguardar
button2	Text	SALIR
	Name	btnsalir

Ahora que tenemos toda la interfaz diseñada procedamos a la codificación:



```

2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11 namespace Ejemplo1
12 {
13     public partial class frmventana : Form
14     {
15         public frmventana()
16         {
17             InitializeComponent();
18         }
19
20     }
21 }
22

```

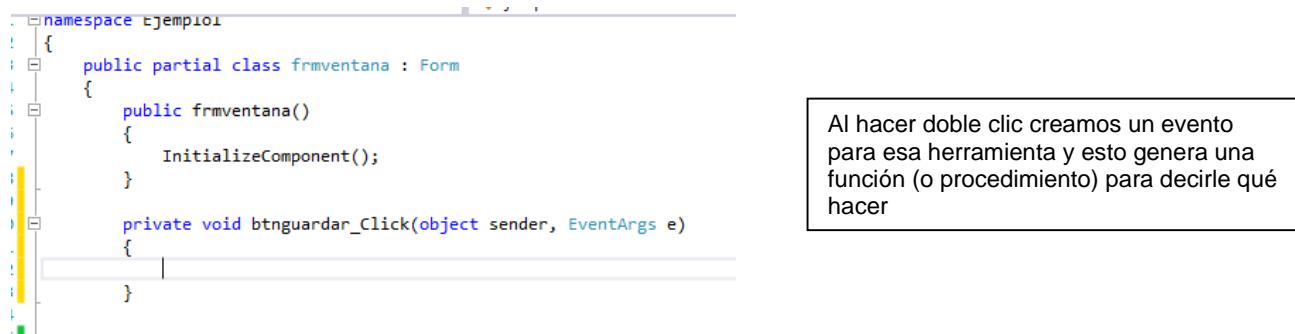
Entremos al código dando **F7** o clic derecho sobre el form y eligiendo < >View Code.

Para regresar al diseño cambiamos de pestaña y seleccionamos Form1.cs [Design]

Programando eventos

Hasta ahora se ha creado un formulario, se han establecido algunas de sus propiedades. Como siguiente paso codificaremos los eventos de cada herramienta, de forma que logremos obtener nuestro mensaje.

1. Damos doble clic sobre el btnguardar y nos mostrará algo como esto:



```

namespace Ejemplo1
{
    public partial class frmventana : Form
    {
        public frmventana()
        {
            InitializeComponent();
        }

        private void btnguardar_Click(object sender, EventArgs e)
        {
        }
    }
}

```

Al hacer doble clic creamos un evento para esa herramienta y esto genera una función (o procedimiento) para decirle qué hacer

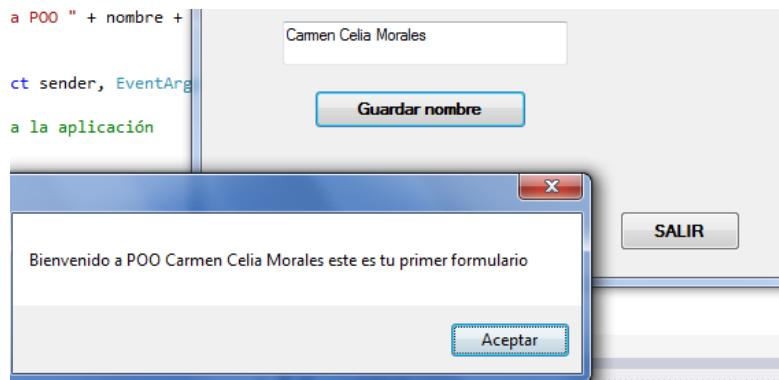
1.1 Ingresamos el siguiente fragmento de código en este método

```
string nombre = txtNombre.Text;  
MessageBox.Show("Bienvenido a POO " + nombre + " este es tu primer formulario");
```

2. Como segundo paso hagamos la misma rutina pero con el btnsalir y dentro de su método escribamos:

```
Application.Exit(); //termina la aplicación
```

3. Estamos listos para probar el programa, ejecútelo con Start o F5 e ingrese su nombre en el textbox.
Al hacer esto debe aparecernos esta ventana



4. Para concluir, de clic en Aceptar y luego clic en el botón salir que usted programó.

CONSIDERACIONES

- Los label y los messagebox, suelen ser utilizados como los encargados de mostrar mensajes para el usuario (hacen la función de un Console.WriteLine()) pero en el entorno gráfico. Aunque no son los únicos sí son de los más frecuentes
- Hay varias formas de capturar información: textbox, radiobutton, listbox, comboBox. (Muchas formas de un Console.ReadLine())
- Todos los datos recogidos son considerados texto (string), así que si deseamos hacer cálculos con ellos (como una suma) hay que convertir eso a int, float, double o a lo que necesitemos.