

## FORMULARIOS DE WINDOWS

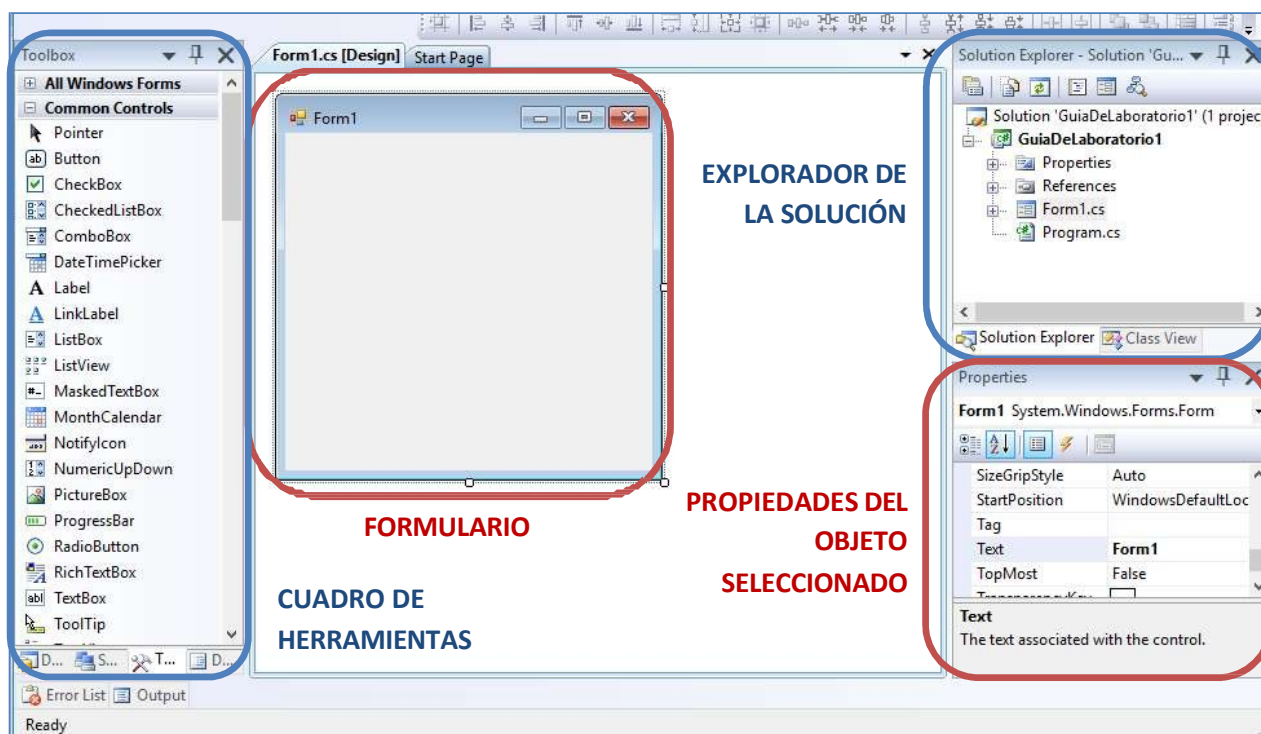
### Guía de laboratorio – Capítulo II

#### Contenidos:

- ✓ Elementos del entorno de aplicación de Windows Form C#.
- ✓ Los formularios y sus propiedades.
- ✓ Introducción al uso de eventos y métodos comunes.

#### IDENTIFICACION DEL ENTORNO WINDOWS FORMS, EL FORMULARIO Y SUS PROPIEDADES

Recordemos lo visto en la Guía N°1, donde conocimos el entorno de desarrollo de Visual C# y conozcamos más acerca de los formularios de Windows. Para ello, deberán crear un nuevo proyecto y seguir los pasos de cada actividad:



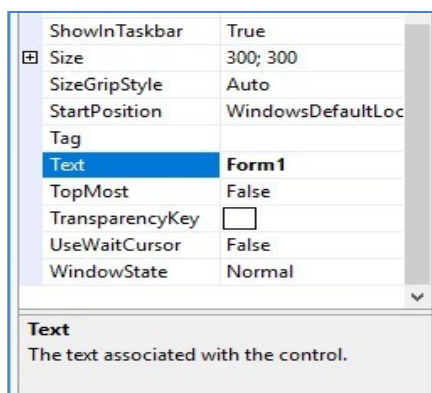
Un **Form** (formulario) es una representación que constituye la interfaz de usuario de una aplicación. La clase Form puede usarse para crear ventanas flotantes, estándar y sin bordes. También se puede usar la clase Form para crear ventanas modales, como un cuadro de diálogo.

En el panel derecho izquierdo encontraremos las herramientas que luego incluiremos dentro de nuestro formulario y de las que seguiremos hablando más adelante.

En el sector derecho se encuentra el explorador de la solución (una solución puede contener más de un proyecto, un proyecto más de un formulario y clases), donde podremos ver todos los elementos que la conforman, como proyectos y formularios como el que comenzaremos a modificar: "Form1".

También en el sector derecho veremos las **propiedades** de dicho formulario y en esto nos vamos a detener.

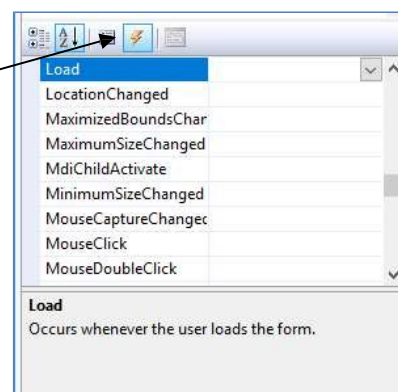
Al hacer clic sobre alguna de ellas, aparecerá una leyenda debajo de la lista con una breve descripción de esa propiedad. En el caso de la imagen, se encuentra seleccionada la propiedad “Text” del formulario (“Form1”) y la leyenda explica que refiere al texto asociado con el mismo.



En el mismo cuadro de Propiedades, veremos un botón identificado con un rayo, el cual nos permitirá acceder a los “Eventos” del Formulario.

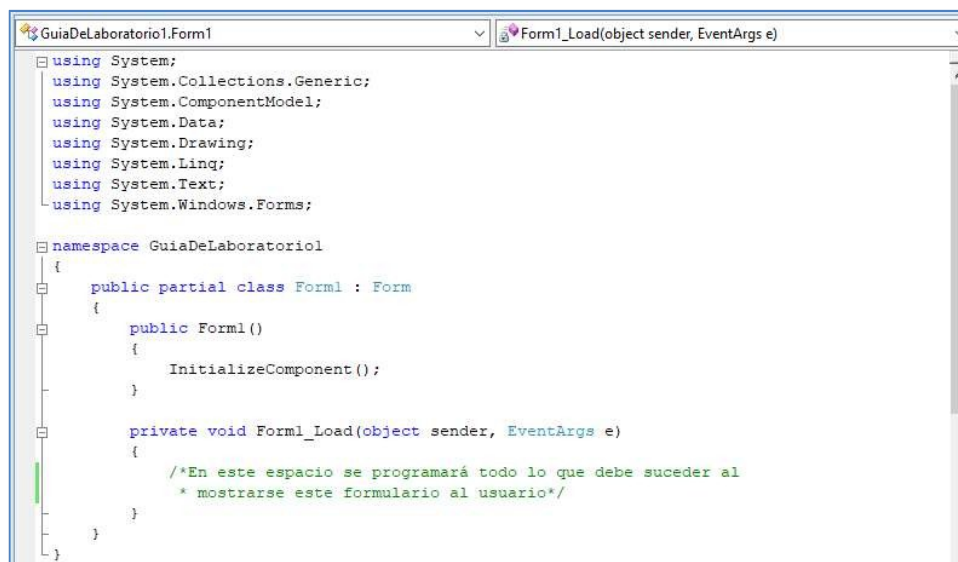
Un evento es una señal que comunica a una aplicación que ha sucedido algo importante. Por ejemplo, cuando un usuario hace clic en un control de formulario, este último puede provocar un evento Clic y llamar a un procedimiento que controla esta acción.

También al seleccionar alguno de estos eventos, aparecerá una breve reseña del mismo.



un

El evento por defecto del formulario (al hacer doble clic sobre él) y que se genera al iniciar la ejecución del programa, es “Load”.



De no aparecer alguno de estos cuadros en la ventana, se pueden habilitar en la opción “Ver” de la barra de menú.

Pueden modificar el tamaño del formulario arrastrando los puntos blancos en los lados del mismo y en su esquina inferior derecha.

### PROPIEDADES DE LOS FORMULARIOS USADAS CON MÁS FRECUENCIA

PROPIEDAD	DESCRIPCIÓN	EJEMPLO
<b>Name</b>	<i>Establece el nombre con el que identificaremos el objeto en el programa (no pueden existir dos nombres iguales en el mismo programa)</i>	FrmEjemplo
<b>Text</b>	<i>Es el título visible al usuario al momento de visualizar la ventana</i>	Pantalla de inicio
<b>StartPosition</b>	<i>Define la posición del formulario al ejecutar</i>	CenterScreen (recomendado)
<b>FormBorderStyle</b>	<i>Habilita o deshabilita la barra de título, el cuadro de control y la posibilidad de cambiar el tamaño del formulario en ejecución</i>	Fixed3D (recomendado)
<b>BackColor</b>	<i>Color de fondo del formulario</i>	192; 192; 255
<b>BackgroundImage</b>	<i>Permite seleccionar una imagen para el fondo del formulario</i>	(en archivos)
<b>BackgroundImageLayout</b>	<i>Permite ajustar la imagen seleccionada a la ventana o hacerle zoom</i>	Stretch
<b>MaximizeBox</b>	<i>Habilita o deshabilita la opción maximizar del formulario</i>	True/false
<b>MinimizeBox</b>	<i>Habilita o deshabilita la opción minimizar del formulario</i>	True/false
<b>ControlBox</b>	<i>Habilita o deshabilita el cuadro de control completo del formulario</i>	True/false
<b>Font</b>	<i>Determina la fuente por defecto que tendrán los objetos a ubicar en el Form</i>	Microsoft Sans Serif; 8,25pt
<b>ForeColor</b>	<i>Define el color por defecto que tendrá el texto de los objetos a ubicar en el formulario</i>	Black
<b>Icon</b>	<i>Establece el ícono que identifica el formulario en la barra de títulos</i>	(en archivos)
<b>WindowState</b>	<i>Determina el estado en el que se inicia el formulario, normal, maximizado, minimizado.</i>	Normal
<b>ShowIcon</b>	<i>Especifica si se ve o no el ícono del formulario</i>	True/false

Recuerden, la sintaxis de las propiedades en líneas de código es: **objeto.propiedad = valor**

Ej: `label1.Text = "Hola Mundo";`

## EVENTOS USADOS CON MÁS FRECUENCIA

EVENTO	DESCRIPCIÓN/EJEMPLO
<b>Load</b>	<p>Es lo primero que se ejecutará antes de presentarse el formulario.</p> <pre>private void Form1_Load(object sender, EventArgs e) {     //mostrar un mensaje de bienvenida     MessageBox.Show("Bienvenido al programa"); }</pre>
<b>Clic</b>	<p>Este evento se ejecutará al dar clic sobre el control, en este caso sobre el formulario.</p> <pre>private void Form1_Clic(object sender, EventArgs e) {     //al hacer clic sobre el formulario, se maximiza     this.WindowState = FormWindowState.Maximized;     //this hace referencia a este formulario }</pre>
<b>MouseMove</b>	<p>Este evento indica que cuando pase el puntero del mouse por el formulario se ejecutará lo que este codificado en él.</p> <pre>private void Form1_MouseMove(object sender, MouseEventArgs e) {     //al mover el cursor sobre el formulario,     //éste cambia a color blanco     this.BackColor = Color.White; }</pre>
<b>FormClosing</b>	<p>Este evento se ejecutará cuando se esté cerrando la ventana.</p> <pre>private void Form1_FormClosing(object sender, FormClosingEventArgs e) {     //muestra un mensaje de despedida antes de cerrar     MessageBox.Show("Gracias por utilizar este programa. Hasta luego!"); }</pre>
<b>FormClosed</b>	<p>Este evento se ejecutará luego de cerrarse la ventana.</p> <pre>private void Form1_FormClosed(object sender, FormClosedEventArgs e) {     //muestra un mensaje luego de cerrar     MessageBox.Show("Que tengas buen día!"); }</pre>

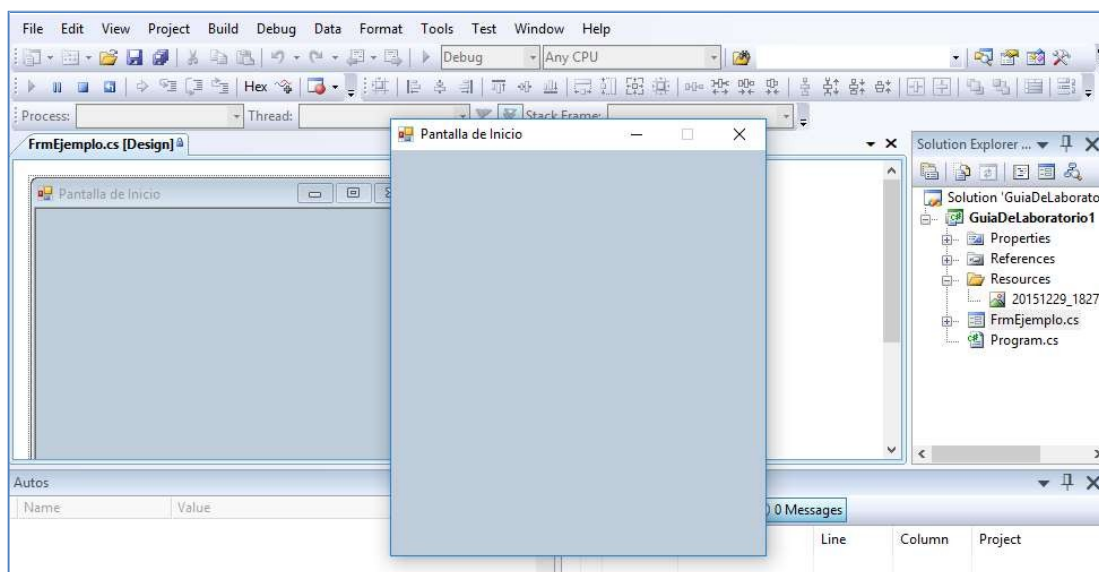
## MÉTODOS USADOS CON MÁS FRECUENCIA

EVENTO	DESCRIPCIÓN/EJEMPLO
<b>Close()</b>	Cierra el formulario. <pre>private void button1_Clic(object sender, EventArgs e) {     this.Close(); }</pre>
<b>Show()</b>	Muestra el formulario indicado. (Permite intercalar entre los formularios visibles al usuario) <pre>private void button1_Clic(object sender, EventArgs e) {     //primero se crea una instancia del Form 1     Form1 F1 = new Form1();     F1.Show(); }</pre>
<b>ShowDialog()</b>	Muestra el formulario como un cuadro de diálogo modal. (No se puede manipular otro formulario hasta cerrar el formulario modal) <pre>private void button1_Clic(object sender, EventArgs e) {     //primero se crea una instancia del Form 1     Form1 F1 = new Form1();     F1.ShowDialog(); }</pre>

✓ **ACTIVIDAD NUMERO 1:** Probemos con establecer las siguientes propiedades como lo indica el cuadro:

Form1	
<b>Name</b>	frmInicio
<b>Text</b>	Formulario de envío
<b>StartPosition</b>	CenterScreen
<b>FormBorderStyle</b>	Fixed3D
<b>BackColor</b>	InactiveCaption
<b>MaximizeBox</b>	False
<b>MinimizeBox</b>	True
<b>ControlBox</b>	True
<b>Font</b>	Trebuchet MS; 9,75pt
<b>ShowIcon</b>	False

El resultado al momento de ejecutar sería el siguiente:



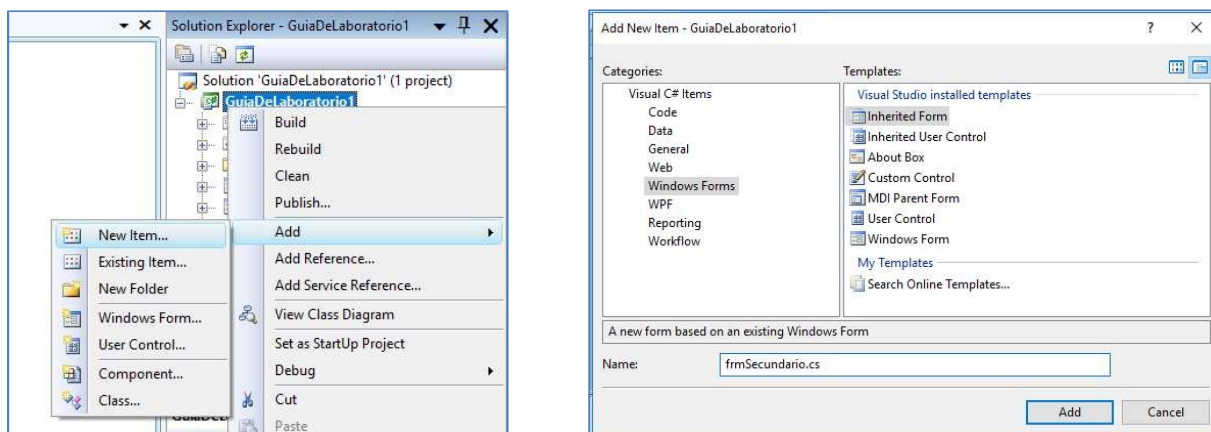
Dichas modificaciones, además del cambio de nombres y color, ubican el formulario en el centro de la pantalla, impiden que el usuario pueda modificar el tamaño de la ventana en ejecución, deshabilitando también el control “maximizar”.

## ✓ ACTIVIDAD NUMERO 2

Vamos a crear un nuevo formulario, al cual accederemos haciendo DobleClic en el primero. El nuevo formulario debe ser de las mismas características que el anterior y debe ser visible al igual que el primero. También deberán cambiar de color (aleatorio) cada vez que el cursor se mueva encima de ellos. Controlar que no se pueda mostrar más de una vez el segundo formulario. Al cerrarse el mismo deberá mostrarse un mensaje de despedida.

En primer lugar debemos crear un nuevo formulario. Lo haremos presionando el clic secundario sobre el proyecto (en el explorador de solución) y a continuación en Agregar (Add) y Nuevo ítem (New ítem).

Luego en la ventana de opciones seleccionamos Windows Forms e indicamos cual será el nombre del mismo:



Hecho esto, el formulario se visualizará con las propiedades por defecto como el primer formulario. Procedemos a modificar sus propiedades tal como el FrmEjemplo.

Una vez realizado esto, nos dirigimos al formulario FrmEjemplo y en su evento MouseMove colocamos lo siguiente:

```
private void FrmEjemplo_MouseMove(object sender, MouseEventArgs e)
{
    Random generador = new Random();
    int r = generador.Next(0, 256);
    int g = generador.Next(0, 256);
    int b = generador.Next(0, 256);
    this.BackColor = Color.FromArgb(r, g, b);
}
```

Lo que hacemos en estas líneas de código es crear tres valores aleatorios de tipo Random los cuales reemplazarán los valores de una estructura de color de 8 bits (rojo, verde y azul).

Copiamos el mismo código en el evento MouseMove del frmSecundario.

Para visualizar el formulario secundario y controlar que sólo se una sola vez, creamos una variable booleana como se ve a continuación:

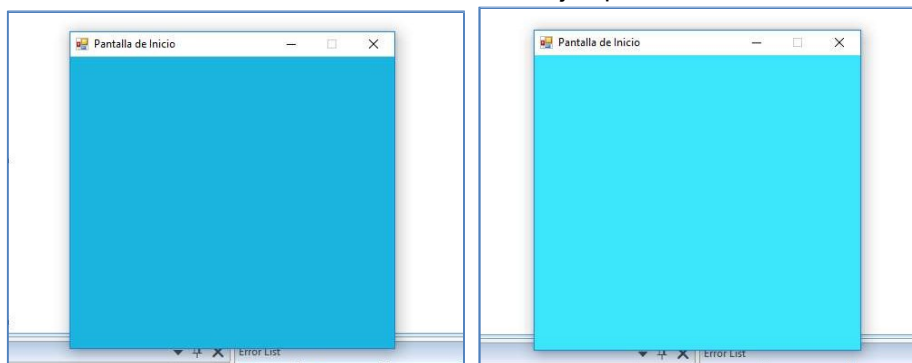
```
namespace GuiaDeLaboratorio1
{
    public partial class FrmEjemplo : Form
    {
        public FrmEjemplo()
        {
            InitializeComponent();
        }
        bool Mostrar = true;
    }
}
```

Luego en el evento DoubleClic del frmEjemplo insertamos el siguiente código:

```
private void FrmEjemplo_DoubleClic(object sender, MouseEventArgs e)
{
    //creamos una instancia de la clase "frmSecundario"
    frmSecundario F1 = new frmSecundario();
    //controlamos que la variable no haya cambiado aún
    //en caso de ser "false" significa que F1 ya fue abierto
    if (Mostrar == true)
    {
        F1.Show();
        Mostrar = false;
    }
}
```

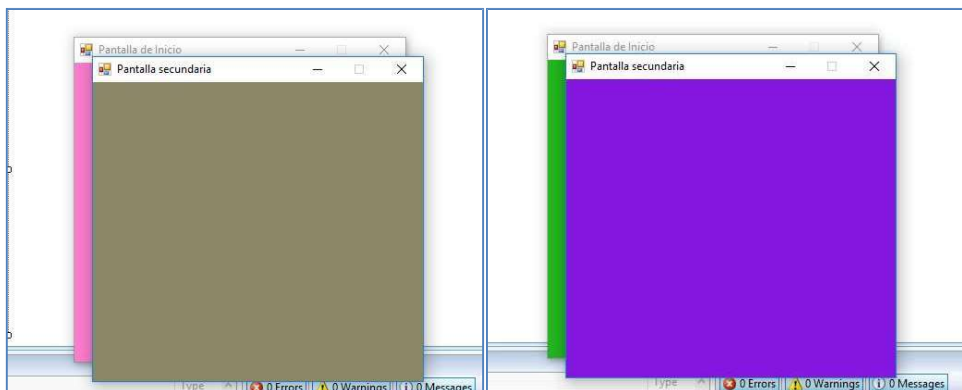
De esta forma, el programa funcionaría de la siguiente manera:

Al iniciar, cada movimiento del mouse sobre el frmEjemplo cambiará el color del mismo:



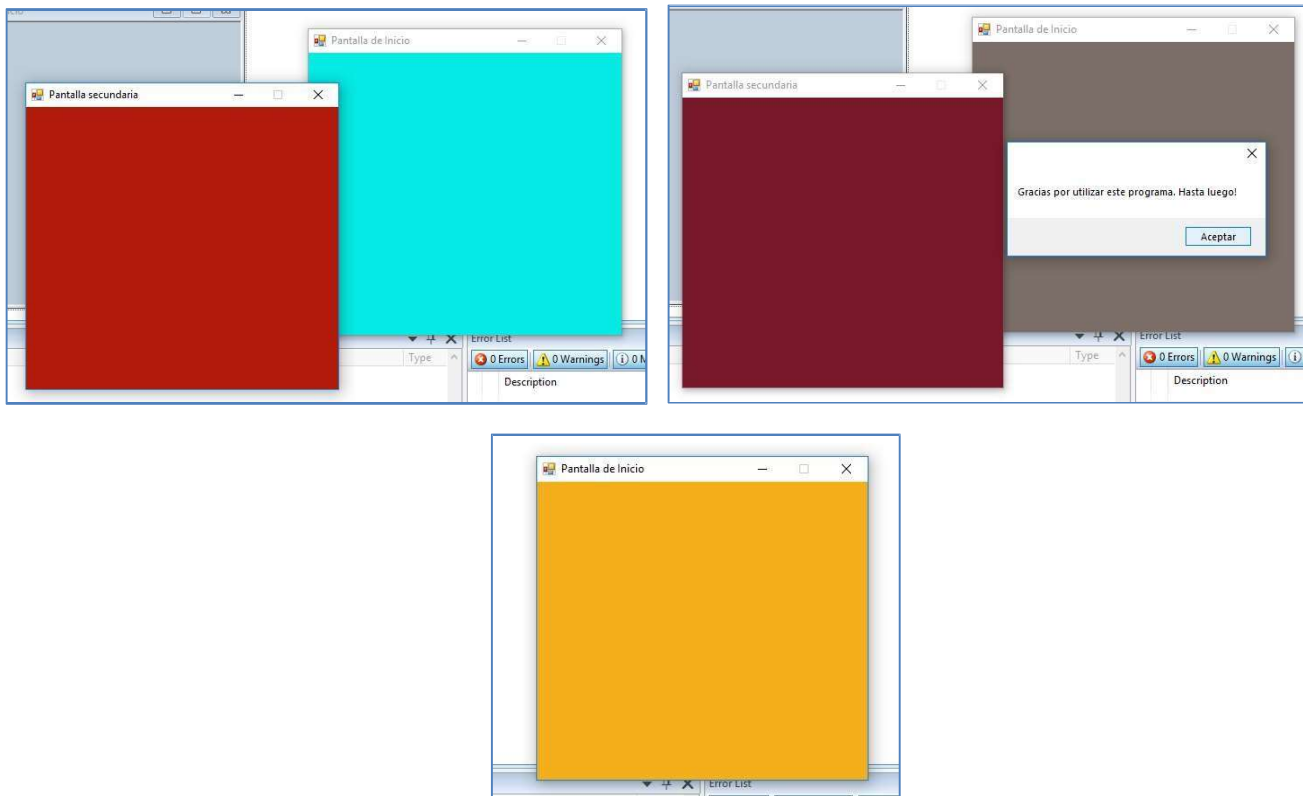


Haciendo dobleClic sobre él, se visualizará el frmSecundario, el cual también cambiará de color con cada movimiento del mouse sobre él



Ambos formularios pueden moverse y utilizarse, de modo que los dos continuarán cambiando de color.

Al cerrarse frmSecundario emergerá un mensaje de despedida y se verá nuevamente sólo el frmEjemplo:



Al cerrarse frmEjemplo se cierra el programa.

**2.1. Ejercicio práctico (no entregable):** Crear un formulario de nombre “Calcular vuelto”, que inicie en el centro de la ventana, que no permita cambiar sus dimensiones y tampoco maximizar. Guardar y reservar para realizar en él el ejercicio de la Guía 3. Se valorará cambio de color (uno que no dañe la vista) y el cambio de fuente, con un tamaño acorde a lo solicitado en el ejercicio 3.1. Recuerden que el diseño debe ser intuitivo y lo más agradable a la vista posible.