

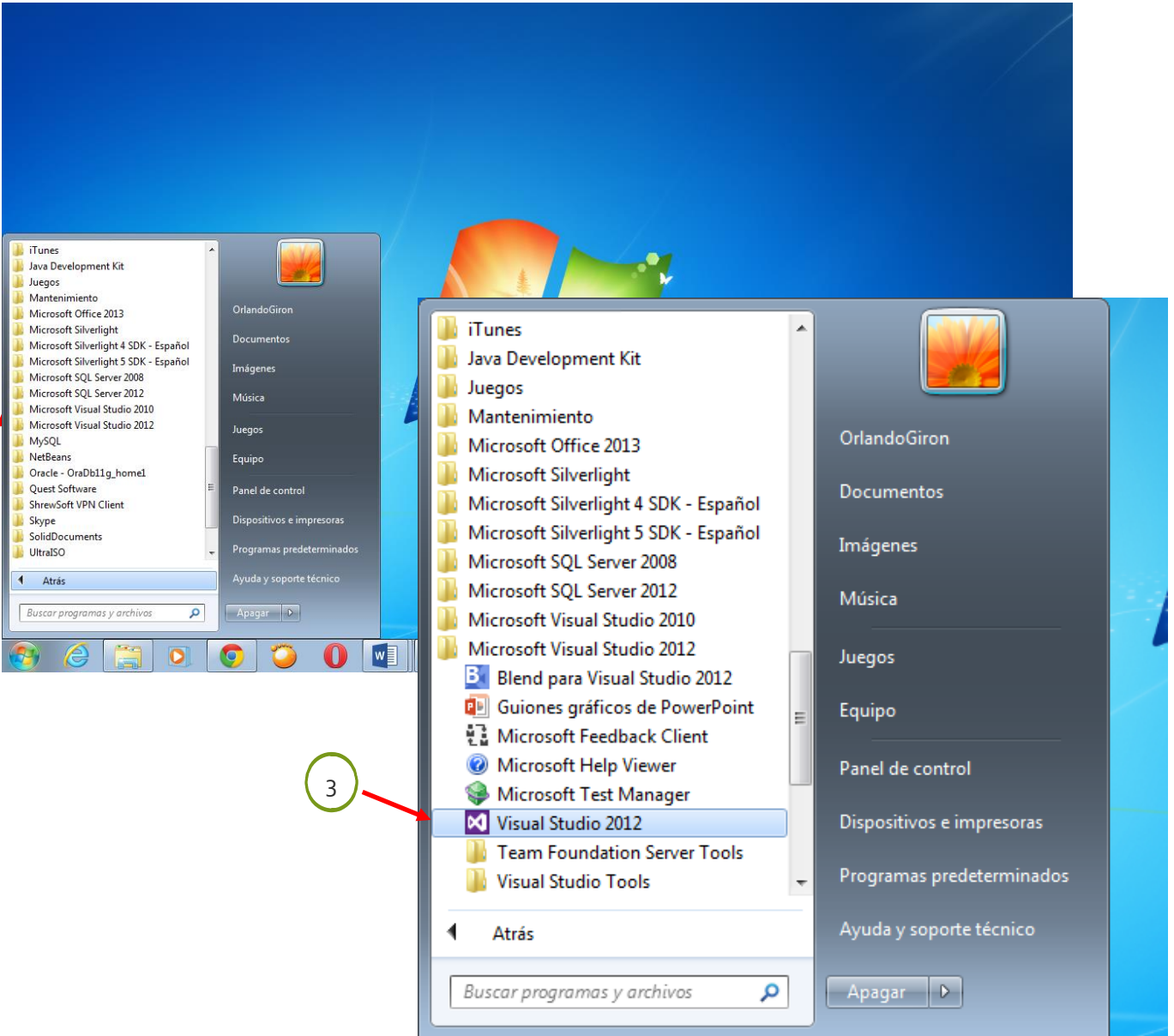
# Guía de ejercicios en C#

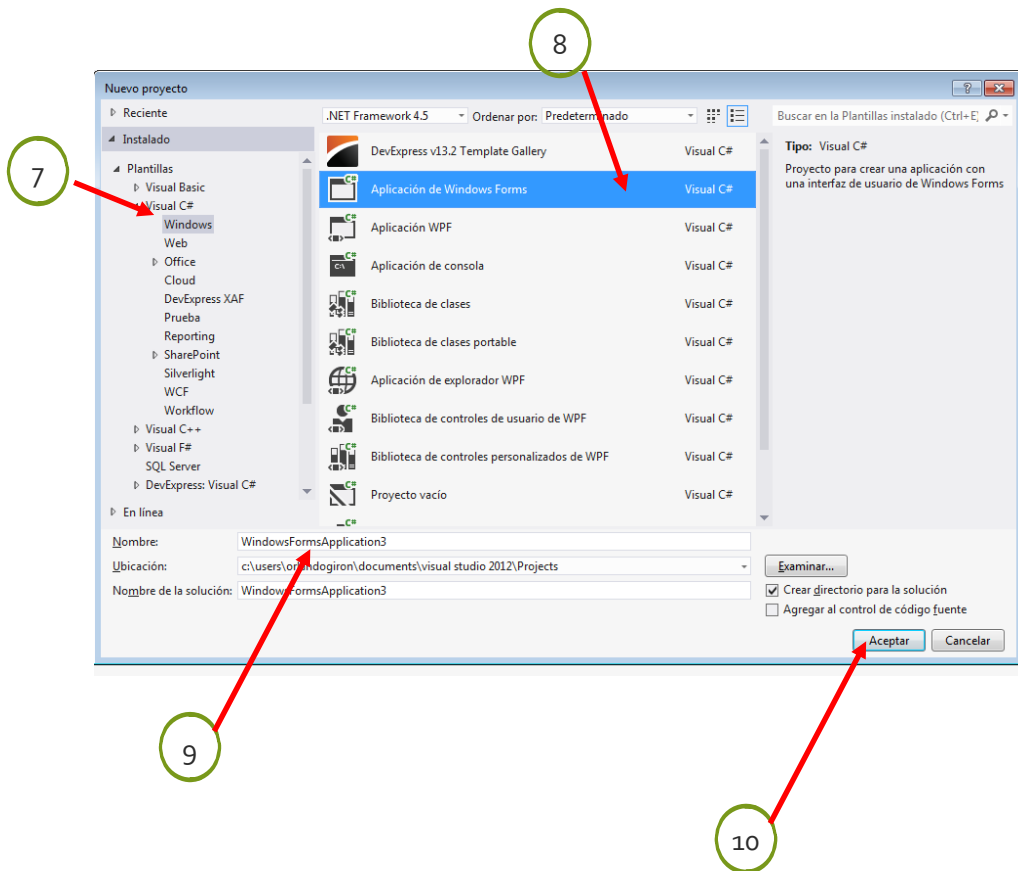
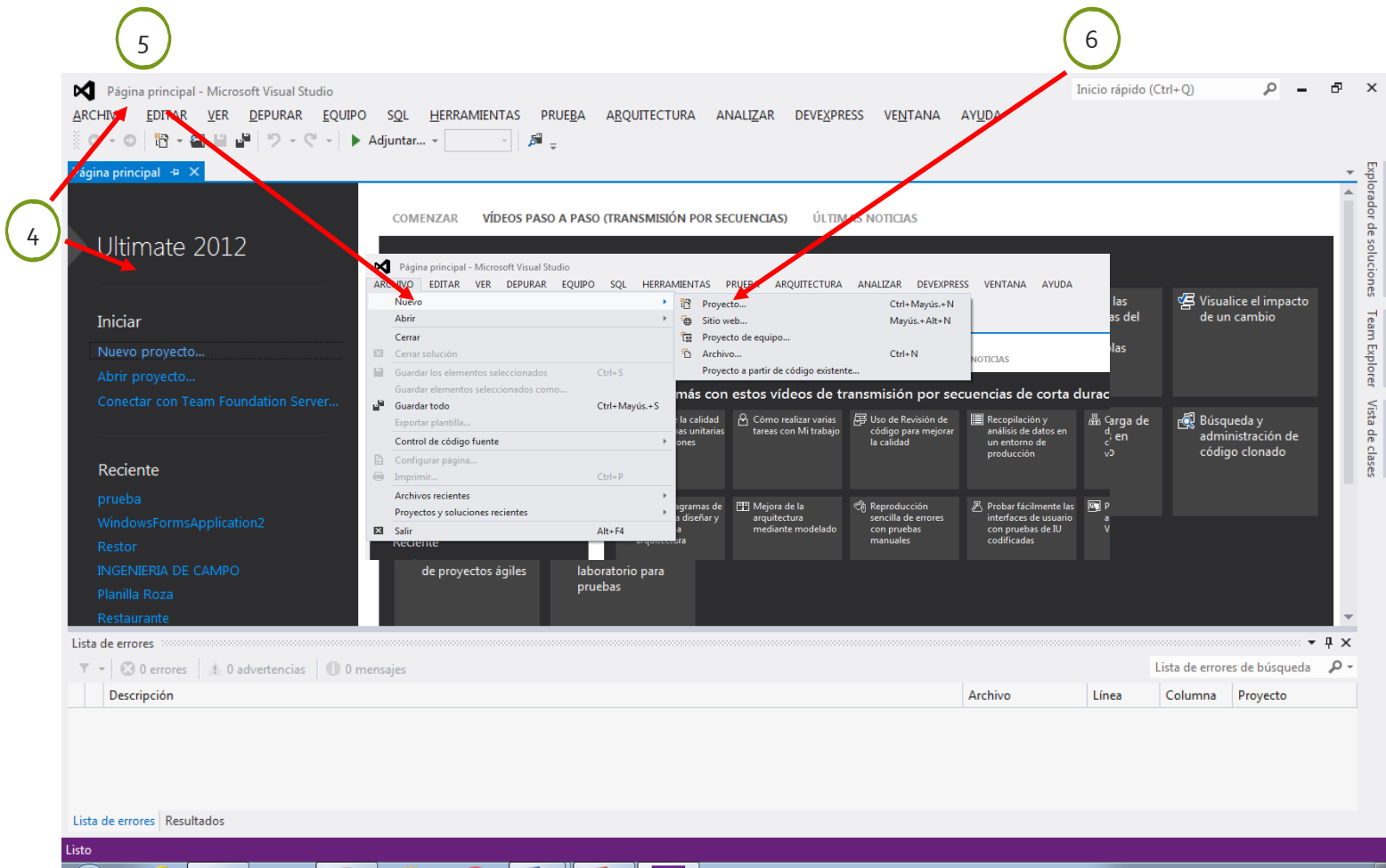
---

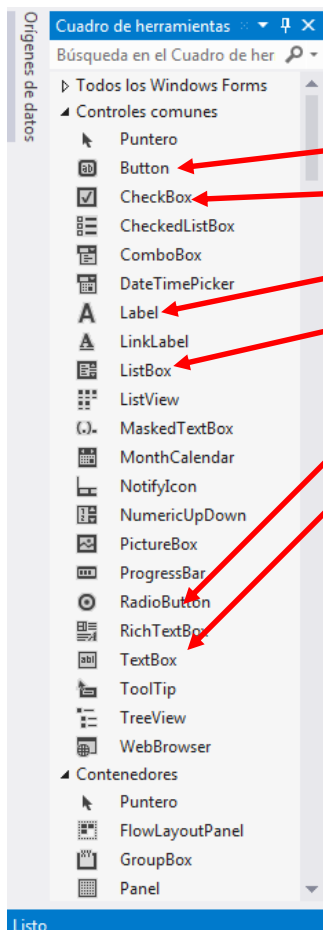
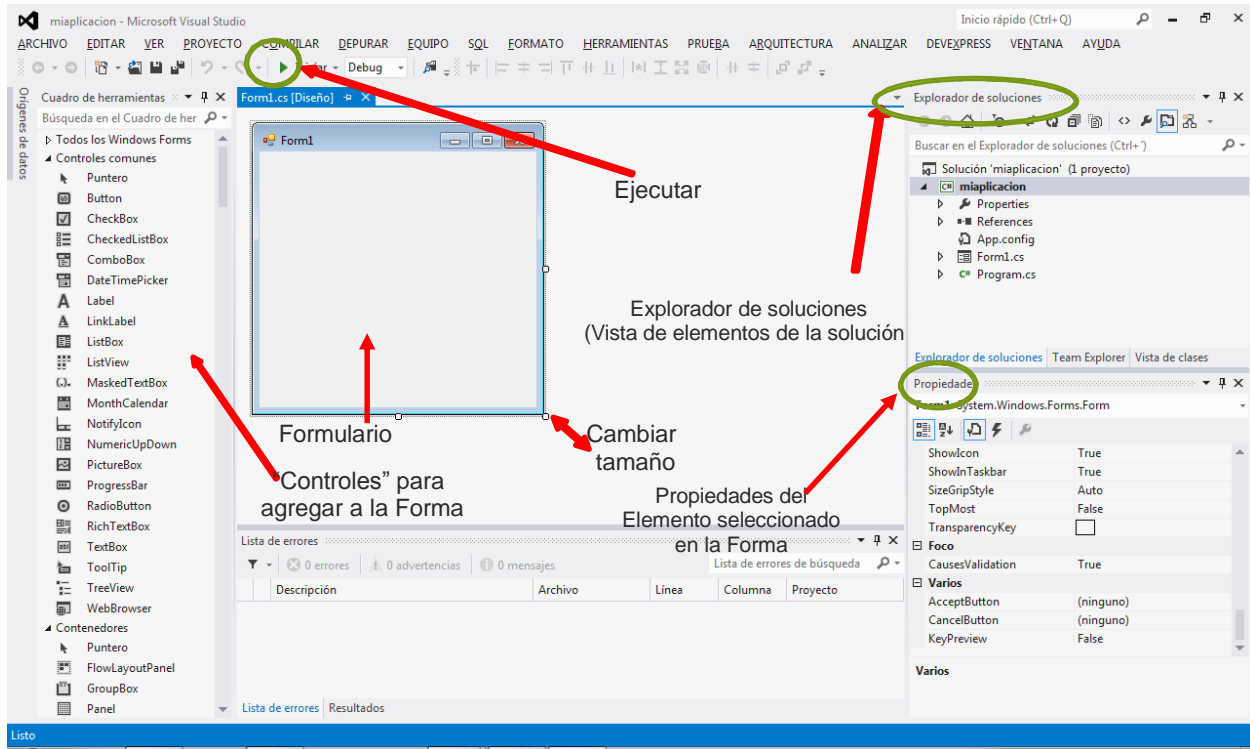
ING. ORLANDO GIRON / TEC. SAMUEL RODRIGUEZ

Programación 1

UNIVERSIDAD TECNOLÓGICA DE EL SALVADOR | FACULTAD DE INFORMÁTICA Y CIENCIAS  
APLICADAS







## Algunos de los controles más usados

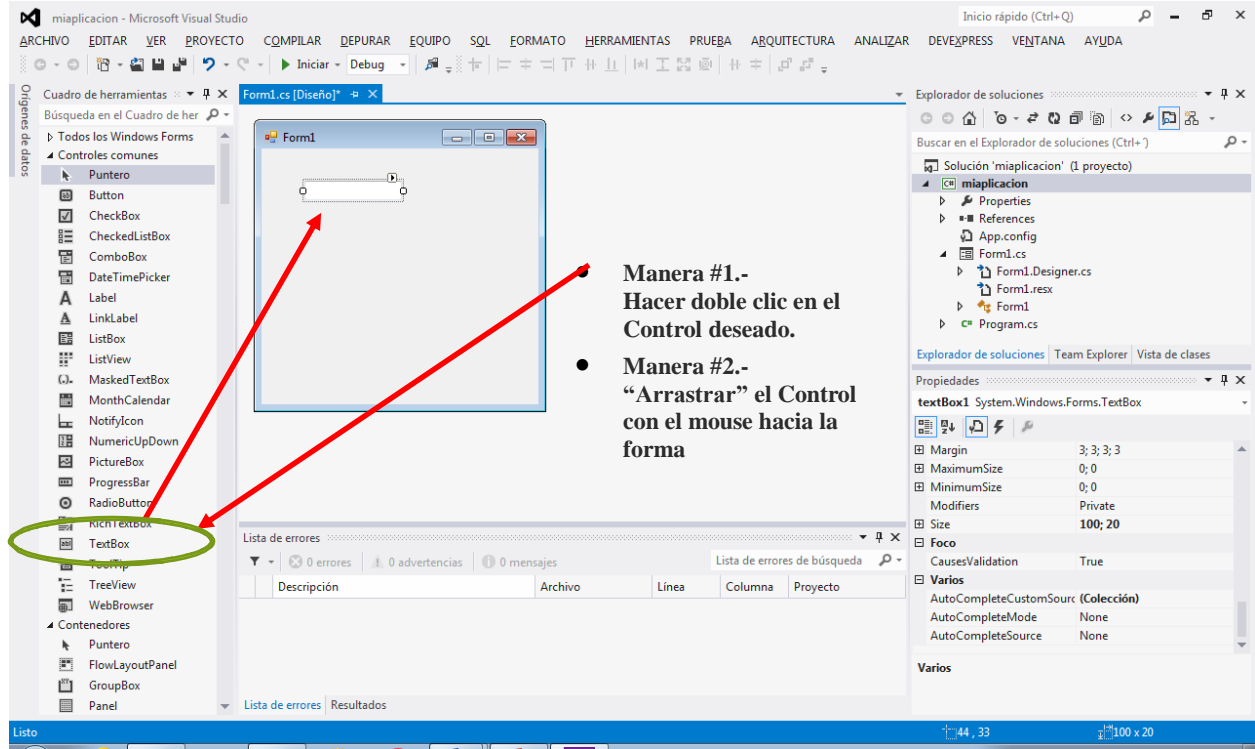
- BUTTON (BOTON DE COMANDOS).**-Ejecuta instrucciones al ser oprimido
- CHECKBOX (CHECK).**-Permite seleccionar varias opciones simultáneamente
- LABEL (ETIQUETA).**-Muestra mensajes fijos en la Forma
- LISTBOX (LISTA).**-Muestra mensajes fijos en la Forma
- RADIOBUTTON (RADIO BOTON).**-Permite seleccionar una sola opción entre varias.
- TEXTBOX (CUADRO DE TEXTO).**-Permite al usuario introducir / recibir datos del programa.

## Solución-proyecto-formulario

- Una Solución puede contener uno o más proyectos.
- Un Proyecto puede contener una o más Formas
- Un Formulario puede contener uno o más “controles”.

**El Explorador de Soluciones** muestra de manera gráfica los archivos involucrados en Soluciones, Proyectos y Formas.

# CONTROLES AL FORMULARIO

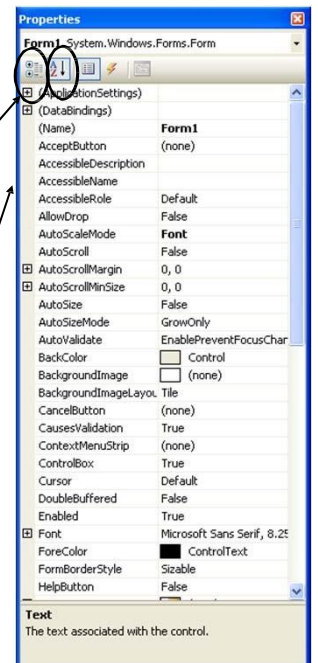


## Personalizando los controles (Propiedades)

- Una propiedad es una característica de un objeto (color, tamaño, tipo de letra, etc.).
- La ventana de “Propiedades” muestra todas las propiedades del objeto que se encuentra seleccionado con el mouse, y permite modificar sus valores.

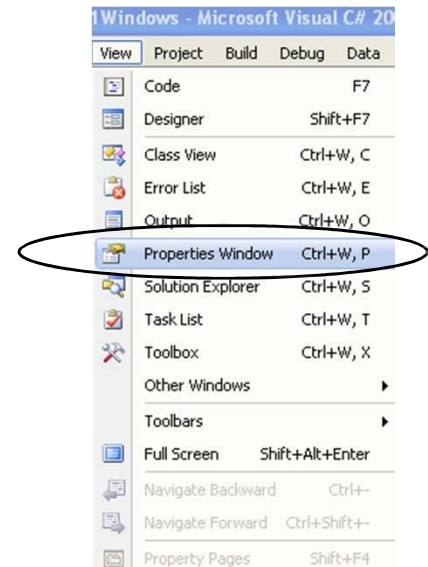
*Ordenar propiedades por categorías*

*Ordenar propiedades alfabéticamente*



## Si no aparece la ventana de propiedades...

- Seleccionar del menú:  
Ver/ventana de propiedades
- Presionar la tecla F4



## Propiedades "NAME" y "TEXT"

- Todos los controles tienen una propiedad NAME que los identifica de manera única en el programa. No se puede tener controles con los mismos "NAMES".
- La propiedad "TEXT" generalmente indica el texto que será desplegado en el control. Sí pueden existir controles con los mismos "TEXT".
- EVITAR confundir NAME y TEXT.

# EJERCICIO 1

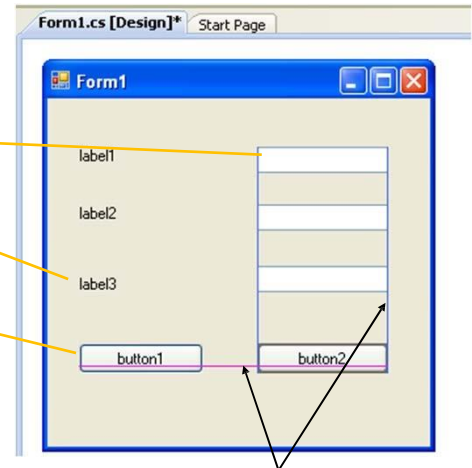
## ENUNCIADO

- Realizar una interfaz gráfica que permita al usuario introducir dos números.
- El programa calculará y desplegará la suma de ambos.
- Se requieren 3 etiquetas, 3 cuadros de texto y dos botones.

Caja de Texto

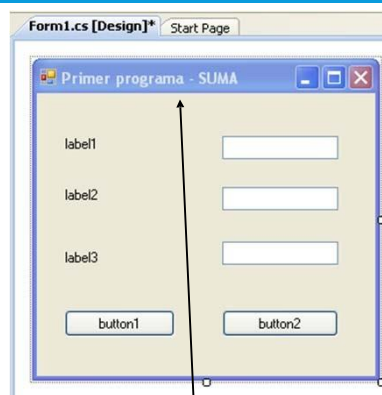
Etiqueta

Botón

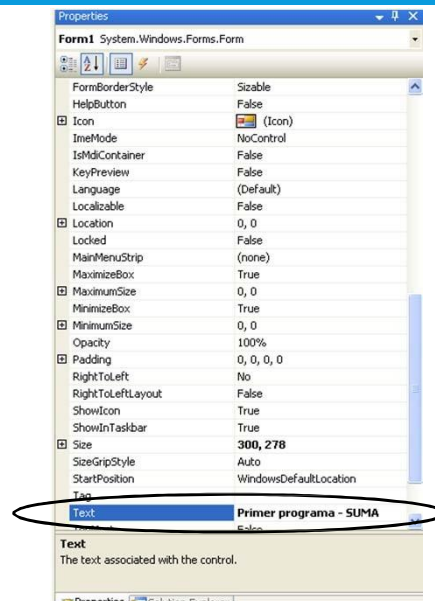


Notar las “Líneas guía” que el editor muestra al acomodar los controles para alinearlos fácilmente.

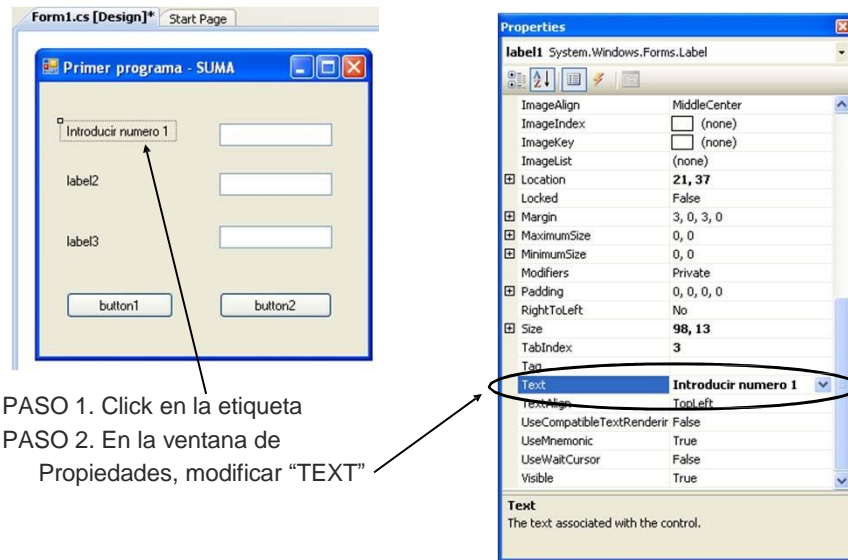
## PONIENDOLE TÍTULO AL FORMULARIO



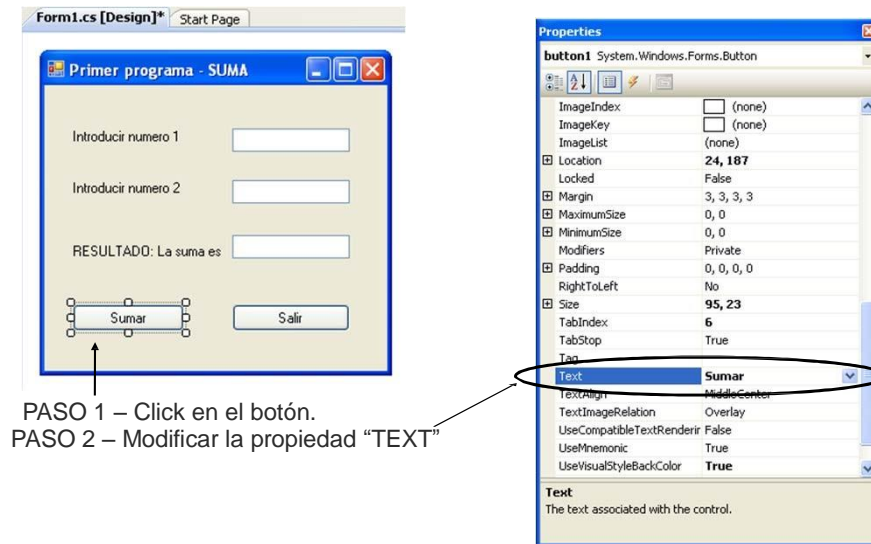
PASO 1 - Clic en el formulario  
PASO 2 – Modificar la propiedad “TEXT”



# PONIENDOLE TEXTO A LAS ETIQUETAS



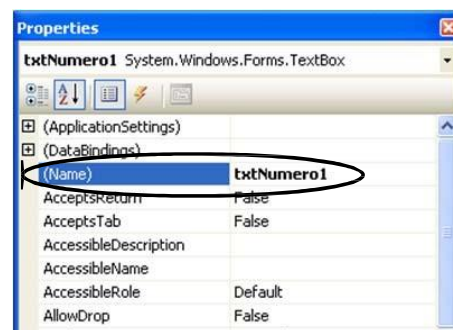
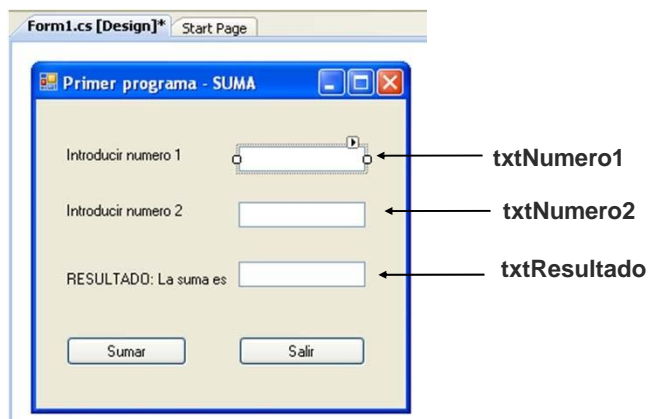
# PONIENDOLE TEXTO A LOS BOTONES





# NOMBRE A LAS CAJAS DE TEXTO

- PASO 1 – Click en el cuadro de texto
- PASO 2 – Cambiar la propiedad  
“NAME” En el Ejercicio, utilizar los  
siguientes nombres:



# COLOCANDO CODIGO A LOS BOTONES

PASO 1 – DOBLE CLICK en el botón donde se tecleará código

PASO 2 – digitar el siguiente código del programa

```
private void button1_Click(object sender, EventArgs e)
{
    int n1, n2, suma;
    n1 = int.Parse(txtNumero1.Text);
    n2 = int.Parse(txtNumero2.Text);
    suma = n1 + n2;
    txtResultado.Text = suma.ToString();
}

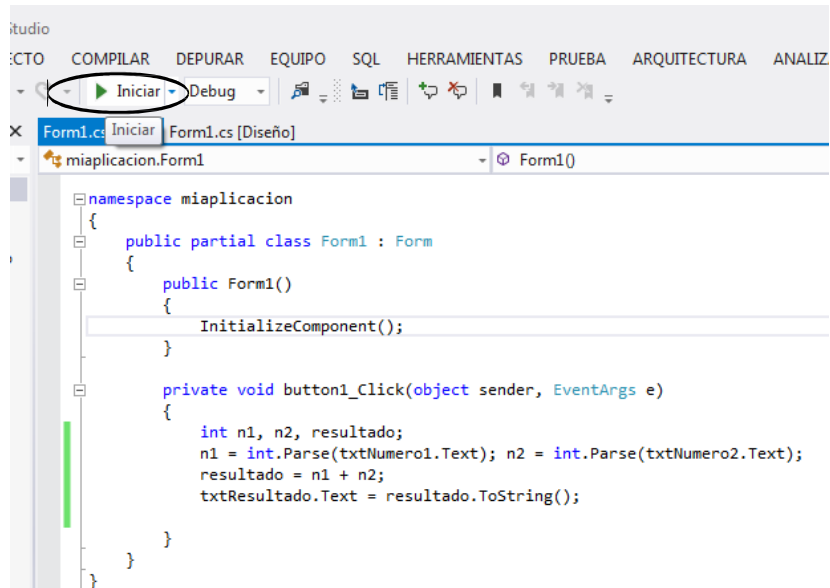
private void button2_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

Botón para SUMAR

Botón para SALIR

# EJECUTANDO EL PROGRAMA

- Manera #1 – Presionar TECLA F5
- Manera #2 – Presionar botón PLAY (START)



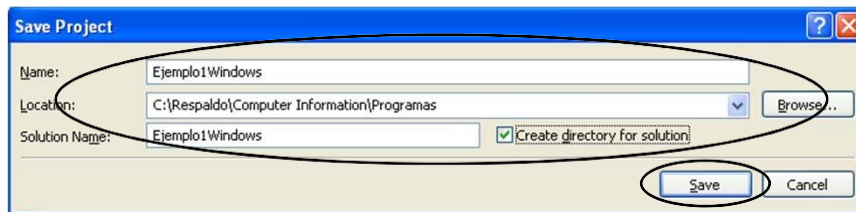
## EJERCICIOS

- Agregar un botón más a la forma para “Limpiar” el contenido de los cuadros de texto. Con el siguiente código:  

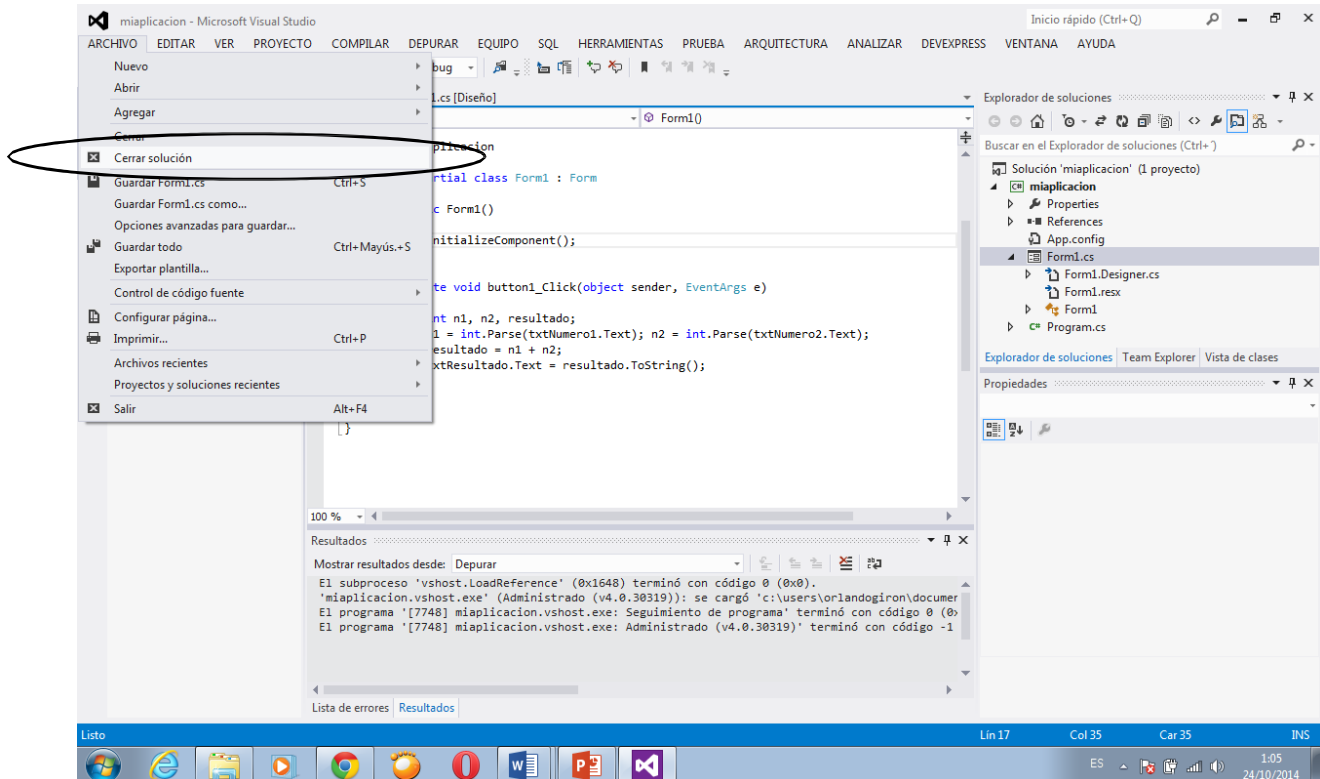
```
txtNumero1.Text = "";  
txtNumero2.Text = "";  
txtResultado.Text = "";
```
- Ejecutar el programa; posteriormente cambiar el código anterior por el siguiente y comprobar que hacen lo mismo:  

```
txtNumero1.Clear();  
txtNumero2.Clear();  
txtResultado.Clear();
```

# GUARDAR EL PROYECTO



# CERRAR EL PROYECTO



## EJERCICIO<sub>2</sub>

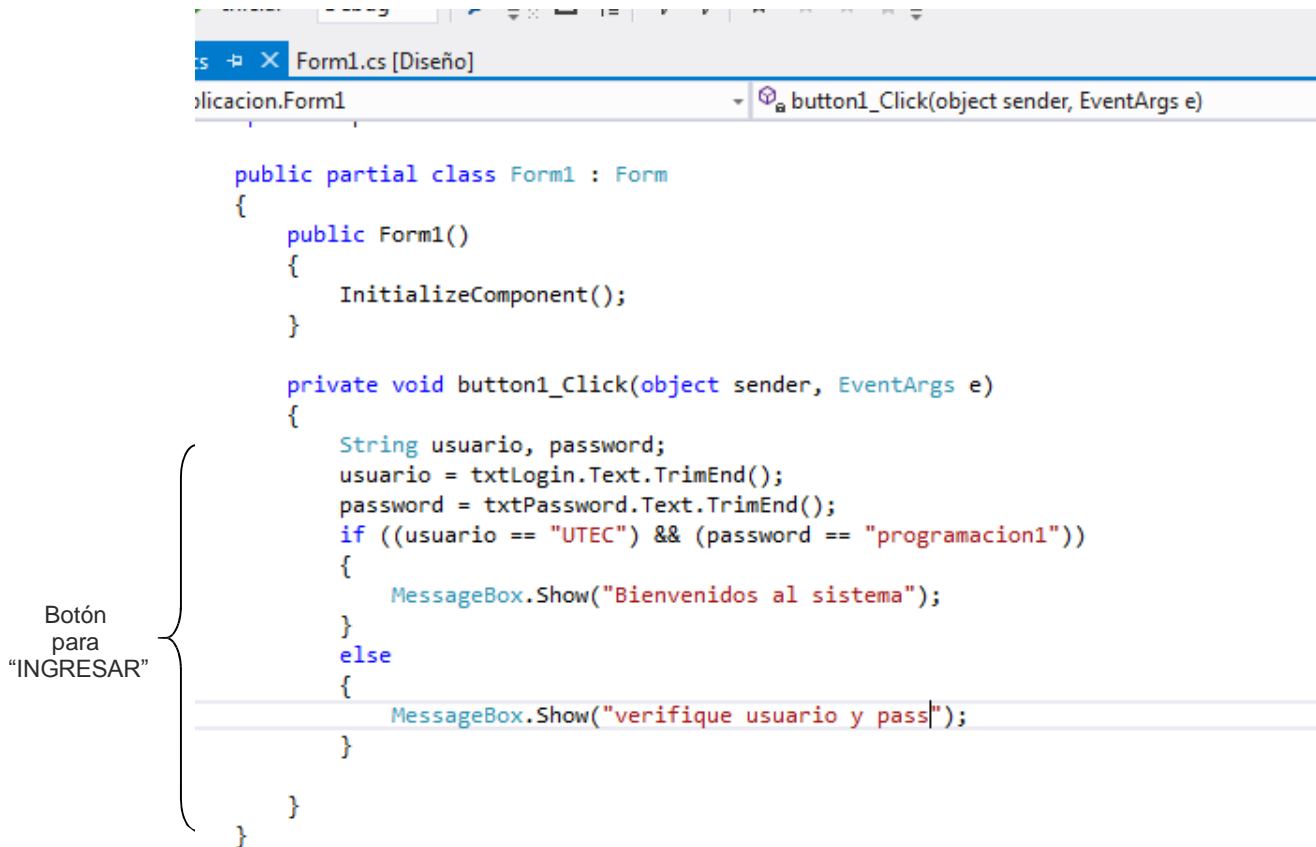
- Realizar una pantalla que pida "Login" y "Password" a un usuario. Mostrar un mensaje de "Bienvenida" si los datos son correctos, o un mensaje de "Rechazo" si no lo son.
- Datos correctos:
  - Login: "UTEC"
  - Password: "programacion1"
  - (Tomar en cuenta Mayúsculas y minúsculas)
  - Diseñar la siguiente pantalla



## CODIGO DEL PROGRAMA

Notar que...

- El método `TrimEnd()` elimina los espacios en blanco hasta el final.

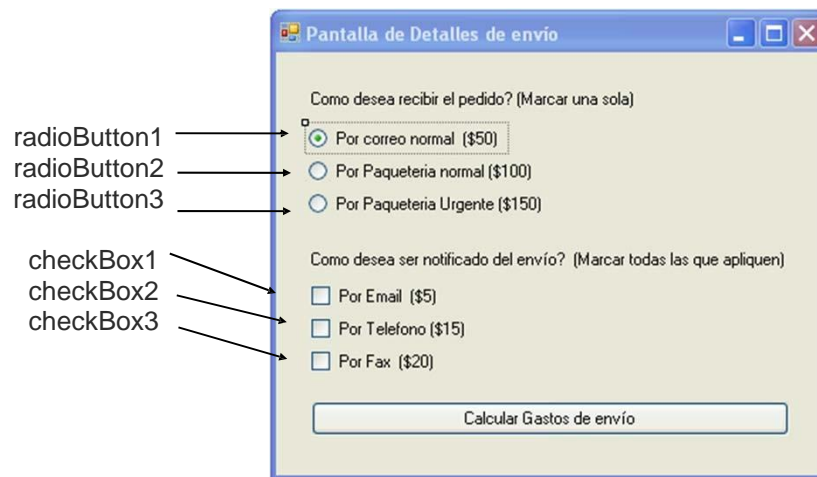


- MessageBox.Show ("Mensaje") muestra una ventana con un mensaje para el usuario.

## EJERCICIO3

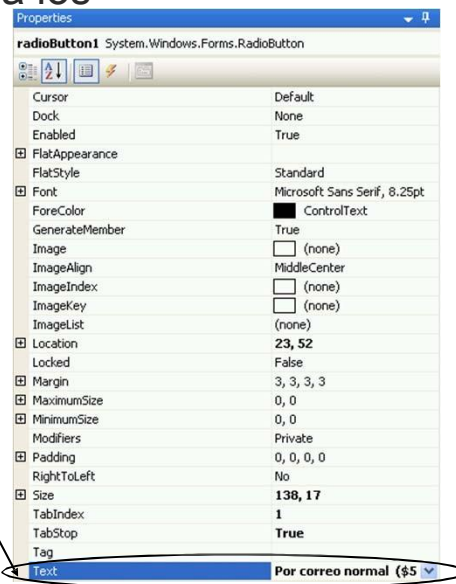
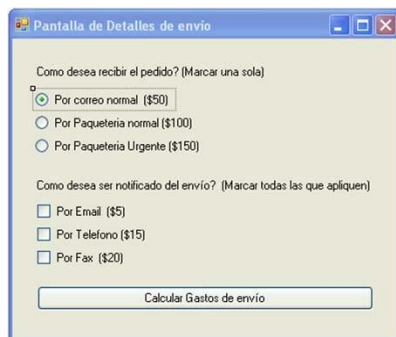
Realizar una Pantalla para calcular el costo de los gastos de envío de un pedido dependiendo de las opciones seleccionadas. Mostrar el monto total al presionar el botón "Calcular Gastos de Envío"

El primer radio botón debe estar marcado al iniciar el programa.



Poniendo texto a los radiobotones y a los checkboxes.

1. Click en el radioboton o en el CheckBox
2. En la ventana de propiedades, poner en la propiedad "Text" el texto que se desplegará en ellos.



## Activando un radio botón / checkbox

1. Click en el radio botón
2. Cambiar la propiedad "CHECKED" a TRUE

Pantalla de Detalles de envío

Como desea recibir el pedido? (Marcar una sola)

☒ Por correo normal (\$50)

☐ Por Paquetería normal (\$100)

☐ Por Paquetería Urgente (\$150)

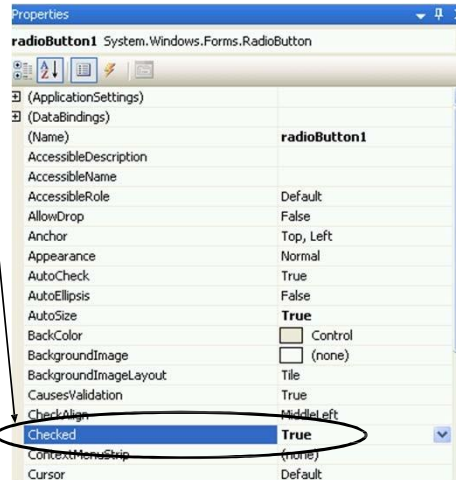
Como desea ser notificado del envío? (Marcar todas las que apliquen)

☐ Por Email (\$5)

☐ Por Teléfono (\$15)

☐ Por Fax (\$20)

Calcular Gastos de envío



```
int total = 0;
if (radioButton1.Checked == true)
{
    total = total + 50;
}
if (radioButton2.Checked == true)
{
    total = total + 100;
}
if (radioButton3.Checked == true)
{
    total = total + 150;
}
if (this.checkBox1.Checked == true)
{
    total = total + 5;
}
if (this.checkBox2.Checked == true)
{
    total = total + 15;
}
if (this.checkBox3.Checked == true)
{
    total = total + 20;
}
MessageBox.Show (" El total de gastos de envío es: " + total.ToString("c2"));
```

Pantalla de Detalles de envío

Como desea recibir el pedido? (Marcar una sola)

☒ Por correo normal (\$50)

☐ Por Paquetería normal (\$100)

☐ Por Paquetería Urgente (\$150)

Como desea ser notificado del envío? (Marcar todas las que apliquen)

☒ Por Email (\$5)

☐ Por Teléfono (\$15)

☒ Por Fax (\$20)

Calcular Gastos de envío

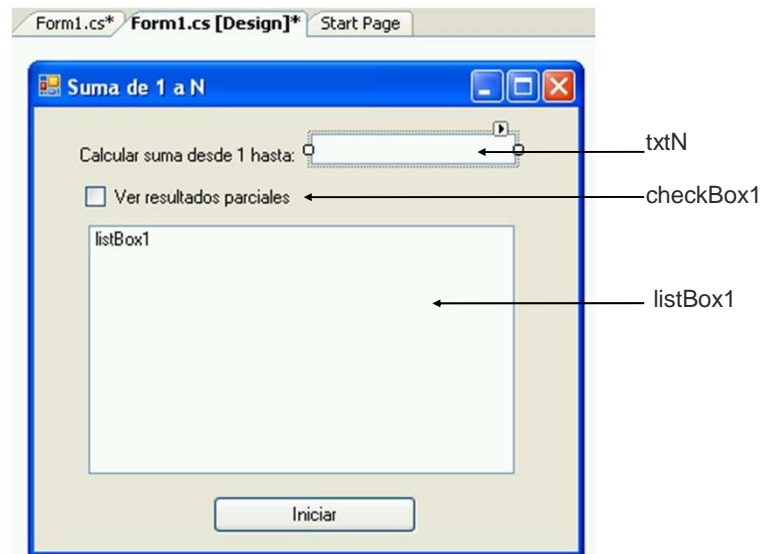
El total de gastos de envío es: \$75.00

OK

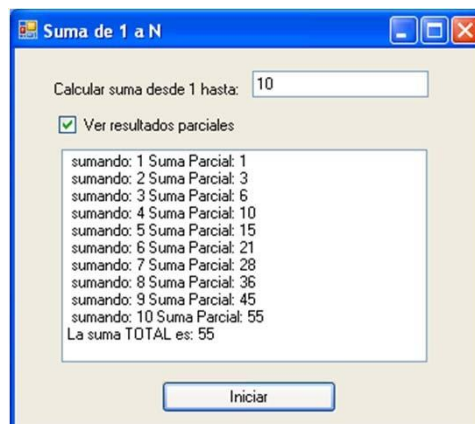


# EJERCICIO4

- Realizar una interfaz de usuario donde se calcule la suma de los numeros desde 1 hasta donde el usuario indique.
- Imprimir los resultados en una lista (ListBox).
- Utilizar un checkbox para que el usuario indique si quiere ver todos los resultados parciales o solo el resultado final.



```
int x, N, suma=0;
listBox1.Items.Clear();
N = int.Parse(txtN.Text);
for (x = 1; x <= N; x++)
{
    suma = suma + x;
    if (checkBox1.Checked == true)
    {
        listBox1.Items.Add("sumando: " + x + " Suma Parcial: " + suma);
    }
}
listBox1.Items.Add("La suma TOTAL es: " + suma);
```

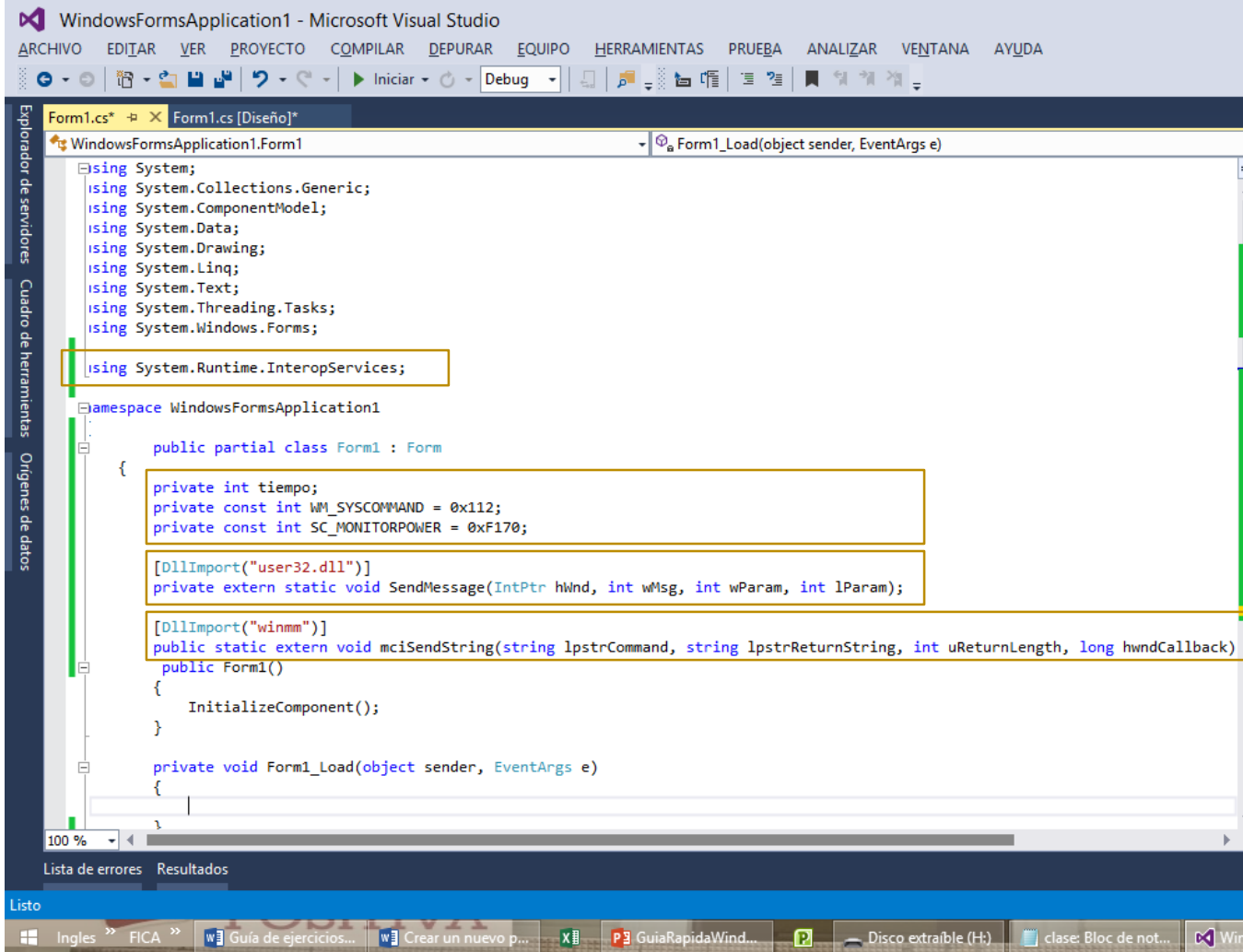


# EJERCICIO4

- Crear el siguiente formulario



The image shows a screenshot of a Windows application window titled "Uso de dll". The window has a standard Windows XP-style title bar with minimize, maximize, and close buttons. The main content area is light gray and contains four buttons arranged in a 2x2 grid. The buttons are labeled "apagar monitor", "apagar monitor con tiempo", "Abrir CD", and "Cerrar CD". The buttons are simple gray rectangles with black text.



WindowsFormsApplication1 - Microsoft Visual Studio

ARCHIVO EDITAR VER PROYECTO COMPILAR DEPURAR EQUIPO HERRAMIENTAS PRUEBA ANALIZAR VENTANA AYUDA

Form1.cs\* Form1.cs [Diseño]\*

WindowsFormsApplication1.Form1 Form1\_Load(object sender, EventArgs e)

```
private void ApagarMonitor()
{
    SendMessage(this.Handle, WM_SYSCOMMAND, SC_MONITORPOWER, 2);
    //El 2 es para apagar
}

private void EncenderMonitor()
{
    SendMessage(this.Handle, WM_SYSCOMMAND, SC_MONITORPOWER, -1);
    //El -1 es para encender
}

private void button1_Click(object sender, EventArgs e)
{
    ApagarMonitor();
}

private void button2_Click(object sender, EventArgs e)
{
    tiempo = 5;
    ApagarMonitor();
    timer1.Start();
}

private void timer1_Tick(object sender, EventArgs e)
{
    if (tiempo != 0)
    {
        tiempo--;
    }
    else
    {

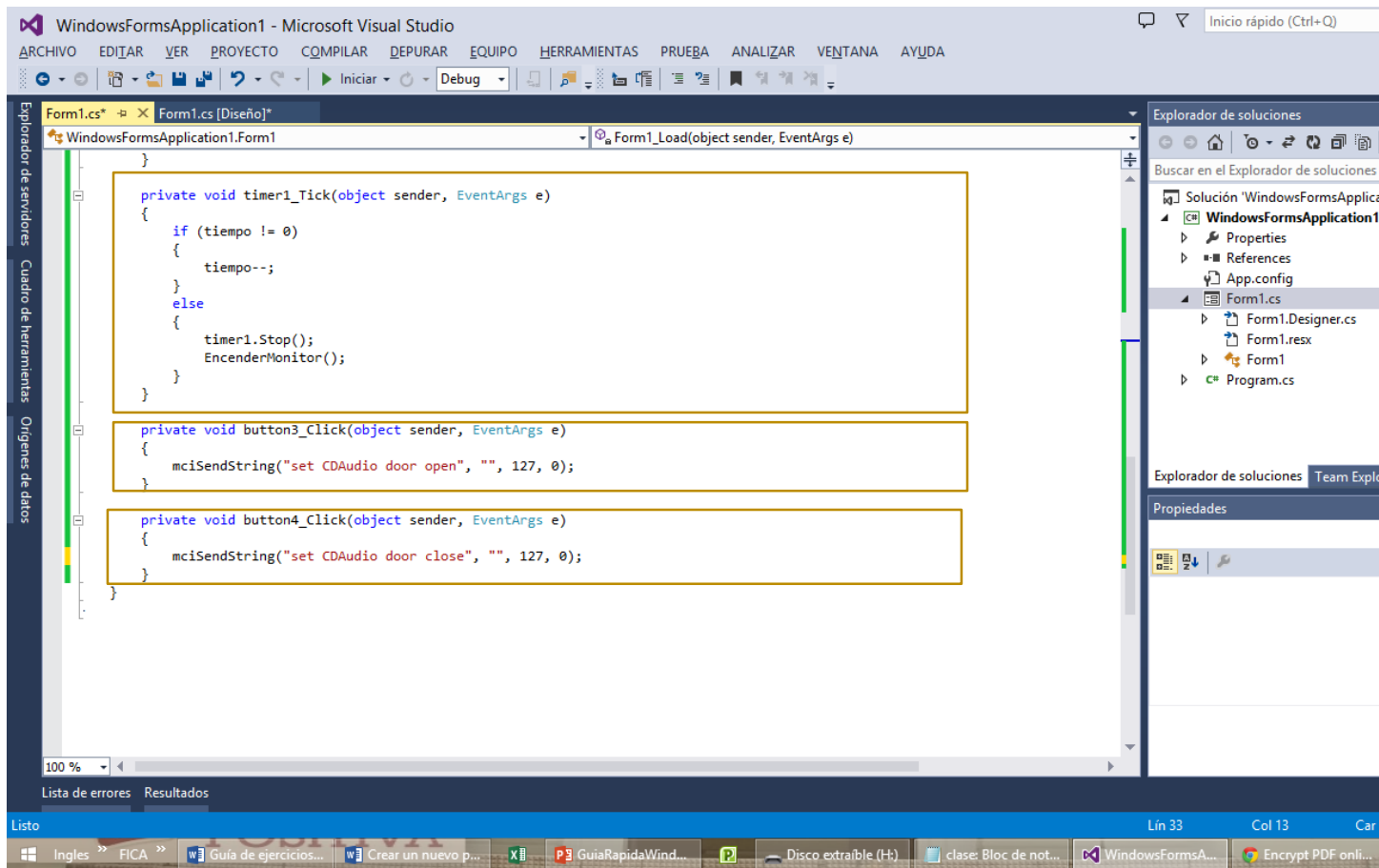
```

100 %

Lista de errores Resultados

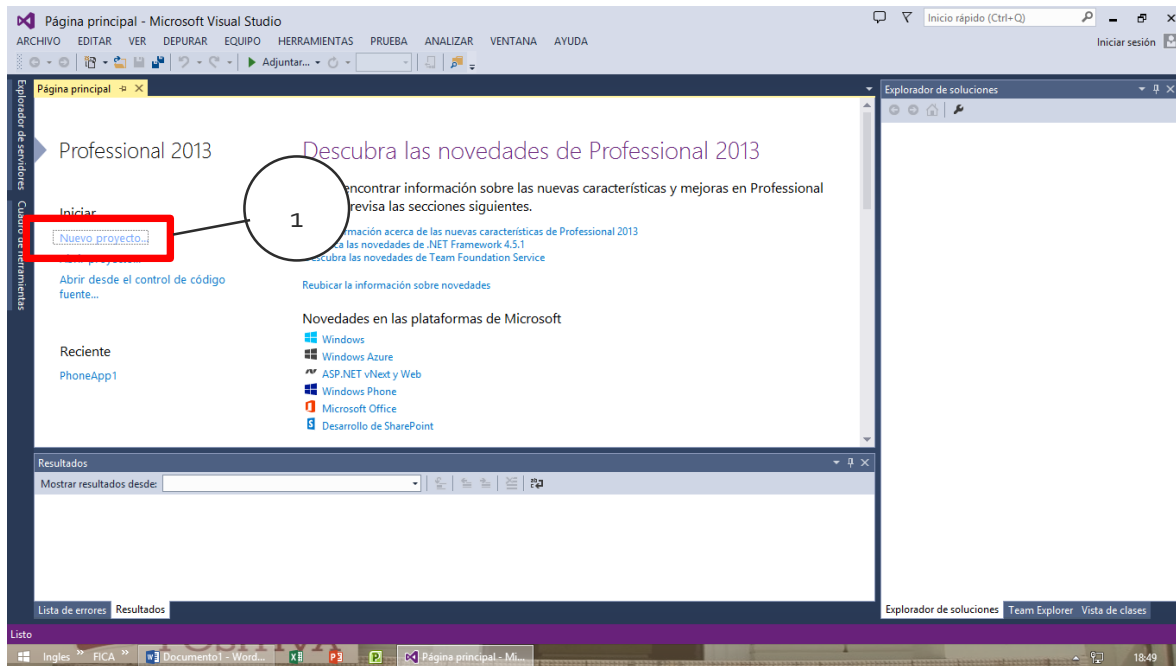
Listo Lin 33

WindowsFormsApplication1

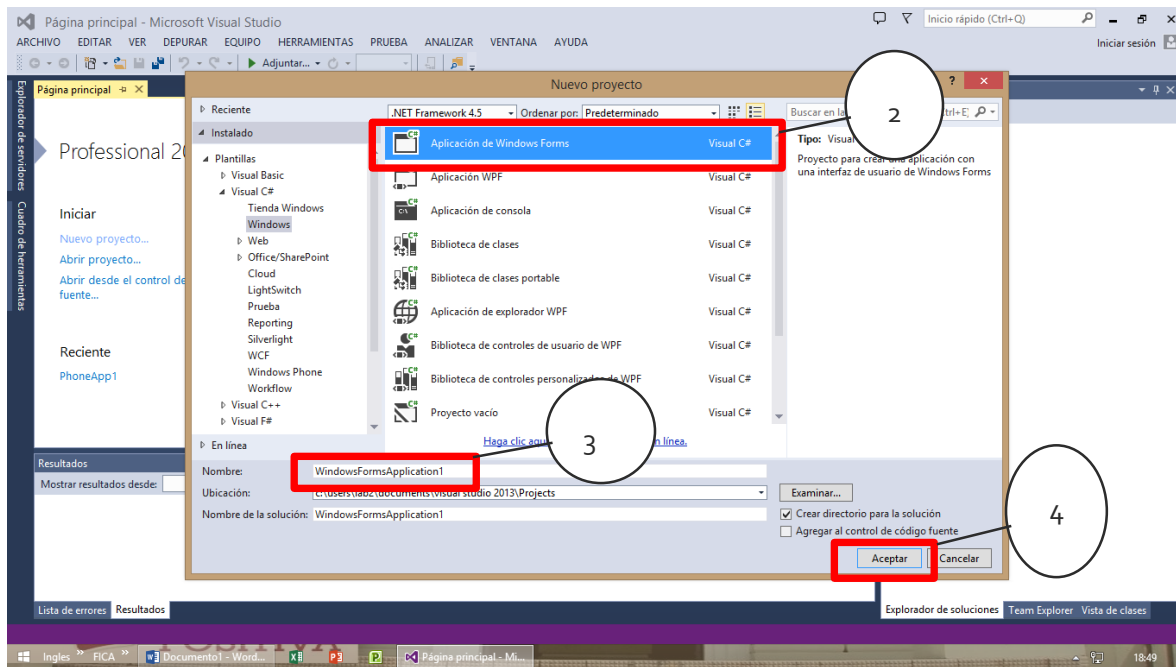


## EJERCICIO5

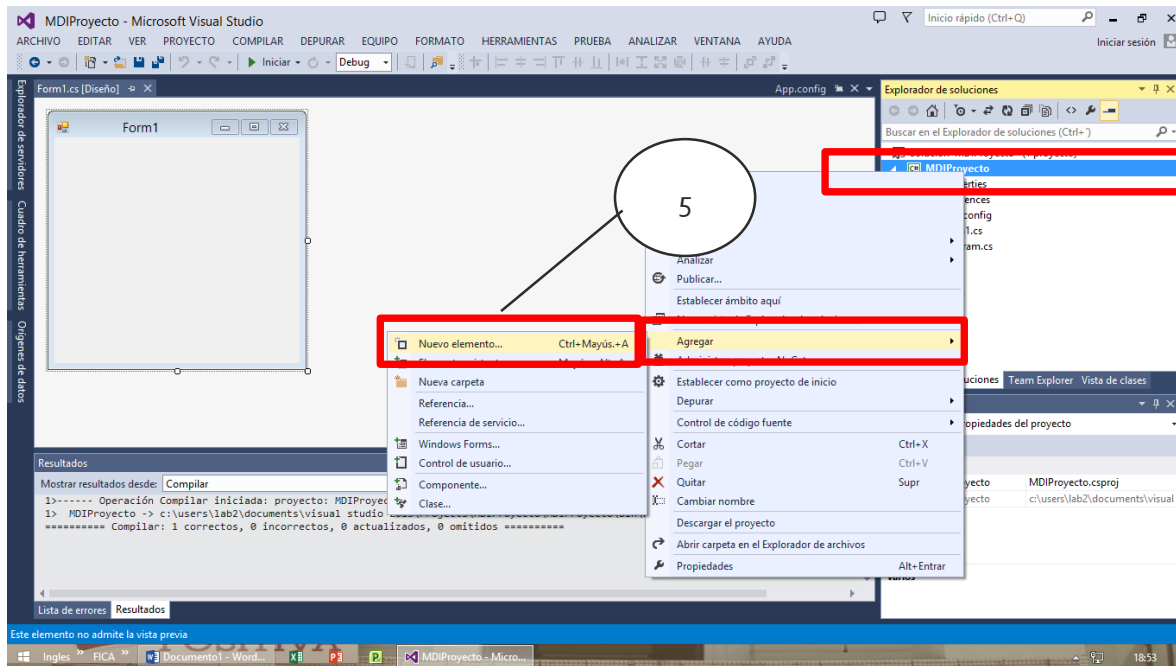
- Antes del MDI colocar un formulario con Progress bar
- Realizar una interfaz MDI y colocar todos los ejercicios en el mdi



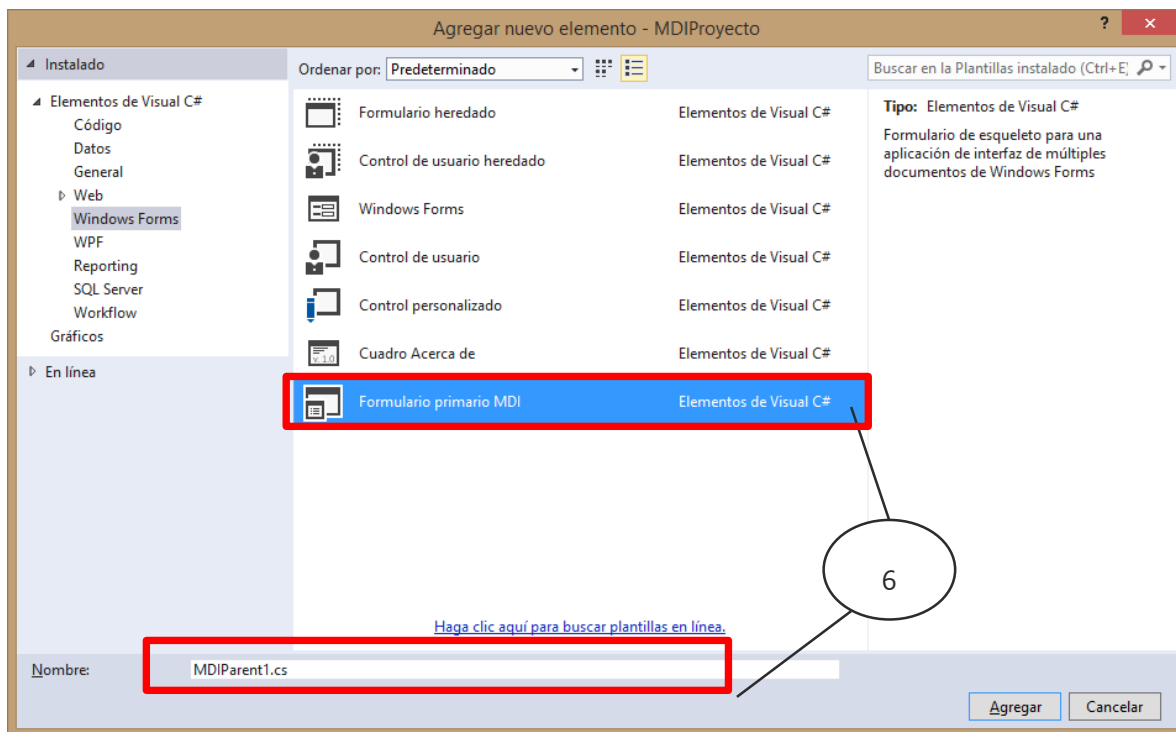
## 1- Crear un nuevo proyecto



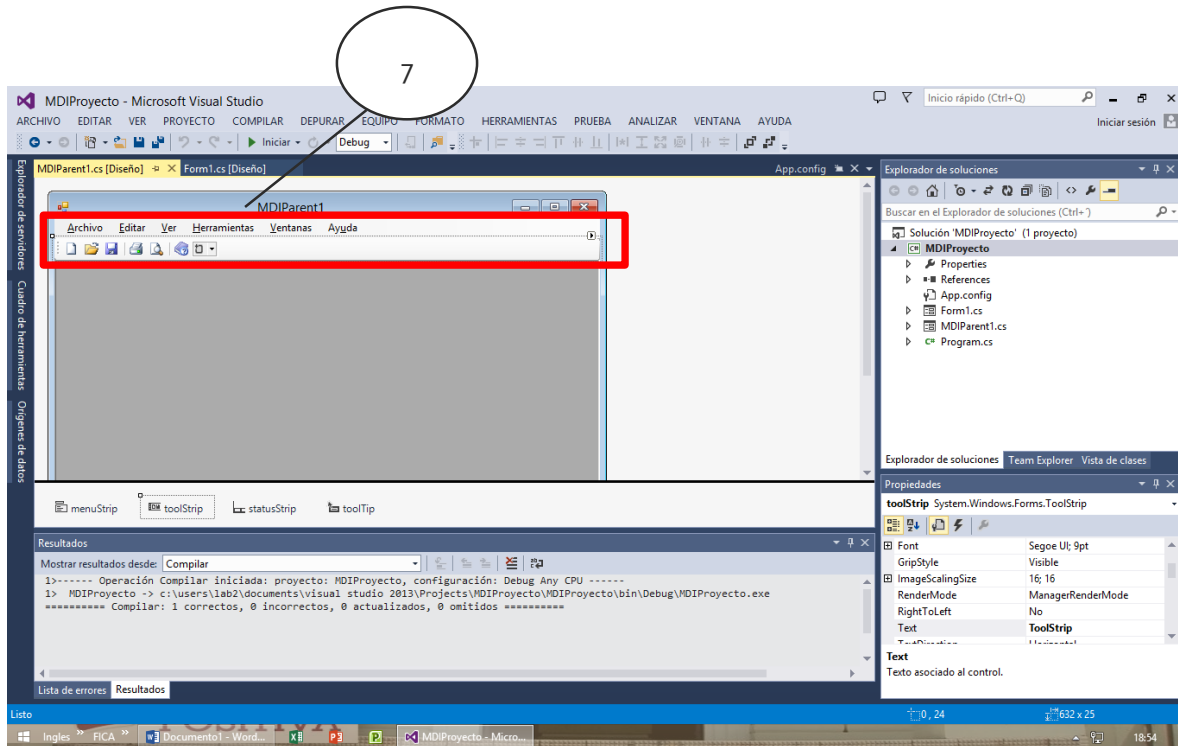
- 2- Seleccionamos aplicación de Windows form
- 3- Colocamos el nombre
- 4- aceptar



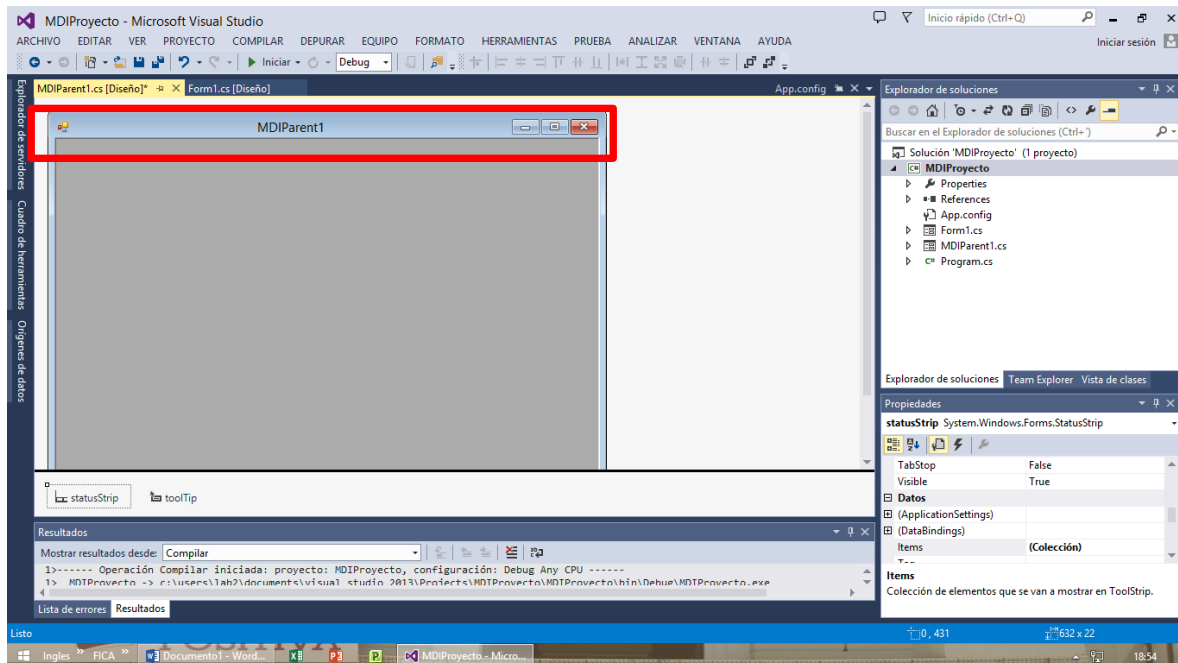
5- damos click derecho al proyecto y agregamos un nuevo elemento



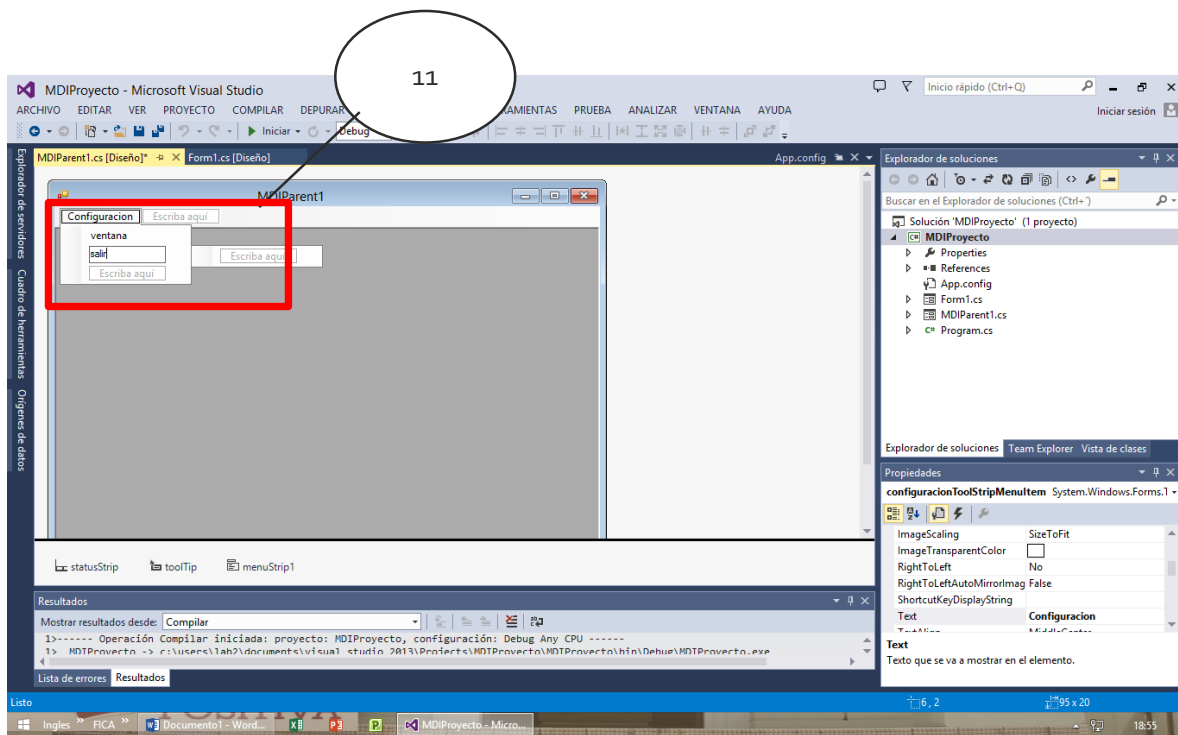
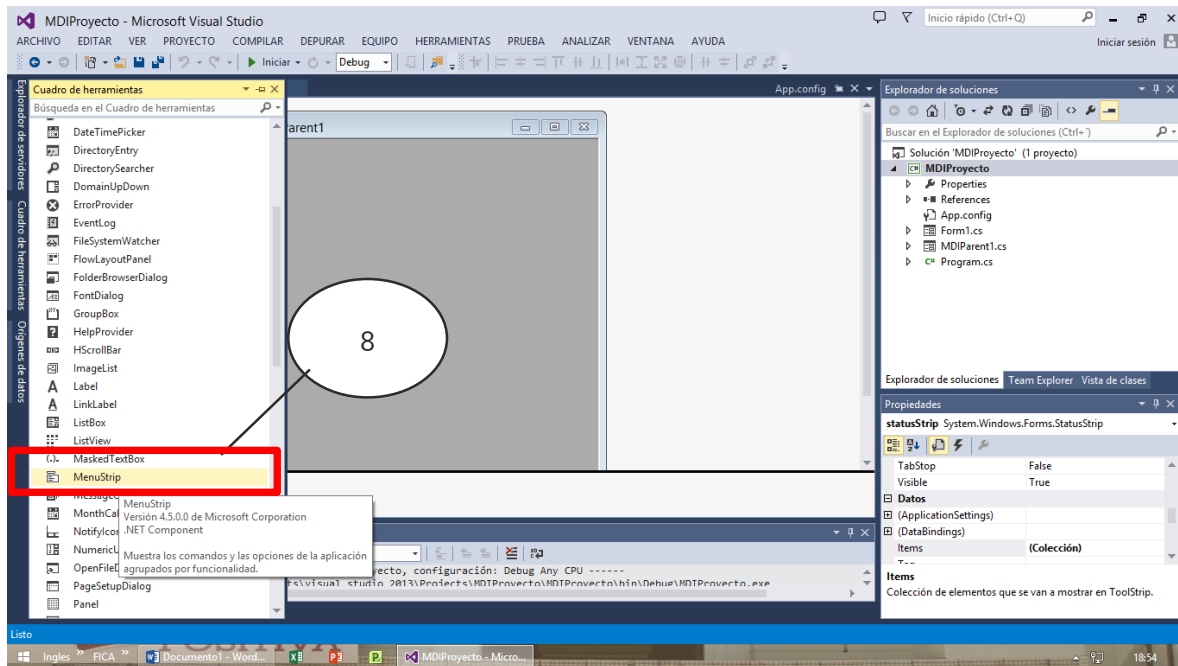
6- elegimos un formulario primario MDI y cambiamos el nombre

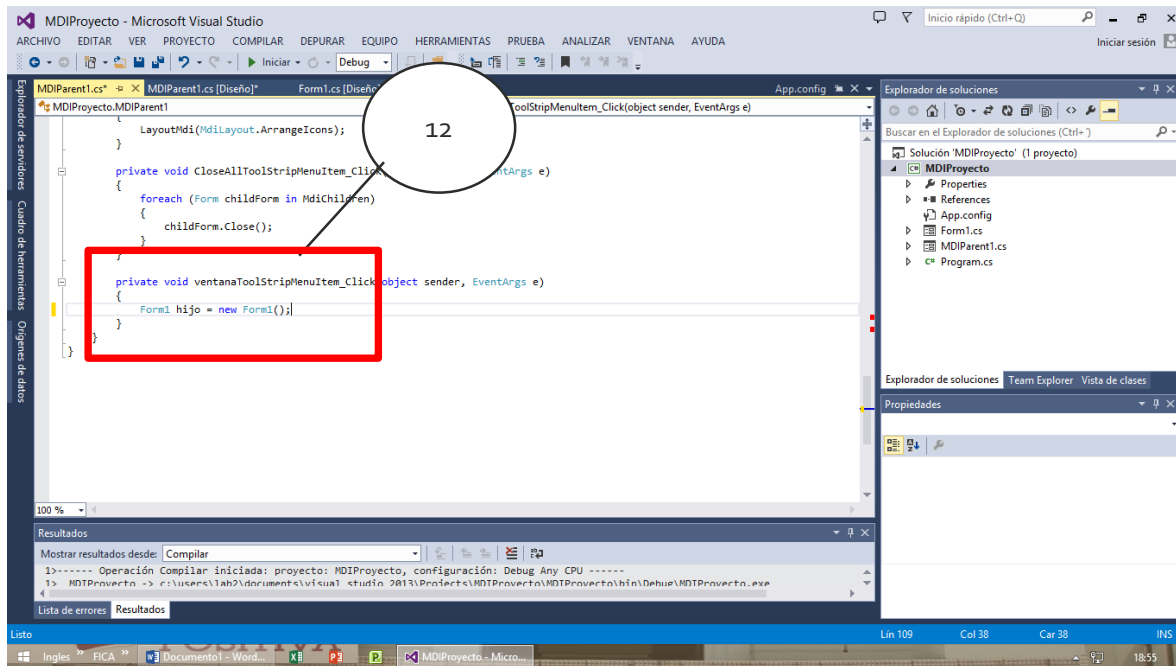


## 7- Eliminamos el menú predefinido









12- creamos un objeto

