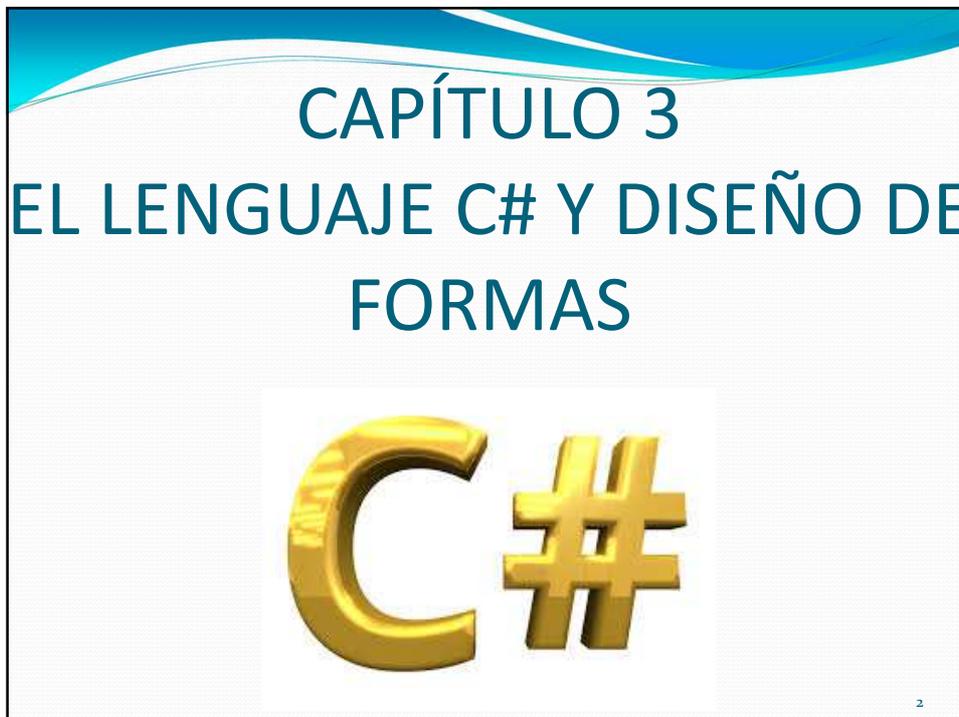


1



2

Preguntas detonadoras

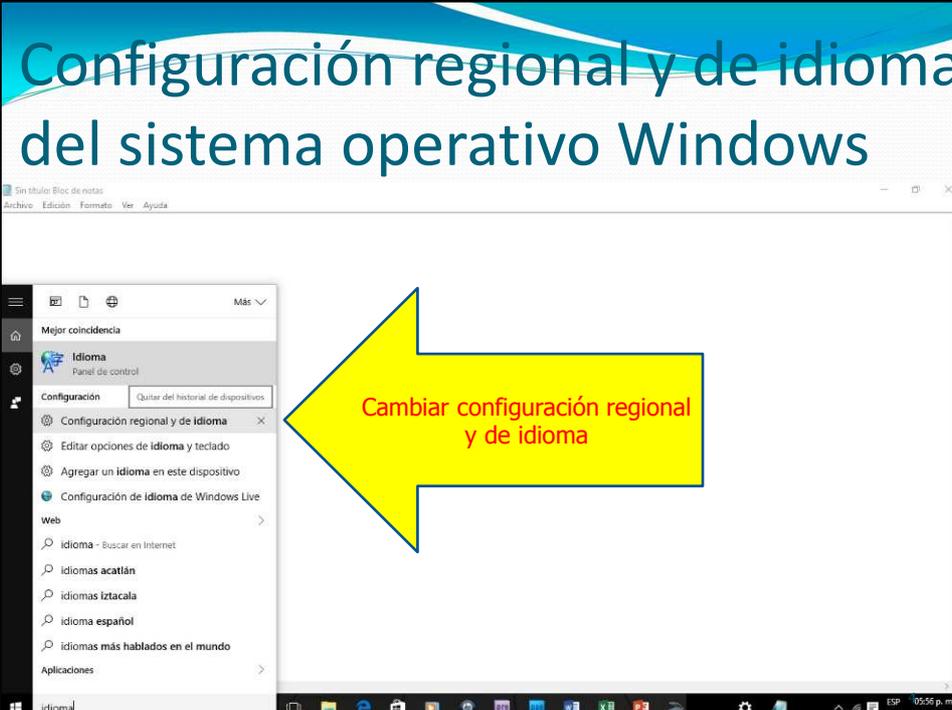


- ❑ ¿Qué es el framework .NET? ¿Cómo está organizado? ¿Para qué sirve?
- ❑ ¿Cuál es la herramienta de software necesaria para implementar aplicaciones en la plataforma .NET?
- ❑ ¿Representa lo mismo la programación visual que la programación orientada a objetos?
- ❑ ¿Se puede programar orientado a objetos en modo consola?
- ❑ ¿Cuáles son los controles visuales más comunes en una aplicación visual? ¿Cómo se utilizan?
- ❑ ¿Por qué es importante utilizar nomenclaturas estándar para identificar los componentes de una aplicación?
- ❑ ¿Cuáles son las recomendaciones de nomenclatura para identificar los componentes de una aplicación?

3

3

Configuración regional y de idioma del sistema operativo Windows



Cambiar configuración regional y de idioma

4

4

Seleccionar el país y el idioma

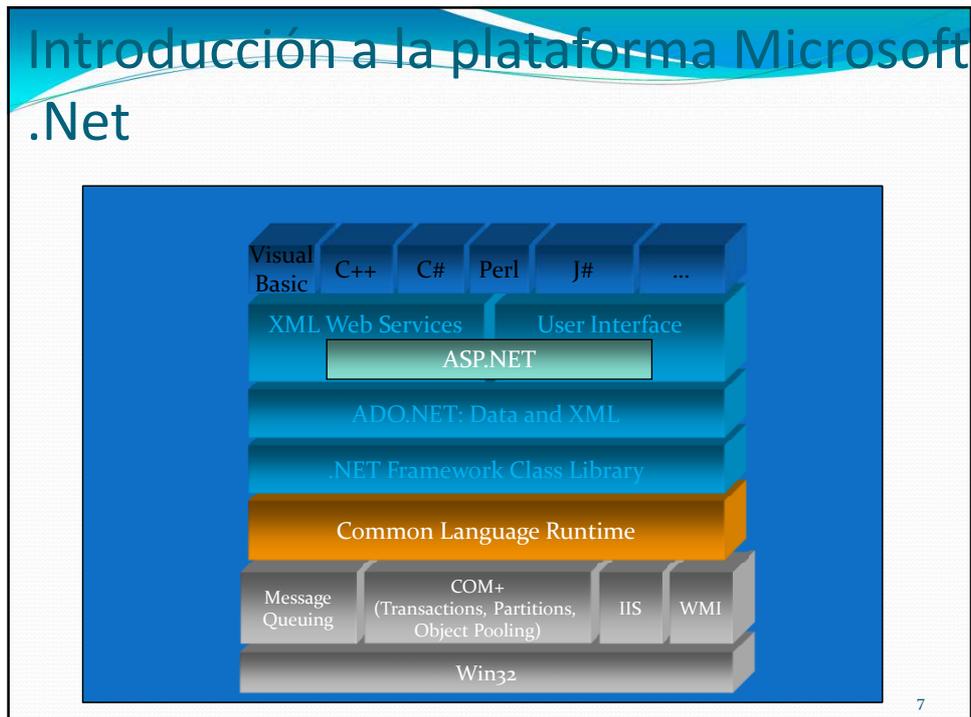
The screenshot shows the Windows Settings application, specifically the 'Hora e idioma' (Time & Language) section. The 'País o región' (Country or region) dropdown menu is set to 'México'. Below it, the 'Idiomas' (Languages) section shows a list of installed languages. The first language, 'Español (México)', is circled in red. Two yellow arrows point to these elements: one points to the 'México' dropdown with the text 'Seleccione el país deseado', and the other points to the circled 'Español (México)' entry with the text 'Asegúrese seleccionar el idioma correcto'. The page number '5' is visible in the bottom right corner of the screenshot.

5

Introducción a la programación en C# .NET

- **Introducción a la plataforma Microsoft .NET**
 - El framework .NET y sus componentes
 - Principales clases del framework .NET
- **Microsoft Visual Studio .NET**
 - Compilación y ejecución de programas
 - El debugger
 - Breakpoints
 - Watches
 - Ejecutar paso a paso
 - Examinar variables en tiempo de ejecución

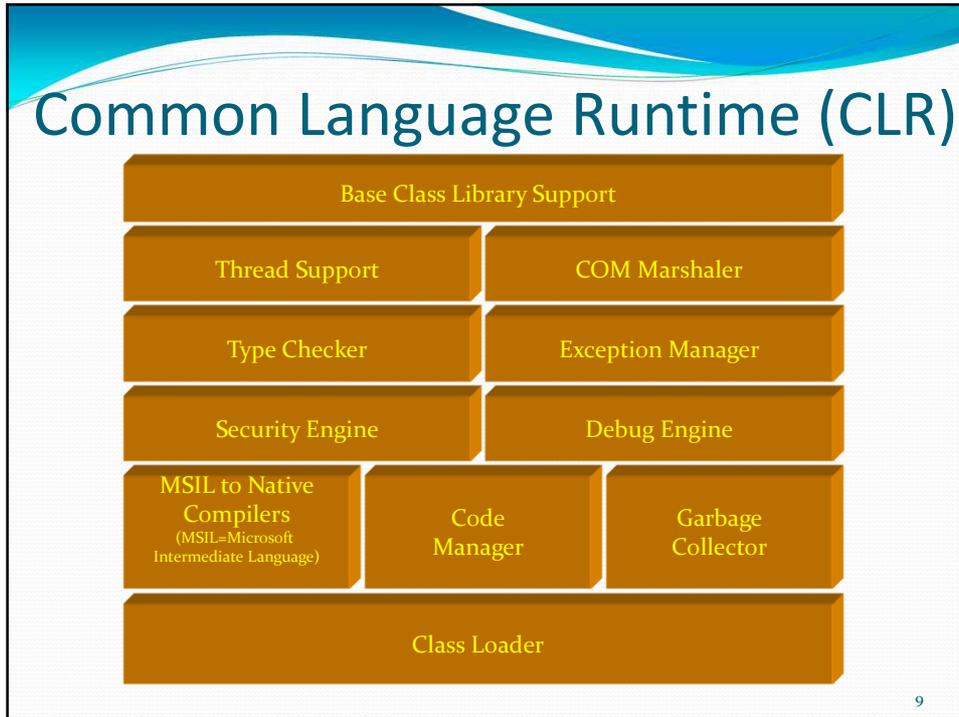
6



7



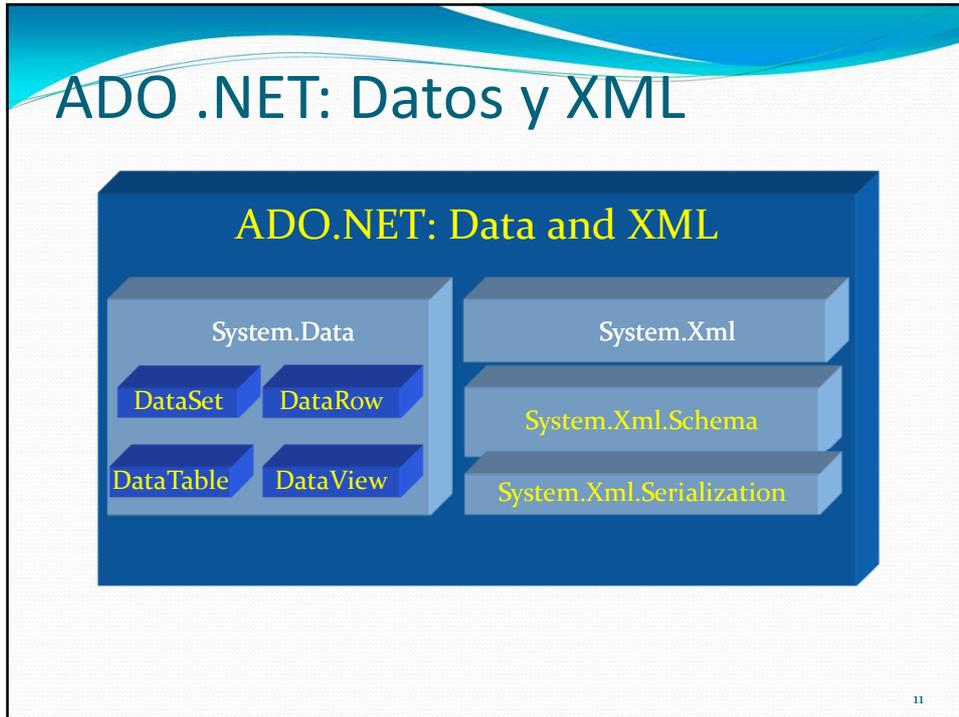
8



9



10

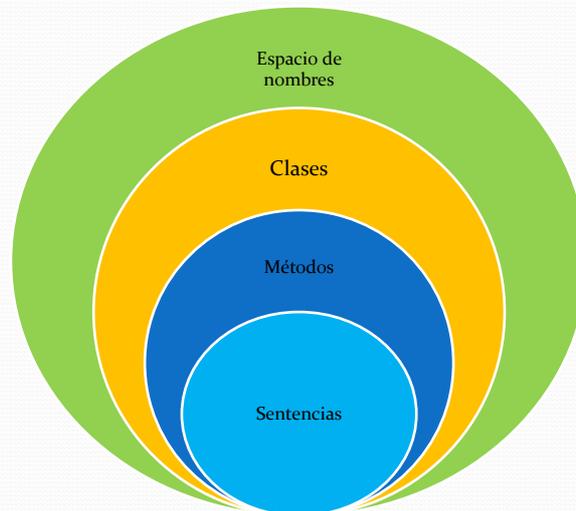


11



12

Estructura jerárquica de una aplicación en C#



13

13

Proyecto de consola en C#

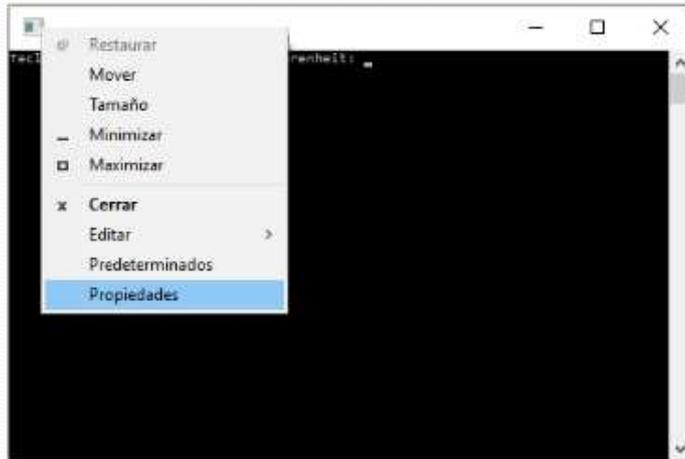
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Ejercicio1 // Espacio de nombres (nombre del proyecto)
{
    class Program // Clase del programa
    {
        static void Main(string[] args) // Método principal
        {
        }
    }
}
```

14

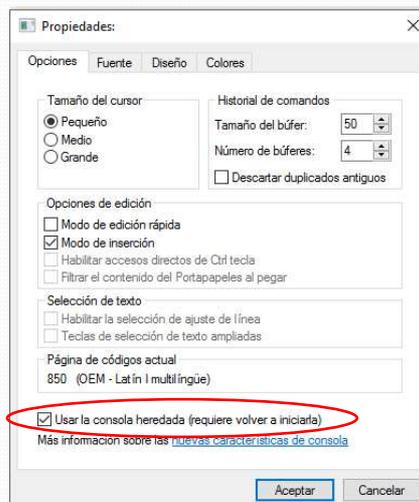
14

Configurar las propiedades de la consola



15

Configurar las propiedades de la consola (cont.)



16

Proyecto de formas en C#

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

namespace Ejercicio2 // Espacio de nombres (nombre del proyecto)
{
    public partial class Form1 : Form // Clase
    {
        public Form1()
        {
            InitializeComponent();
        }

        // Método principal que se ejecuta al cargar el proyecto
        private void Form1_Load(object sender, EventArgs e)
        {
        }
    }
}
```

17

17

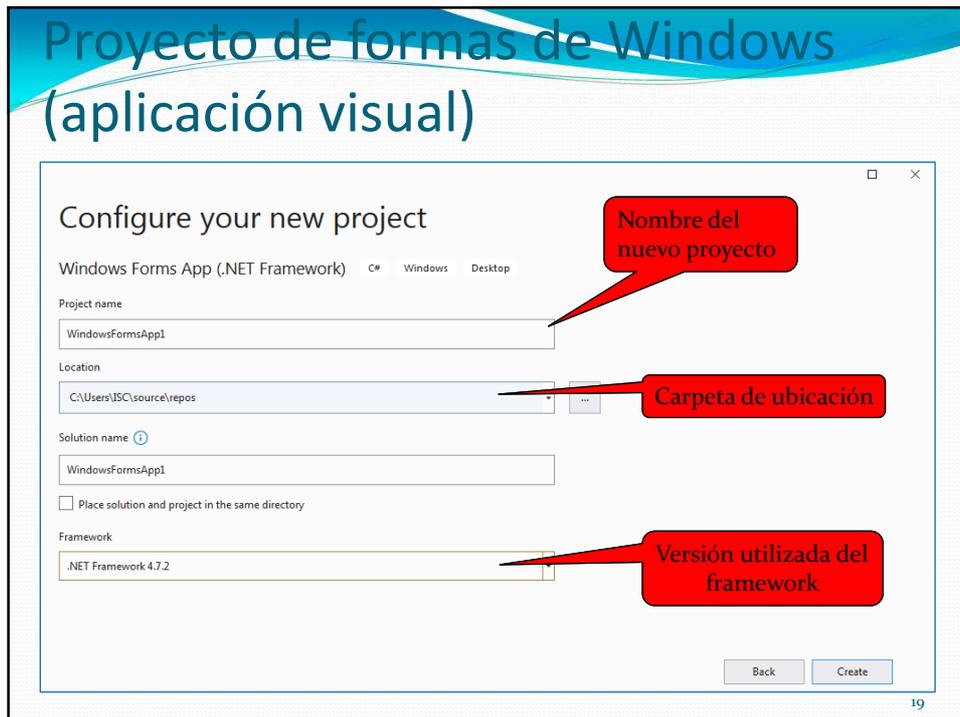
Proyecto de formas de Windows (aplicación visual)

Seleccionar el lenguaje C#

Proyecto de formas de Windows

18

18



19



20

Windows Forms App (.NET Framework) vs. Windows Forms App

- Windows Forms App (.NET Framework)**
 - Este tipo de aplicación contiene internamente la infraestructura del framework
 - Se puede seleccionar la versión a utilizar del framework
- Windows Forms App**
 - Este tipo de aplicación se ejecuta en versiones modernas de .NET evolucionada a partir de “.NET Core”
 - Contiene librerías que no están instaladas por defecto en las PC's (por lo que se deben instalar)
 - Pero .NET Core brinda opciones adicionales al framework como embeber el runtime dentro del ejecutable

21

21

ejemplo - Microsoft Visual Studio

Form1 [Diseño] | Página de inicio

Propiedades

Form1 System.Windows.Forms.Form

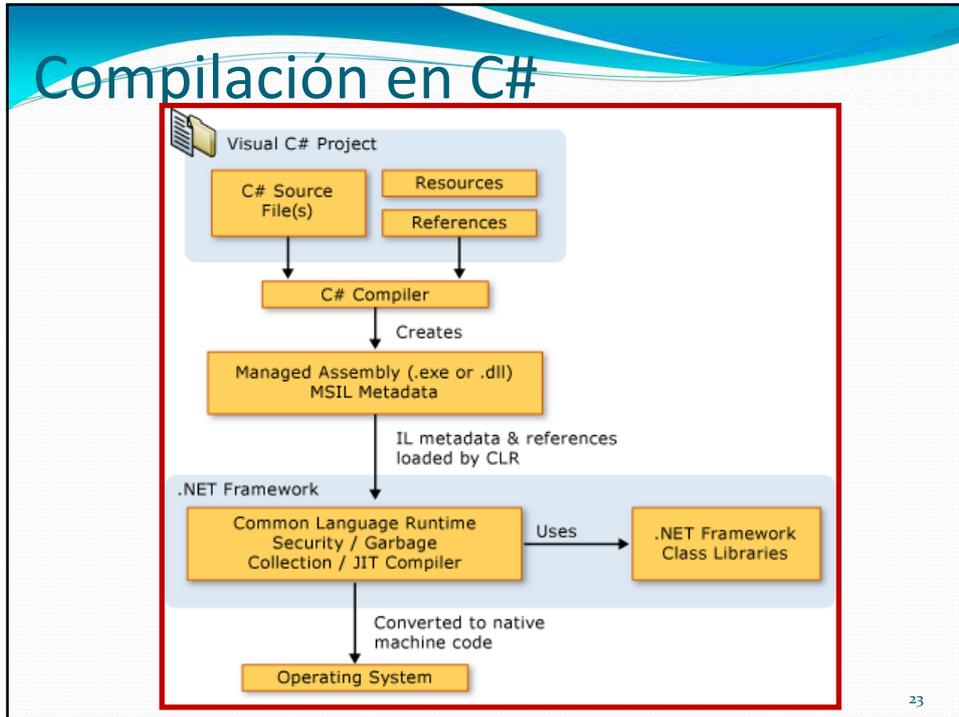
FormBorderStyle	Sizable
HelpButton	False
Icon	(Icono)
ImeMode	NoControl
IsMdiContainer	False
KeyPreview	False
Language	(Predeterminado)
Localizable	False
Location	0, 0
Locked	False
MainMenuStrip	(ninguno)
MaximizeBox	True
MaximumSize	0, 0
MinimizeBox	True
MinimumSize	0, 0
Opacity	100%
Padding	0, 0, 0, 0
RightToLeft	No
RightToLeftLayout	False
ShowIcon	True
ShowInTaskbar	True
Size	300, 300
SizeGripStyle	Auto
StartPosition	WindowsDefaultLoc...
Tag	
Text	Form1
TopMost	False

Forma de Windows

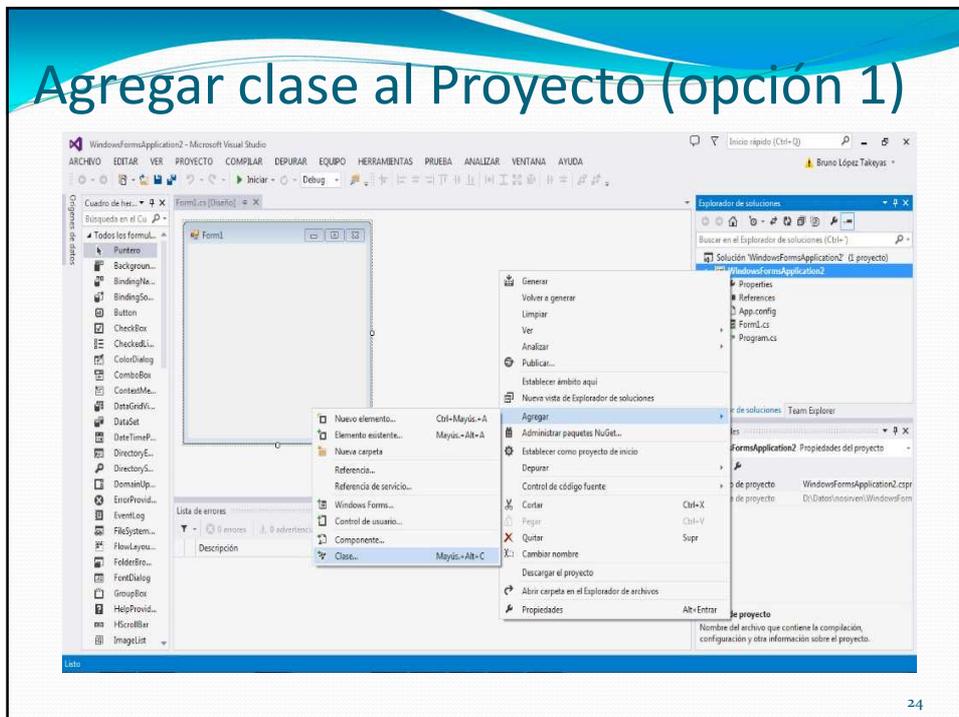
Tabla de Propiedades

Cuadro de controles

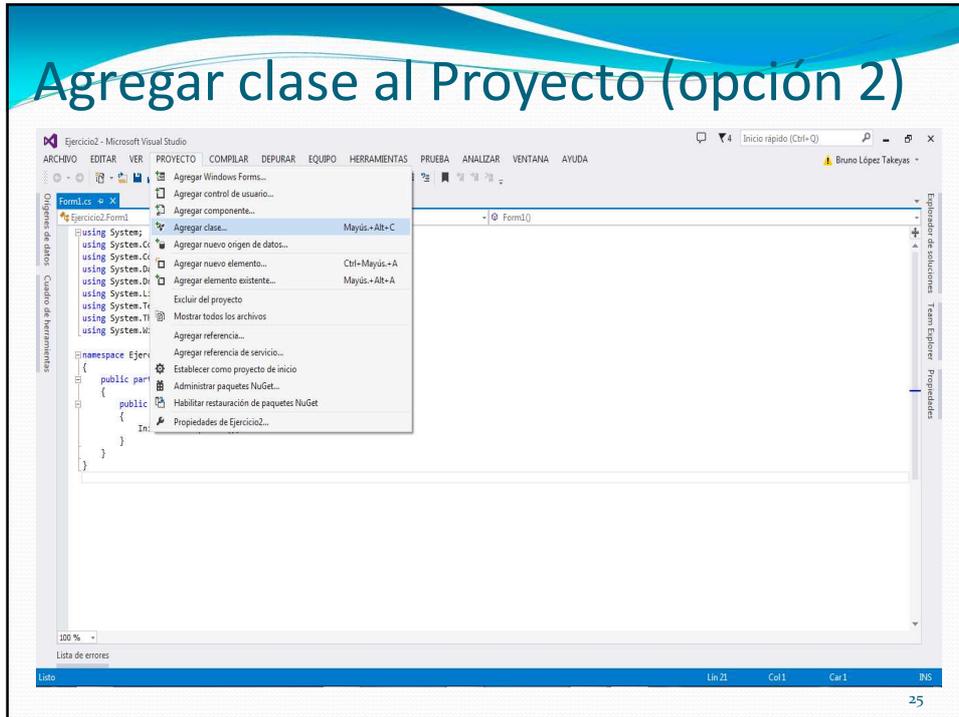
22



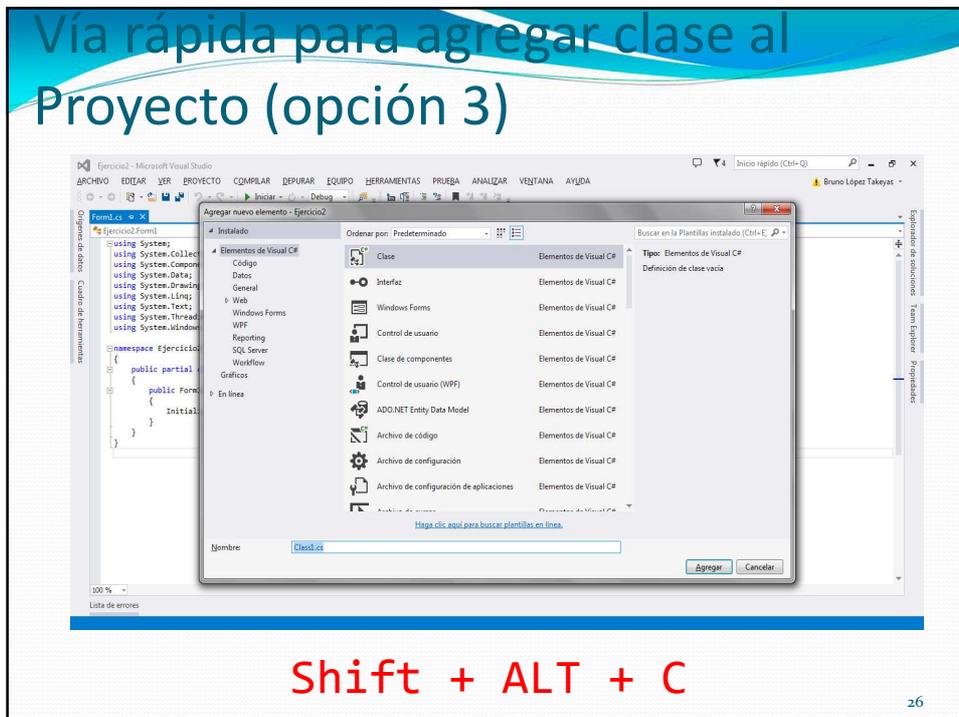
23



24



25



Shift + ALT + C

26

Propiedades "Name" y "Text"

- Todos los controles tienen la propiedad **Name** que los identifica como un objeto de la forma
- La propiedad **Text** se utiliza para especificar el texto que despliega el control

27

27

Propiedades "Name" y "Text"

Form1

Ejecutar

Propiedades

button1 System.Windows.Forms.Button

(DataBindings)	
(Name)	button1
AccessibleDescription	
AccessibleName	
AccessibleRole	Default
AllowDrop	False
Anchor	Top, Left
AutoEllipsis	False
AutoSize	False
AutoSizeMode	GrowOnly
BackColor	Control
BackgroundImage	(ninguno)
BackgroundImageLayout	Tile
CausesValidation	True
ContextMenuStrip	(ninguno)
Cursor	Default
DialogResult	None
Dock	None
Enabled	True
FlatAppearance	
FlatStyle	Standard
Font	
ForeColor	ControlText
GenerateMember	True
Image	(ninguno)

28

28

Controles visuales

Botón: Ejecuta un conjunto de Sentencias cuando se oprime

CheckBox: Se utiliza para seleccionar varias opciones de un conjunto

Etiqueta: Muestra un mensaje fijo En la forma

Lista: Muestra una cuadro con una lista de mensajes

RadioBotón: Se utiliza para seleccionar Sólo una opción de un conjunto

Cuadro de Texto: Se utiliza para introducir o mostrar datos

29

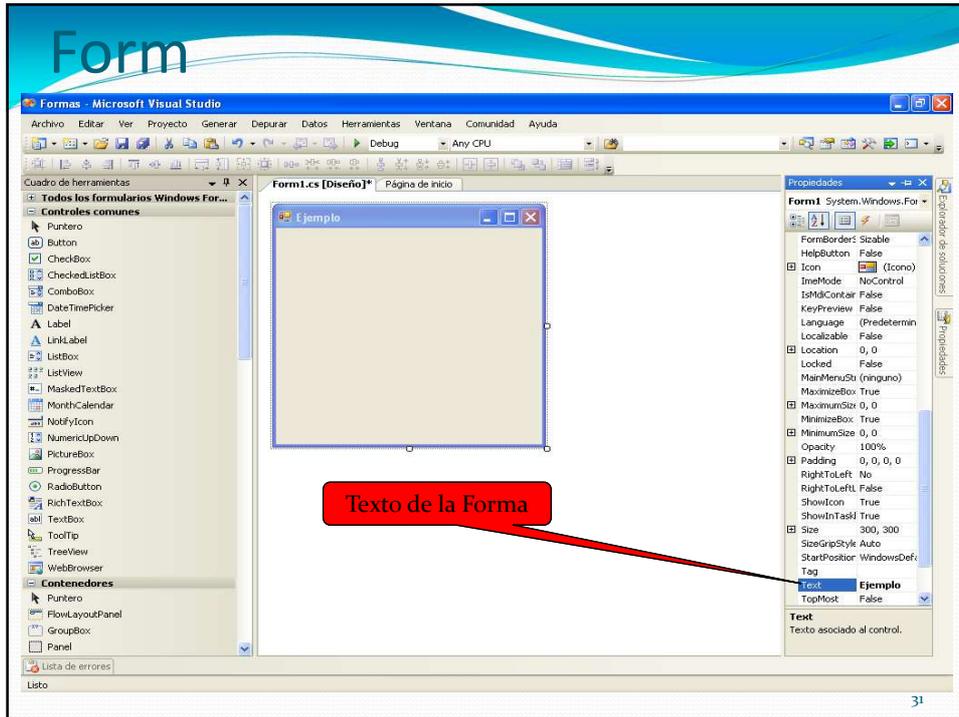
29

Terminar una aplicación

`Application.Exit();`

30

30



31

textBox

- Se usa para capturar datos
- Para ver su contenido se usa:
 - `textBox1.Text`
- Siempre es de tipo cadena
- Si requiere otro tipo de dato, es necesario usar `Parse()`
 - `int x = int.Parse(textBox1.Text);`
- Para limpiar su contenido:
 - `textBox1.Clear();`

Ejemplo

Dato

textBox1

label1

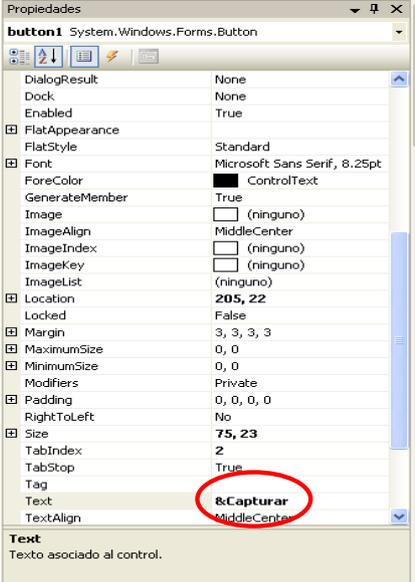
32

32

button

- Use la propiedad Text para colocar el mensaje al botón
- Coloque un & para habilitar la tecla directa
 - P. ejem. ALT-C
- Al dar doble click, codifique:

```
private void button1_Click(object sender, EventArgs e)
{
    //Aquí se coloca el código
}
```



33

33

MessageBox



`MessageBox.Show("Hola Mundo", "Este es un MessageBox", MessageBoxButtons.OK, MessageBoxIcon.Information);`



`MessageBox.Show("Hola Mundo");`

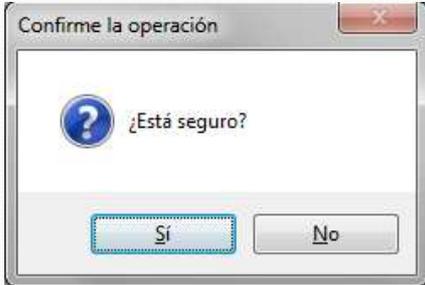


`MessageBox.Show("Hola Mundo", "Este es un Mensaje de Error", MessageBoxButtons.OK, Cancel, MessageBoxIcon.Error);`

34

34

MessageBox



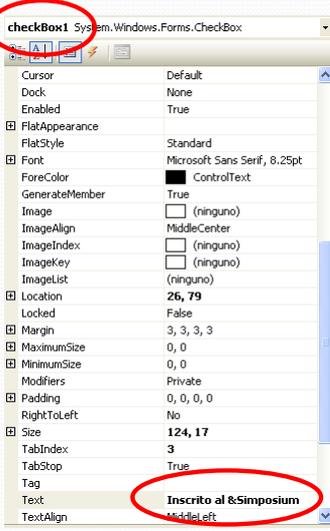
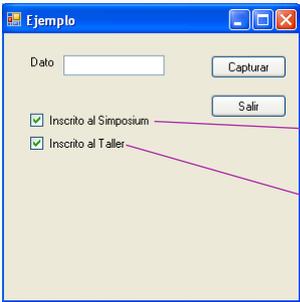
```
DialogResult Respuesta;  
Respuesta = MessageBox.Show("¿Está seguro?", "Confirme la operación",  
MessageBoxButtons.YesNo, MessageBoxIcon.Question);  
  
if (Respuesta==DialogResult.Yes)  
    MessageBox.Show("Aceptado");  
else  
    MessageBox.Show("Rechazado");
```

35

35

checkBox

■ Permite seleccionar varias opciones



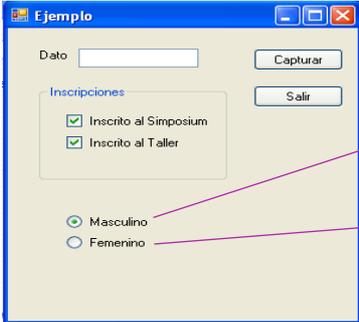
```
if (checkBox1.Checked)  
    MessageBox.Show("Opción : "+checkBox1.Text);  
  
if (checkBox2.Checked)  
    MessageBox.Show("Opción : "+checkBox2.Text);
```

36

36

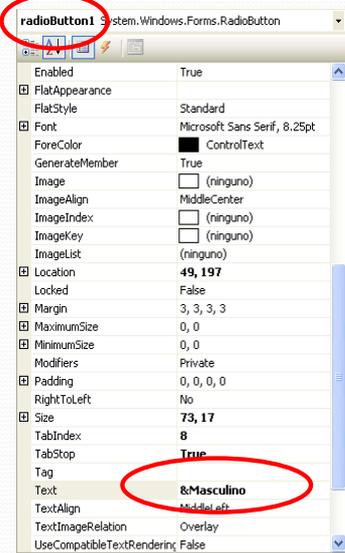
radioButton

■ Permite seleccionar **sólo una opción**



```
if (radioButton1.Checked)
    MessageBox.Show(radioButton1.Text);

if (radioButton2.Checked)
    MessageBox.Show(radioButton2.Text);
```

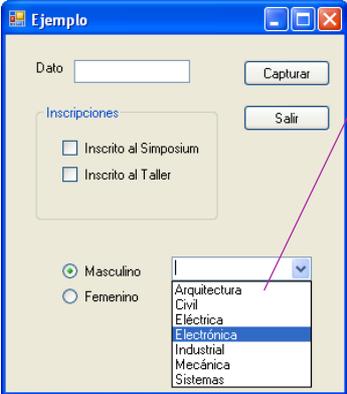


37

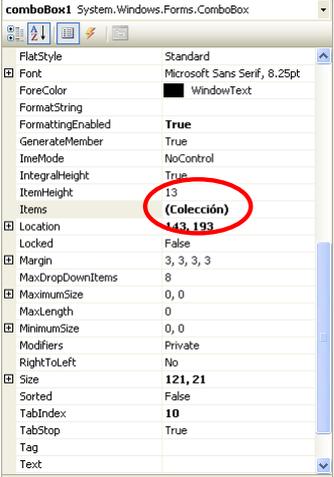
37

comboBox

■ Permite seleccionar **sólo una opción** de una lista desplegable



```
MessageBox.Show(comboBox1.Text);
```



38

38

comboBox

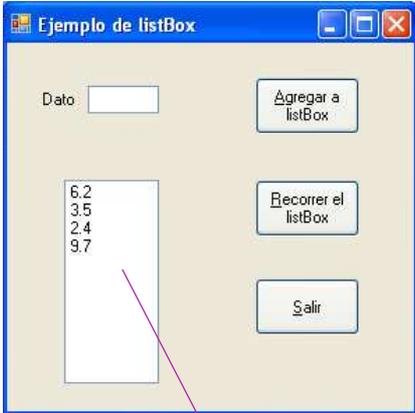
- **Modifique la propiedad DropDownStyle con el valor DropDownList para evitar que el usuario edite el contenido del comboBox y solamente seleccione alguna de las opciones disponibles**

39

39

listBox

- **Muestra una lista de datos de tipo string**
- **Es semejante a un arreglo**



- **Para agregar datos:**
 - `listBox1.Items.Add(textBox1.Text);`
- **Para recorrer el listBox:**

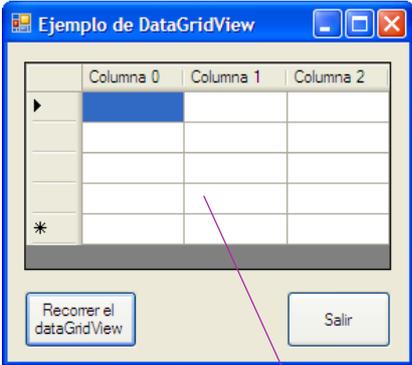
```
string salida = "";  
for (int i = 0; i < listBox1.Items.Count; i++)  
    salida = salida + "\n" + listBox1.Items[ i ];  
MessageBox.Show(salida);
```

40

40

dataGridView

- Muestra un conjunto de datos de tipo **string**
- Es semejante a un arreglo bidimensional (matriz)



- Para crear las columnas:

```
dataGridView1.Columns.Add("Columna 0", "Columna 0");  
dataGridView1.Columns.Add("Columna 1", "Columna 1");  
dataGridView1.Columns.Add("Columna 2", "Columna 2");
```

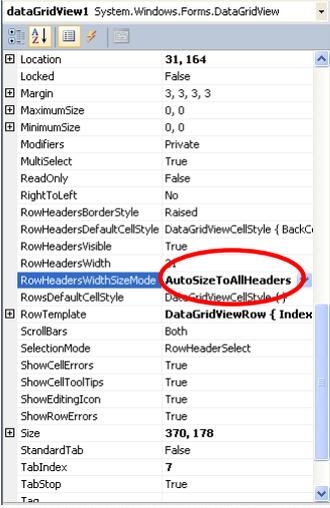
41

41

dataGridView

- Para etiquetar los renglones:

```
dataGridView1.Rows.Add(12);  
dataGridView1.Rows[0].HeaderCell.Value = "Enero";  
dataGridView1.Rows[1].HeaderCell.Value = "Febrero";  
dataGridView1.Rows[2].HeaderCell.Value = "Marzo";  
dataGridView1.Rows[3].HeaderCell.Value = "Abril";
```



42

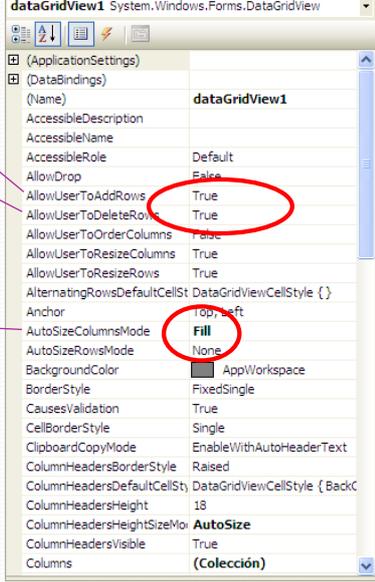
42

Propiedades de un dataGridView

Permitir agregar renglones

Permitir borrar columnas

Ancho de columna automatico

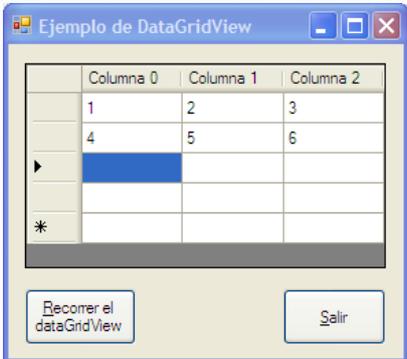


43

43

Recorrido de un dataGridView

- Se requieren 2 ciclos: Columnas y Renglones
- Cada celda se accesa mediante:
 - `dataGridView1[columna, renglon].Value`

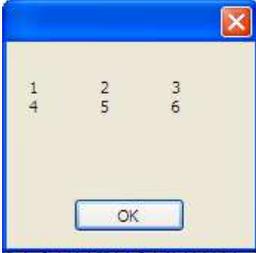


NOTA:
El direccionamiento de las celdas es: [columna, renglon]

44

44

Recorrido de un dataGridView



```
int r, c;
string salida = "";
for (r = 0; r < dataGridView1.Rows.Count; r++)
{
    salida = salida + "\n";
    for (c = 0; c < dataGridView1.Columns.Count; c++)
        salida = salida + dataGridView1[c, r].Value+"\t";
}
MessageBox.Show(salida);
```

45

45

Ajustando las propiedades del dataGridView

```
// No permitir agregar ni eliminar renglones
dataGridView1.AllowUserToAddRows = false;
dataGridView1.AllowUserToDeleteRows = false;

// Autoajustar el ancho de las columnas
dataGridView1.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill;

// Seleccionar un renglón completo al hacer click
dataGridView1.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
```

46

46

Ajustando las propiedades del dataGridView

```
// No se permite seleccionar varios renglones  
dataGridView1.MultiSelect = false;  
  
// Modo de solo lectura  
dataGridView1.ReadOnly = true;
```

47

47

Definir el formato de una columna del dataGridView

- Se puede definir el formato con el que se desea mostrar los datos de las columnas del dataGridView
- Se utiliza la propiedad `DefaultCellStyle.Format`
- Formatos disponibles:
 - *X = Hexadecimal*
 - *N = Numérico*
 - *C = Moneda (currency)*

48

48

Ejemplo para definir el formato de una columna del dataGridView

```

dgEmpleados.Columns.Add("Número", "Número");
dgEmpleados.Columns.Add("Nombre", "Nombre");
dgEmpleados.Columns.Add("Sueldo", "Sueldo");

dgEmpleados.Columns["Sueldo"].DefaultCellStyle.Format = "C";
    
```

```

private void btnCapturar_Click(object sender, EventArgs e)
{
    Empleado nuevoEmpleado = new Empleado();
    nuevoEmpleado.Numero = int.Parse(txtNumero.Text);
    nuevoEmpleado.Nombre = txtNombre.Text;
    nuevoEmpleado.Sueldo = double.Parse(txtSueldo.Text);

    dgEmpleados.Rows.Add(nuevoEmpleado.Numero,
        nuevoEmpleado.Nombre, nuevoEmpleado.Sueldo);
}
    
```

49

Ejemplo para definir el formato de una columna del dataGridView

	Número	Nombre	Sueldo
▶	4	Pepe	\$1,234.00
	5	Bruno	\$1,235.00

```

dgEmpleados.Columns["Sueldo"].DefaultCellStyle.Format = "C";
    
```

- Automáticamente muestra el sueldo del empleado con formato de moneda (currency) en la columna del dataGridView

50

50

¿Cómo pasar los datos de un renglón de un dataGridView a los textBoxes?

- Hacer doble click en el dataGridView para abrir el método `CellClick()`

```
private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
{
    . . .
}
```

- Establecer la propiedad para seleccionar solamente un renglón del dataGridView

```
// Seleccionar un renglón completo al hacer click
dataGridView1.SelectionMode = DataGridViewSelectionMode.FullRowSelect;
// No se permite seleccionar varios renglones
dataGridView1.MultiSelect = false;
```

51

51

¿Cómo seleccionar un renglón de un dataGridView?

```
// Declaración y creación de un objeto local
Auto miAuto = new Auto();

// Verificar si se seleccionó un renglón del dataGridView1
if (dataGridView1.CurrentRow == null)
{
    MessageBox.Show("Seleccione un auto de la lista");
    return;
}

// Obtiene las placas del auto seleccionado
miAuto.Placas = dataGridView1.CurrentRow.Cells[0].Value.ToString();
miAuto.Marca = dataGridView1.CurrentRow.Cells[1].Value.ToString();
miAuto.Modelo = dataGridView1.CurrentRow.Cells[2].Value.ToString();
```

52

52

Prefijos sugeridos para los nombres de los controles visuales (notación húngara)

Control	Prefijo	Ejemplo(s)
Button	btn	btnSalir
CheckBox	chk	chkInscrito
ComboBox	cbo	cboEspecialidad
DataGridView	dtg	dtgIngredientes
Form	frm	frmPrincipal
GroupBox	grp	grpDatosAlumnos
Label	lbl	lblNombre
ListBox	lst	lstAlumnos
RadioButton	rad	radMasculino
TextBox	txt	txtRadio

<http://support.microsoft.com/kb/173738/es>

53

53

Control	Prefijo	Ejemplo
Button	btn	btnSave
Calendar	cal	calMyDate
CheckBox	chk	chkMailList
CheckBoxList	chkl	chklAddress
ComboBox	cbo	cboProvincias
DataGrid	dtg	dtgClientes
DataList	dlst	dlstProductos
DateTimePicker	ntp	ntpFechaInicio
Dialog, ColorDialog, FontDialog	dlg	dlgFileSave, dlgColores
Image	img	imgLogo
Label	lbl	lblApellido
LinkLabel	llbl	llblWebPage
ListBox	lst	lstCompany
Listview	lvw	lvwRecibos
MainMenu	mnu	mnuArchivo
MenuItem	mnu	mnuCerrar
Panel	pnl	panSection
PictureBox	pic	picLogo
RadioButton	rad	radSex
RadioButtonList	rbl	rblAgeGroup
RichTextBox	rtb	rtbDocumento
Repeater	rep	repSection
TabControl	tab	tabUsuario
TextBox	txt	txtApellido
TreeView	twv	twvDirectorios
ToolBar	tbr	tbrStatus
Timer	tmr	tmrSegundos
Validators (Todos)	val	valCreditCardNumber
ValidationSummary	vals	valsErrors

4

54

Calcular diferencia de tiempo

- Se declaran las variables de tipo DateTime

```
DateTime dtmHoraEntrada, dtmHoraSalida;  
  
dtmHoraEntrada=DateTime.Now;  
.  
.  
.  
dtmHoraSalida=DateTime.Now;  
  
TimeSpan intervalo = dtmHoraSalida - dtmHoraEntrada;  
  
MessageBox.Show(intervalo.TotalMilliseconds);
```

55

55

Proyecto con varias formas



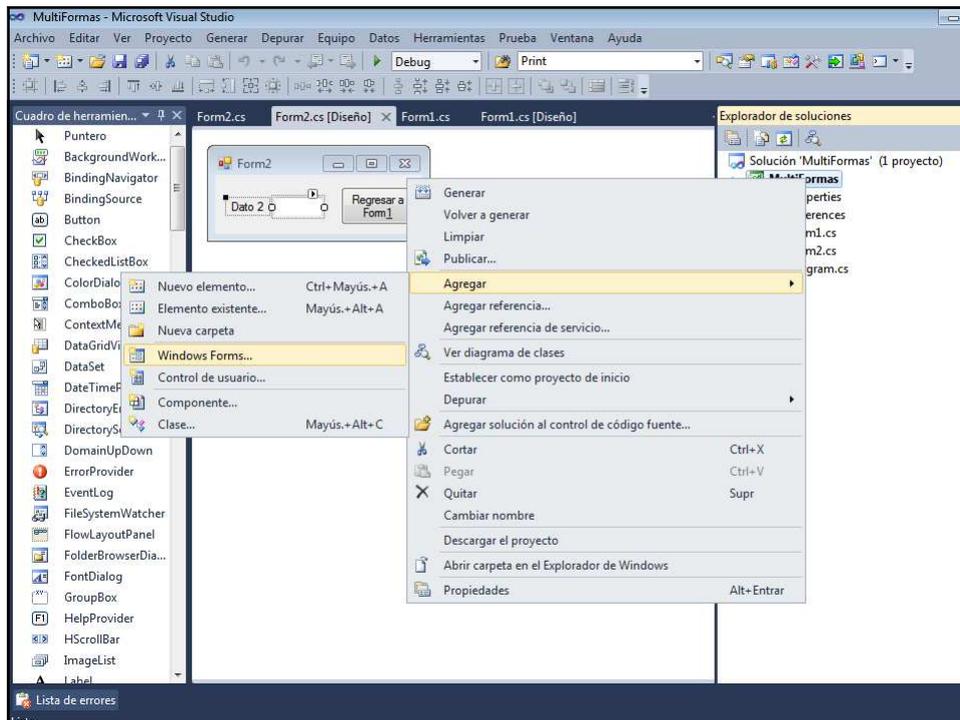
A screenshot of a Windows form titled 'Form1'. It contains a text box labeled 'Dato 1', a button labeled 'Abrir Form2', and a button labeled 'Salir'.



A screenshot of a Windows form titled 'Form2'. It contains a text box labeled 'Dato 2' and a button labeled 'Regresar a Form1'.

56

56



57

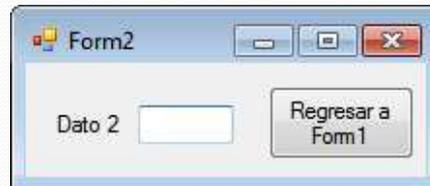
Invocar la Form2 desde la Form1

```
private void btnAbrirForm2_Click(object sender, EventArgs e)
{
    Form2 miForma2 = new Form2();
    miForma2.Show();
}
```

The image shows a slide with a blue header and a white background. The title is 'Invocar la Form2 desde la Form1'. Below the title is a screenshot of a Windows Form titled 'Form1'. The form has a text box labeled 'Dato 1', a button labeled 'Abrir Form2', and a button labeled 'Salir'. Below the screenshot is a code block with the following C# code:

58

Regresar a la Form1

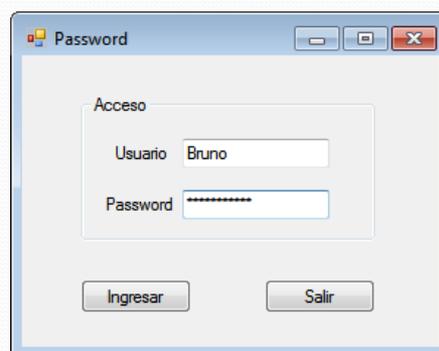


```
private void btnRegresarForm1_Click(object sender, EventArgs e)
{
    this.Close();
}
```

59

59

Uso de contraseña (password)

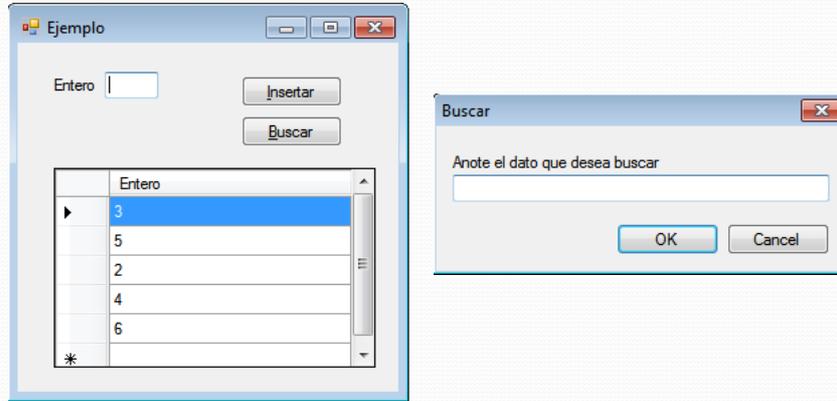


```
private void Form1_Load(object sender, EventArgs e)
{
    txtPassword.PasswordChar = '*';
}
```

60

60

Implementar un InputBox como en Visual Basic



61

61

```
public static DialogResult InputBox(string title, string promptText, ref string value)
{
    Form form = new Form();
    Label label = new Label();
    TextBox textBox = new TextBox();
    Button buttonOk = new Button();
    Button buttonCancel = new Button();

    form.Text = title;
    label.Text = promptText;
    textBox.Text = value;

    buttonOk.Text = "OK";
    buttonCancel.Text = "Cancelar";
    buttonOk.DialogResult = DialogResult.OK;
    buttonCancel.DialogResult = DialogResult.Cancel;

    label.SetBounds(9, 20, 372, 13);
    textBox.SetBounds(12, 36, 372, 20);
    buttonOk.SetBounds(228, 72, 75, 23);
    buttonCancel.SetBounds(309, 72, 75, 23);

    label.AutoSize = true;
    textBox.Anchor = textBox.Anchor | AnchorStyles.Right;
    buttonOk.Anchor = AnchorStyles.Bottom | AnchorStyles.Right;
    buttonCancel.Anchor = AnchorStyles.Bottom | AnchorStyles.Right;

    form.ClientSize = new Size(396, 107);
    form.Controls.AddRange(new Control[] { label, textBox, buttonOk, buttonCancel });
    form.ClientSize = new Size(Math.Max(300, label.Right + 10), form.ClientSize.Height);
    form.FormBorderStyle = FormBorderStyle.FixedDialog;
    form.StartPosition = FormStartPosition.CenterScreen;
    form.MinimizeBox = false;
    form.MaximizeBox = false;
    form.AcceptButton = buttonOk;
    form.CancelButton = buttonCancel;

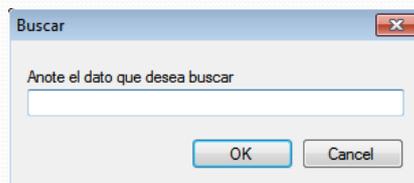
    DialogResult dialogResult = form.ShowDialog();
    value = textBox.Text;
    return dialogResult;
}
```

62

62

Invocando el InputBox

```
private void btnBuscar_Click(object sender, EventArgs e)
{
    string dato="";
    InputBox("Buscar", "Anote el dato que desea buscar", ref dato);
    MessageBox.Show(dato);
}
```



63

63

PictureBox

- Este control permite colocar una imagen en una forma



pictureBox
(picFoto)

Button
(btnCargarFoto)

64

64

Código del botón para seleccionar la imagen

```
private void btnCargarFoto_Click(object sender, EventArgs e)
{
    string strNombreArchivo=CargarImagen();

    if(strNombreArchivo != null)
        MostrarImagen(strNombreArchivo);
}
```

65

65

Método para seleccionar el archivo con la imagen

```
private string CargarImagen()
{
    string strNombreArchivo = null;

    // Declaración de variable para seleccionar el archivo
    OpenFileDialog miArchivoFoto = new OpenFileDialog();

    miArchivoFoto.Title = "Seleccione la imagen que desea cargar";
    miArchivoFoto.Filter= "Archivos JPEG (*.jpg) | *.jpg";
    miArchivoFoto.InitialDirectory = "Mis documentos";

    if (miArchivoFoto.ShowDialog() == DialogResult.OK)
    {
        strNombreArchivo = miArchivoFoto.FileName;
        return (strNombreArchivo);
    }
    else return (null);
}
```

66

Método para mostrar la imagen seleccionada en el pictureBox

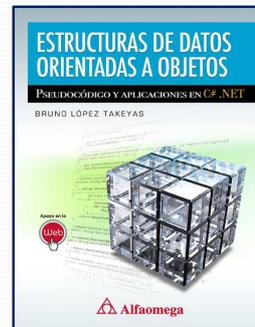
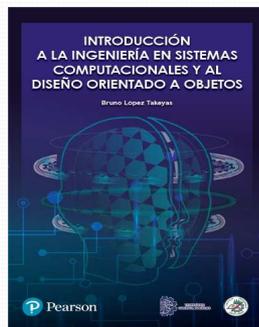
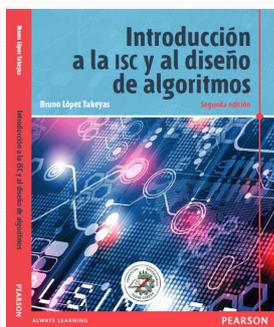
```
private void MostrarImagen(string strNombreArchivo)
{
    Bitmap miImagen = new Bitmap(strNombreArchivo);
    picFoto.SizeMode = PictureBoxSizeMode.StretchImage;
    picFoto.Image = (Image)miImagen;
    picFoto.Refresh();
}
```

67

67

Otros títulos del autor

<https://nlaredo.tecnm.mx/takeyas/Libro>



✉ bruno.lt@nlaredo.tecnm.mx

 Bruno López Takeyas

68