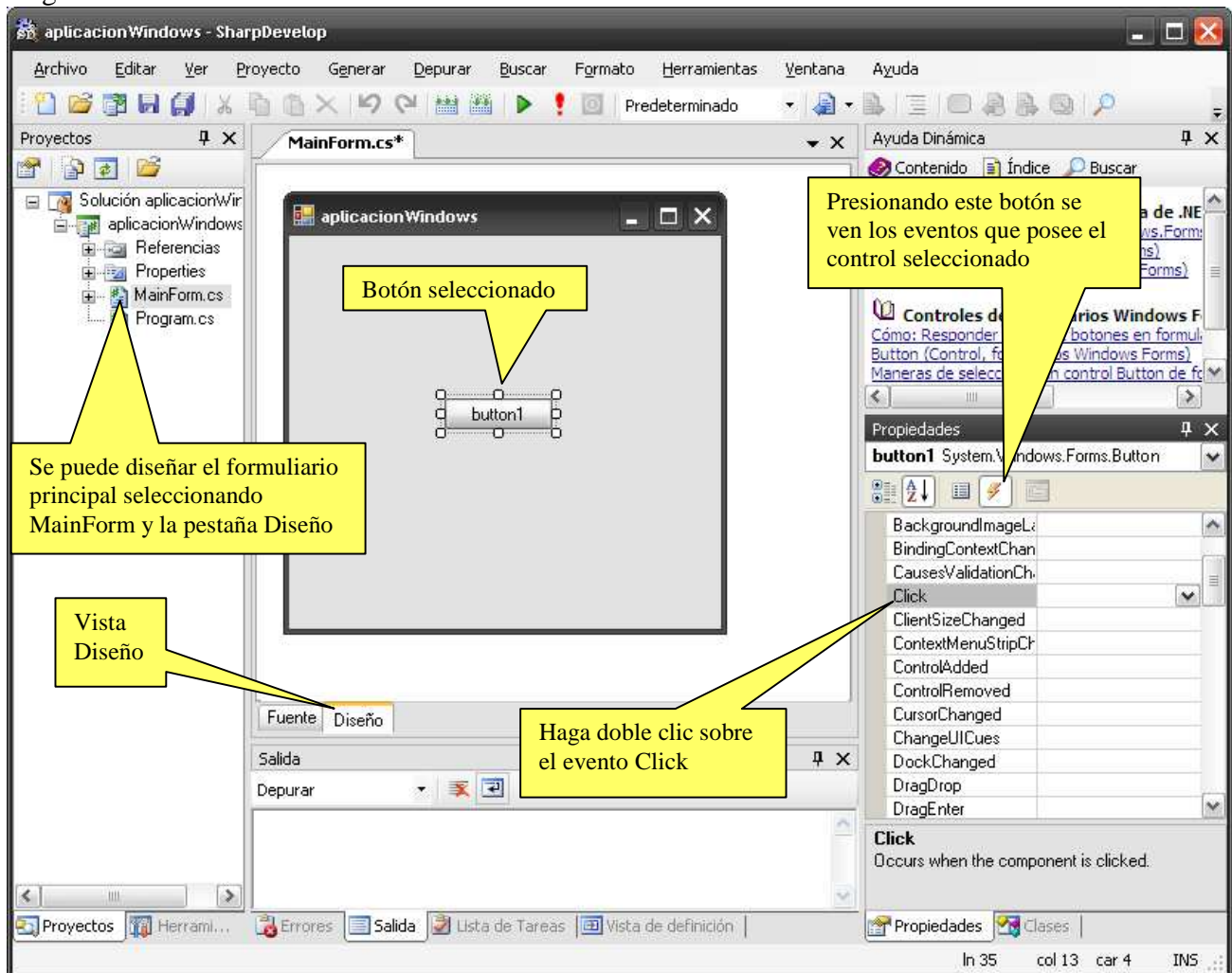


## Actividad sobre formularios Windows

### Tema: Aplicaciones Windows

1) Utilizando SharpDevelop cree una nueva Aplicación Windows. En la vista Diseño agregue un Button (botón) al formulario. Seleccione el Button que agregó y en el panel Propiedades haga clic sobre el botón Eventos para poder ver los eventos relacionados con el Button que pueden manejarse. Haga doble clic sobre el evento Click.



Conteste las siguientes preguntas:

- ¿Qué ocurre cuando se hace doble clic sobre el evento Click?
- Observe que a cada formulario se le asocian dos archivos, en nuestro ejemplo **MainForm.cs** y **MainForm.Designer.cs**. Revise ambos archivos para observar el código generado por SharpDevelop a partir de nuestras acciones en la vista Diseño. Particularmente observe esta porción de código:

```

this.button1 = new System.Windows.Forms.Button();
this.SuspendLayout();
//
// button1
//
this.button1.Location = new System.Drawing.Point(104, 109);
this.button1.Name = "button1";
this.button1.Size = new System.Drawing.Size(75, 23);
this.button1.TabIndex = 0;
this.button1.Text = "button1";
this.button1.UseVisualStyleBackColor = true;
this.button1.Click += new System.EventHandler(this.Button1Click);

```

- b.1. ¿En qué línea se crea la instancia button1?
- b.2. ¿De qué clase es la instancia button1 y en qué espacio de nombres está definida?
- b.3. Por miembros de una clase entendemos: campos, propiedades, métodos, eventos y clases anidadas ¿Qué tipo de miembro es button1.Text?
- b.4. ¿Qué tipo de miembro es button1.Click?
- b.5 ¿Puede reconocer el significado de la instrucción sombreada?
- b.6 ¿A quién hace referencia this? Ayuda: Esta porción de código está dentro del método `private void InitializeComponent()` observe en qué clase está definido este método.
- b.7 ¿Qué tipo de miembro es this.Button1Click?
- b.8 ¿Cuál es la clase en la que se definió el miembro Button1Click?

2) Codifique el método `Button1Click` de la siguiente manera:

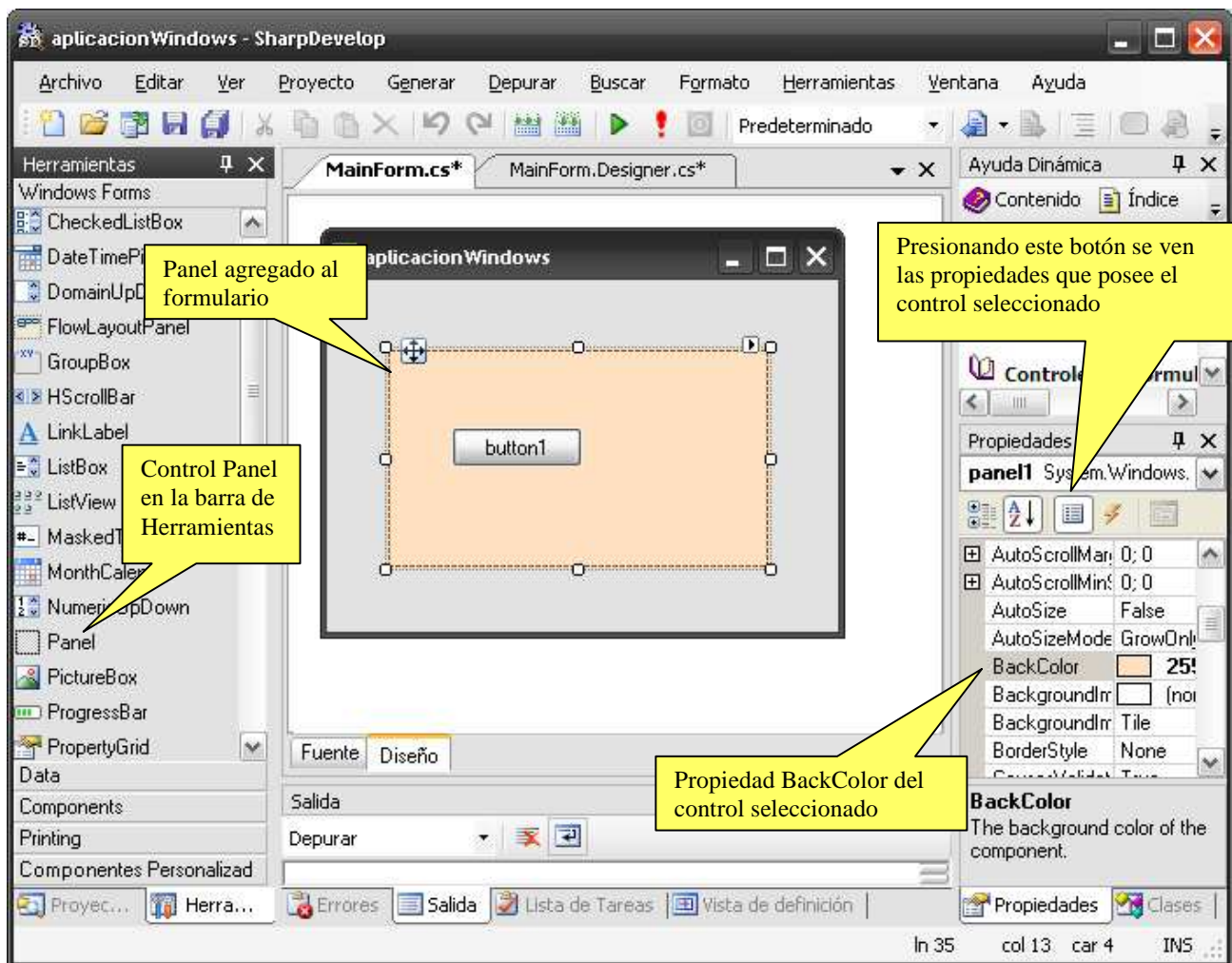
```

void Button1Click(object sender, EventArgs e)
{
    button1.Left += 10;
}

```

- a) Ejecute y observe el comportamiento del programa cuando hace clic sobre el Button.
- b) Qué ocurre si en lugar de `button1.Left += 10;` pone `this.Left += 10;`
- c) Ahora pruebe lo siguiente `this.Left += 10;` `button1.Left -= 10;`
- d) Experimente cambiando los valores de las propiedades Top Width y Height
- e) Para qué sirve la propiedad Text?

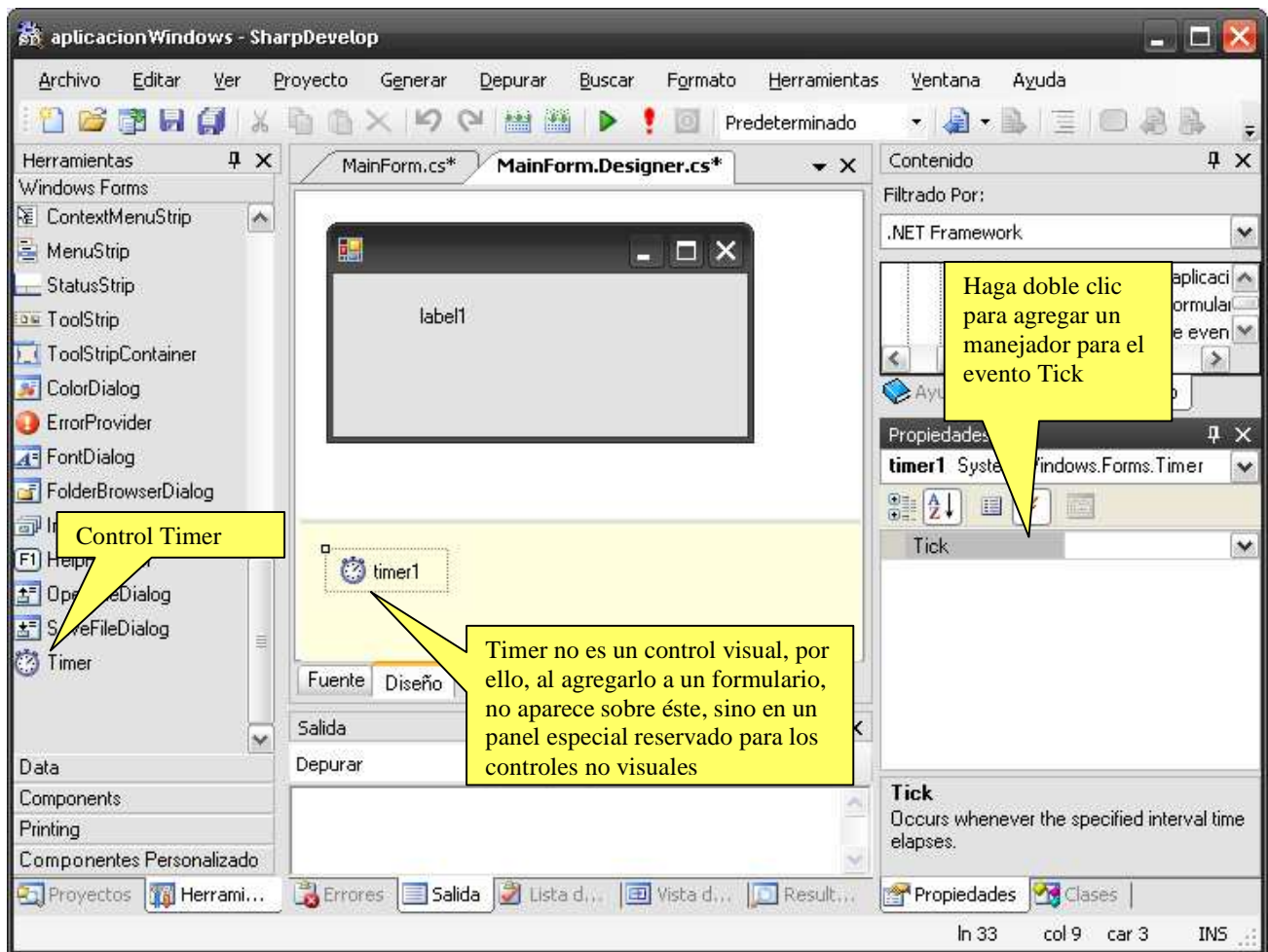
3) Coloque un Panel en el formulario y arrastre el Button dentro del panel. Cambie la propiedad BackColor del Panel para que éste sea identificable cuando el programa esté en ejecución.



- Ejecute el programa, presione el Button y observe cuál es la porción del formulario por la que el Button se puede mover.
- Si hace `button1.Left=0; button1.Top=0;` dónde se posiciona el Button?
- Experimente con la propiedad Dock del Button y del Panel. Para qué sirve? Observe que el formulario no posee esta propiedad.
- Experimente con la propiedad Enabled del Button y del Panel. Para qué sirve? Observe que el formulario también posee esta propiedad.

4) Cree una nueva Aplicación Windows. En la vista Diseño agregue un Label y un Timer. Establezca la propiedad Enabled del Timer en true y agregue el siguiente manejador para el evento Tick.

```
void Timer1Tick(object sender, System.EventArgs e)
{
    label1.Text=System.DateTime.Now.ToString("HH:mm:ss:fff");
}
```



- ¿Qué hace el programa?
- Experimente con distintos valores de la propiedad Interval del control Timer. ¿En qué unidad está expresado su valor?
- Agregue un Button para que al hacer clic cambie el valor de la propiedad Enabled del Timer. ¿Qué sucede, qué pasa con el evento Tick?

5) Cree una nueva Aplicación Windows. En la vista Diseño agregue un ListBox y un ColorDialog. Codifique un manejador para el evento Load del formulario de la siguiente manera:

```
void MainFormLoad(object sender, EventArgs e)
{
    listBox1.Items.Add(Color.Blue);
    listBox1.Items.Add(Color.Brown);
    listBox1.Items.Add(Color.Violet);
    listBox1.Items.Add(Color.Yellow);
    listBox1.Items.Add(Color.Red);
    listBox1.Items.Add("Personalizado");
}
```

Codifique un manejador para el evento SelectedIndexChanged del ListBox de la siguiente manera:

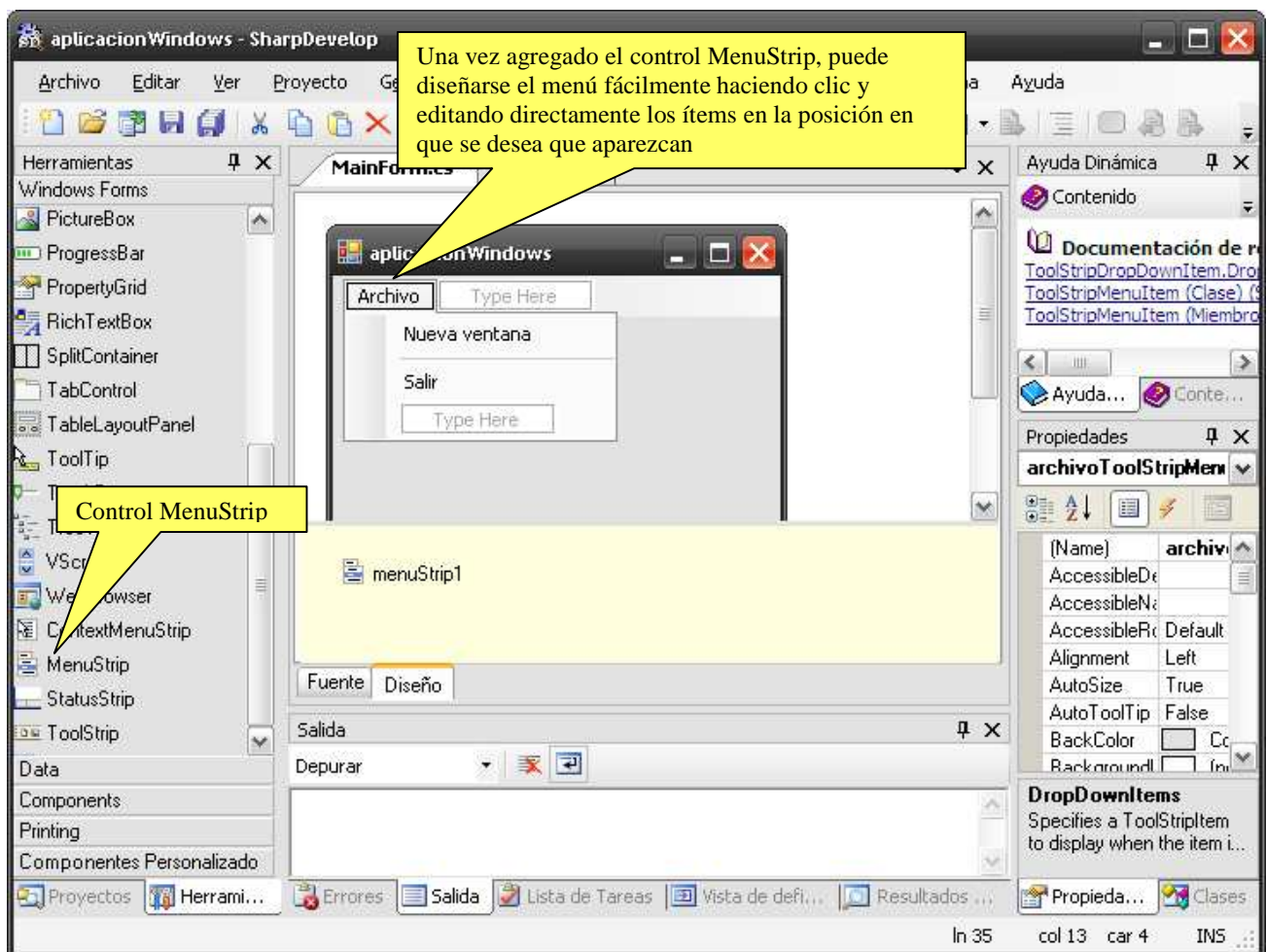
```

void ListBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    if(listBox1.SelectedIndex != listBox1.Items.Count - 1)
        this.BackColor = (Color)listBox1.SelectedItem;
    else {
        colorDialog1.ShowDialog();
        this.BackColor = colorDialog1.Color;
    }
}

```

- ¿Qué hace el programa?
- ¿Qué clase de objetos puede agregarse a la propiedad Items de un ListBox?
- ¿Qué ocurre si en el diálogo para elegir color se presiona el botón Cancelar?
- ¿Cómo haría para que al seleccionar el botón Cancelar no se haga cambio alguno? Consulte en la ayuda el método ShowDialog()

6) Cree una nueva Aplicación Windows. En la vista Diseño agregue un MenuStrip. Diseñe un menú como el que se ve en la imagen de abajo.





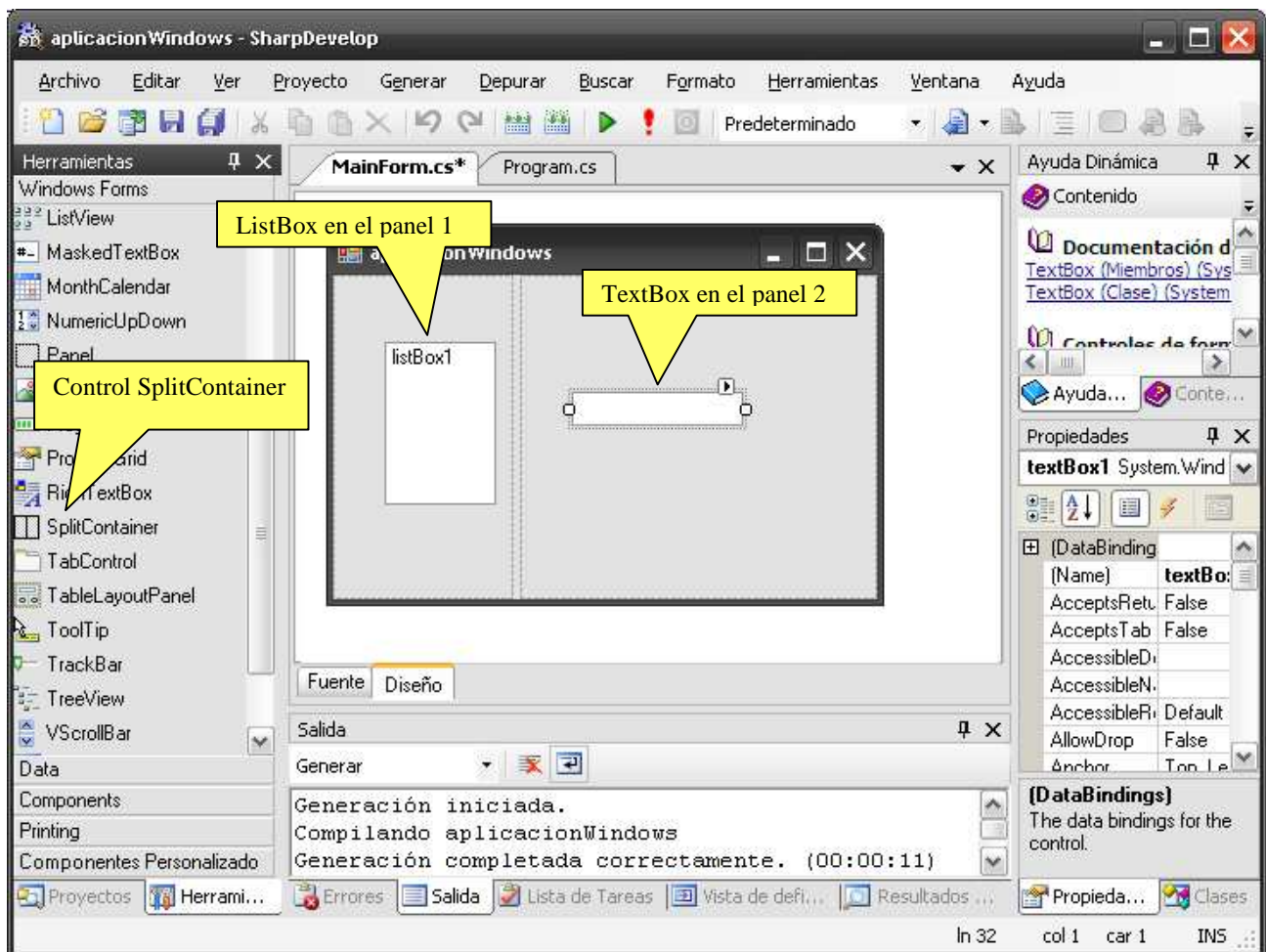
Codifique los eventos Click de ambos items de menú de la siguiente manera:

```
void NuevaVentanaToolStripMenuItemClick(object sender, EventArgs e)
{
    (new MainForm()).ShowDialog();
}

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

- Ejecute el programa y observe su comportamiento
- Cambie el método ShowDialog() por el método Show(). ¿Cuál es la diferencia entre ambos métodos?
- ¿Qué ocurre si teniendo varias ventanas abiertas elige cerrar la primera de todas ellas?

7) Cree una nueva Aplicación Windows. En la vista Diseño agregue un SplitContainer. En el panel1 de este control agregue un ListBox y en el panel2 un TextBox.

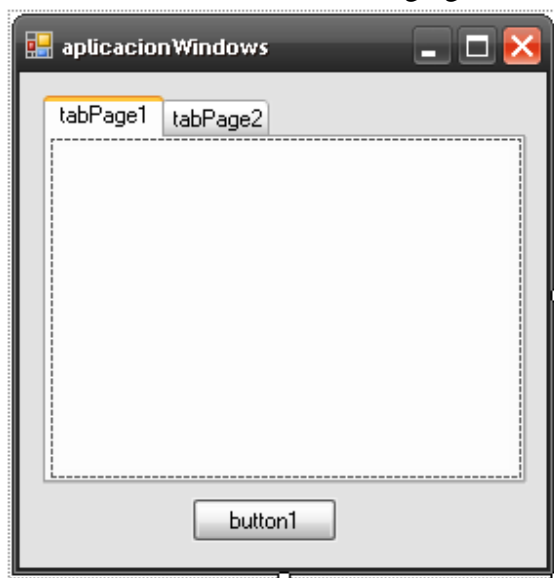


Codifique el manejador para el evento Load del formulario de la siguiente manera:

```
void MainFormLoad(object sender, EventArgs e)
{
    listBox1.Dock=DockStyle.Fill;
    textBox1.Dock=DockStyle.Fill;
    textBox1.Multiline=true;
    System.IO.DirectoryInfo dir = new System.IO.DirectoryInfo(@"c:\windows");
    listBox1.Items.AddRange(dir.GetFiles("*."));
}
```

- a) ¿Cuál es el comportamiento del control SplitContainer?
- b) ¿A qué clase pertenece la variable dir?
- c) ¿Qué devuelve el método GetFiles?
- d) ¿Cuál es la diferencia entre los métodos AddRange y Add de la propiedad Items del ListBox?
- e) ¿Para qué sirve la arroba (@) en la cadena @"c:\windows"?
- f) Modifique el programa para que sólo se muestren los archivos de extensión .ini.
- g) Modifique el programa para que al seleccionar un ítem del ListBox se muestre el contenido del archivo en el TextBox. Para leer un archivo utilice un objeto de la clase System.IO.StreamReader.
- h) Para qué sirven las propiedades Multiline y ScrollBars del TextBox?
- i) Modifique el programa para que sólo muestre los archivos de extensión .jpg y .bmp.
- j) En lugar de utilizar un TextBox en el panel derecho utilice un PictureBox para mostrar las imágenes seleccionadas en el ListBox.
- k) ¿Qué método debe utilizar para cargar una imagen en un PictureBox?
- l) ¿Para qué sirve la propiedad SizeMode del PictureBox?

8) Cree una nueva Aplicación Windows. En la vista Diseño agregue un TabControl y un Button.



Codifique los manejadores para el evento Load del formulario y Click del Button de la siguiente manera:

```
void MainFormLoad(object sender, EventArgs e)
{
    foreach(TabPage tp in tabControl1.TabPages)
        tabControl1.TabPages.Remove(tp);
}
void Button1Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Desea agregar una nueva pestaña",
        "Confirmación", MessageBoxButtons.YesNo) == DialogResult.Yes){
        MonthCalendar calendario=new MonthCalendar();
        calendario.Dock=DockStyle.Fill;
        calendario.TodayDate=DateTime.Today.AddDays(tabControl1.TabPages.Count);
        TabPage tp=new TabPage(calendario.TodayDate.ToString("dd/MM"));
        tp.Controls.Add(calendario);
        tabControl1.TabPages.Add(tp);
    }
}
```

- a) ¿Qué hace el manejador para el evento Load del formulario?
- b) ¿De qué clase es el objeto devuelto por la propiedad TabPages del control TabControl?
- c) Consulte en la ayuda por la clase MessageBox. ¿El método `show` es estático o de instancia? ¿Está sobrecargado?
- d) Identifique la línea de código en la que se añade una pestaña al TabControl
- e) Identifique la línea de código en la que se añade un control al TabPage. ¿De qué clase es el control añadido?
- f) ¿Puede añadir un control dentro de un TextBox? ¿Y dentro de un Button? ¿En general, a qué controles pueden añadirse controles en su interior?